

MASTERARBEIT

---

# KI Unterstützter DevOps Prozess: Arten und Herausforderungen

---



FACHHOCHSCHULE DER WIRTSCHAFT

Masterstudium  
IT und Wirtschaftsinformatik

by  
**Dario Wagner, BSc**  
1910835001

Graz, 11. Dezember 2023

---

Unterschrift

## **EHRENWÖRTLICHE ERKLÄRUNG**

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die benutzten Quellen wörtlich zitiert sowie inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, 11. Dezember 2023

---

Unterschrift

## ABSTRACT

This master's thesis explores the integration of artificial intelligence (AI) into the DevOps and ITOps processes. Through a systematic literature review, a total of 75 relevant papers were deeply investigated across various aspects, including opportunities, challenges, and the different types of artificial intelligence utilized.

The primary focus of this research is to construct a map outlining potential AI integrations within the DevOps process. Numerous studies in AI and Software or Operating systems inadvertently resolve various issues across different stages of DevOps. This study compiles a significant number of scientific papers supporting these stages and summarizes their methodologies.

This approach yields an overview of AI's support for DevOps and the associated challenges. Furthermore, this thesis lists the often-mentioned benefits of AI integration, along with the different KPIs and metrics used to measure the success of the AI method or the integration itself.

This thesis highlights how important it is to understand how AI fits into DevOps and ITOps. After looking at different studies, it's clear that AI can significantly improve how these processes work, making them faster and more accurate. The research suggests a full-picture approach, showing not just the good and challenging parts, but also ways to put AI into action and measure how well it works in DevOps and ITOps. All these findings aim to create a guide that helps professionals and companies use AI in these processes smartly and effectively.

**Keywords:** Campus02, Graz, Masterarbeit, DevOps, AI, Artificial Intelligence, ITOps, Künstliche Intelligenz, KI, IT Operations

## ZUSAMMENFASSUNG

Diese Masterarbeit erforscht die Integration künstlicher Intelligenz (KI) in die DevOps- und ITOps-Prozesse. Durch eine systematische Literaturübersicht wurden insgesamt 75 relevante Arbeiten in verschiedenen Bereichen eingehend untersucht, darunter Chancen, Herausforderungen und die verschiedenen Arten von künstlicher Intelligenz, die genutzt werden.

Der Hauptfokus dieser Forschung liegt darin, eine Abbildung zu erstellen, die potenzielle KI-Integrationen innerhalb des DevOps-Prozesses aufzeigt. Zahlreiche Studien zu KI und Software- oder Betriebssystemen lösen unbeabsichtigt verschiedene Probleme in verschiedenen Phasen von DevOps. Diese Studie fasst eine bedeutende Anzahl wissenschaftlicher Arbeiten zusammen, die diese Phasen unterstützen, und fasst ihre Methoden zusammen.

Dieser Ansatz bietet einen Überblick über die Unterstützung von KI für DevOps und die damit verbundenen Herausforderungen. Darüber hinaus listet diese Arbeit die häufig erwähnten Vorteile der KI-Integration sowie die verschiedenen KPIs und Metriken auf, die zur Messung des Erfolgs der KI-Methode oder der Integration selbst verwendet werden.

Diese Arbeit hebt die Bedeutung hervor, zu verstehen, wie KI in DevOps und ITOps passt. Nach der Betrachtung verschiedener Studien ist klar, dass KI die Arbeitsweise dieser Prozesse erheblich verbessern kann, indem sie sie schneller und genauer macht. Die Forschung schlägt einen ganzheitlichen Ansatz vor, der nicht nur die positiven und herausfordernden Aspekte zeigt, sondern auch Möglichkeiten aufzeigt, KI in DevOps und ITOps einzusetzen und zu messen, wie gut sie funktioniert. Alle diese Ergebnisse zielen darauf ab, einen Leitfaden zu erstellen, der Fachleuten und Unternehmen dabei hilft, KI in diesen Prozessen klug und effektiv zu nutzen.

**Keywords:** Campus02, Graz, Masterarbeit, DevOps, AI, Artificial Intelligence, ITOps, Künstliche Intelligenz, KI, IT Operations

# INHALTSVERZEICHNIS

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation und Ausgangsstellung . . . . .	1
1.2	Problemstellung . . . . .	1
1.3	Fragestellung . . . . .	2
1.3.1	Forschungsfrage . . . . .	2
1.3.2	Hypothesen . . . . .	2
1.3.3	Zielsetzung . . . . .	3
<b>2</b>	<b>DevOps</b>	<b>4</b>
2.1	Grundlagen . . . . .	4
2.1.1	Definition . . . . .	4
2.1.2	Phasen des DevOps Prozesses . . . . .	5
2.1.3	Prinzipien und Werte . . . . .	6
2.2	Continuous Integration und Continuous Delivery/Deployment . . . . .	10
2.2.1	Grundlagen . . . . .	10
2.2.2	Phasen . . . . .	10
2.2.3	Ziele . . . . .	11
2.2.4	Vorteile . . . . .	11
2.3	Automatisierung . . . . .	12
2.3.1	Test Automatisierung . . . . .	12
2.3.2	Continuous Testing . . . . .	13
2.4	Kontinuierliches Monitoring und Feedback . . . . .	13
2.4.1	Monitoring von Anwendungen und Infrastruktur . . . . .	13
2.4.2	Bedeutung von Log- und Metrik-Analyse . . . . .	14
2.4.3	Feedbackschleifen . . . . .	14
2.4.4	Key Performance Indicators (KPIs) . . . . .	14
2.5	Sicherheit in DevOps . . . . .	15
2.5.1	Sicherheit als kontinuierlicher Bestandteil von DevOps . . . . .	15
2.5.2	DevSecOps Integration . . . . .	15
2.5.3	Identifizierung und Behebung von Sicherheitslücken in DevOps . . . . .	16
2.6	Continuous Practices . . . . .	16
2.6.1	Continuous Integration . . . . .	17

2.6.2	Continuous Delivery . . . . .	18
2.6.3	Continuous Deployment . . . . .	18
2.6.4	Continuous Feedback . . . . .	19
<b>3</b>	<b>IT Operations</b>	<b>20</b>
3.1	Definition . . . . .	20
3.2	Infrastruktur . . . . .	20
3.3	Prozesse und Praktiken . . . . .	21
3.3.1	ITIL Framework . . . . .	22
3.3.2	DevOps Verbindung . . . . .	22
3.4	Herausforderungen . . . . .	22
3.5	Automatisierung . . . . .	23
3.5.1	Weitere Prozesse . . . . .	23
3.5.2	Strukturierte Daten . . . . .	24
3.5.3	Komplexität der Fehlererkennung . . . . .	24
3.5.4	Einsatz von KI . . . . .	24
<b>4</b>	<b>Künstliche Intelligenz</b>	<b>25</b>
4.1	Grundlagen . . . . .	25
4.1.1	Definition . . . . .	25
4.1.2	KI-Methoden . . . . .	25
4.1.3	Anwendungsbereiche von KI . . . . .	26
4.2	Kategorien von KI . . . . .	26
4.2.1	Datenzentrierte KI . . . . .	26
4.2.2	Modellzentrierte KI . . . . .	26
4.2.3	Anwendungszentrierte KI . . . . .	27
4.2.4	Menschenzentrierte KI . . . . .	27
4.3	Ethik und Verantwortung . . . . .	27
4.3.1	Herausforderungen und Bedenken im Umgang mit KI . . . . .	28
4.4	Zukunftsausblick und Forschungstrends in der KI . . . . .	28
4.4.1	Aktuelle Entwicklungen und Trends in der KI . . . . .	28
4.4.2	Herausforderungen und Chancen für die Zukunft der KI . . . . .	29
4.4.3	Ausblick auf mögliche Anwendungen und Forschungsrichtungen in der KI . . . . .	30
<b>5</b>	<b>AIOps</b>	<b>31</b>
5.1	Definition . . . . .	31
5.2	Vorteile und Herausforderungen . . . . .	31
5.3	Techniken . . . . .	32
<b>6</b>	<b>Methodik</b>	<b>34</b>
6.1	Wahl der Methode . . . . .	34
6.2	Arten . . . . .	34
6.2.1	Unsystematische Literaturrecherche . . . . .	34

6.2.2	Systematische Literaturrecherche . . . . .	34
6.3	Systematische Literaturrecherche . . . . .	35
6.3.1	Anwendungsbereich . . . . .	35
6.3.2	Konzeptualisierung . . . . .	36
6.3.3	Literatursuche . . . . .	37
6.3.4	Literaturanalyse und -synthese . . . . .	37
6.3.5	Forschungsagenda . . . . .	38
6.4	Durchführung der systematischen Literaturrecherche . . . . .	39
6.4.1	Iterationen . . . . .	39
6.4.2	Definieren des Überprüfungsrahmens . . . . .	39
6.4.3	Konzeptualisierung des Themas . . . . .	40
6.4.4	Literatursuche . . . . .	43
6.4.5	Literaturanalyse . . . . .	51
6.4.6	Forschungsagenda . . . . .	54
<b>7</b>	<b>Ergebnisse</b>	<b>55</b>
7.1	Ausschluss . . . . .	55
7.2	Verteilung der unterstützten DevOps Phasen . . . . .	56
7.3	Herausforderungen . . . . .	56
7.3.1	Verteilung der Kategorien . . . . .	57
7.3.2	Herausforderungen grob - je Phase . . . . .	58
7.3.3	Liste der Herausforderungen . . . . .	58
7.3.4	Lösungen . . . . .	59
7.3.5	Beispiele . . . . .	59
7.4	Verteilung KI-Techniken . . . . .	60
7.5	Verteilung KI-Methoden . . . . .	61
7.6	KI-Methoden - je Phase . . . . .	62
7.7	Vorteile . . . . .	63
7.7.1	Durchschnitt Anzahl Vorteile - je Phase . . . . .	64
7.8	KPIs . . . . .	64
<b>8</b>	<b>Conclusio</b>	<b>67</b>
8.1	Interpretation . . . . .	67
8.2	Hypothesenprüfung . . . . .	69
8.2.1	H1 . . . . .	69
8.2.2	H2 . . . . .	70
8.2.3	H3 . . . . .	71
8.3	Schlussfolgerung auf die Forschungsfrage . . . . .	72
8.3.1	Handlungsempfehlungen basierend auf dem Ergebnis . . . . .	74
8.4	Kritische Betrachtung . . . . .	75
8.4.1	Methode . . . . .	75
8.4.2	Fragebogen . . . . .	75

8.4.3 Limitation . . . . .	75
8.5 Weitere Forschungsmöglichkeiten . . . . .	76
<b>Literatur</b>	<b>77</b>
<b>Abbildungsverzeichnis</b>	<b>81</b>
<b>Tabellenverzeichnis</b>	<b>83</b>
<b>Literatursuche-Quellen</b>	<b>84</b>

# 1 EINLEITUNG

Das Einführungskapitel dient dem Leser als Einstieg in die vorliegende Arbeit. Sein Hauptziel ist es, den Leser mit dem grundlegenden Kontext der Untersuchung vertraut zu machen. In diesem Kapitel werden die Ausgangssituation und die spezifische Problemstellung erläutert, die die Durchführung dieser Studie erforderlich gemacht haben. Anschließend werden in Abschnitt 1.3 die Forschungsfrage und die übergeordneten Ziele der Arbeit eingehend erörtert. Dies beinhaltet eine gründliche Untersuchung der zentralen Forschungsfrage, der damit verbundenen Hypothesen und schließlich der übergreifenden Ziele. Darüber hinaus gibt das Kapitel 6 einen Überblick über die in der Studie angewandte Forschungsmethodik und beschreibt die Methoden und Techniken, die zur Datenerhebung und -analyse verwendet wurden.

## 1.1 Motivation und Ausgangssituation

In den letzten Jahren haben sich die Bereiche DevOps (Entwicklung und Betrieb) und IT-Betrieb durch die rasanten Fortschritte bei den Technologien der künstlichen Intelligenz (KI) erheblich verändert. Da Unternehmen nach mehr Effizienz, Automatisierung und Agilität in ihren Softwareentwicklungs- und IT-Verwaltungsprozessen streben, ist die Integration von KI sehr vielversprechend. In diesem Einführungskapitel soll die Motivation für die Erforschung der Möglichkeiten, die sich aus der Verschmelzung von KI und DevOps/IT-Operations ergeben, näher beleuchtet werden, um die dringende Notwendigkeit zu verdeutlichen, die aktuellen Herausforderungen zu bewältigen.

Durch die Untersuchung der Ausgangssituation schafft diese Arbeit eine Grundlage für das Verständnis der aktuellen Landschaft der KI-Integration in diesen Bereichen. Die zentrale Forschungsfrage in Kapitel 1.3.1 dient als Leitfaden und treibt diese Forschung in unbekanntes Terrain, um das Potenzial und die Hindernisse an der Schnittstelle von KI, DevOps und IT-Operations zu entschlüsseln.

## 1.2 Problemstellung

Im Bereich DevOps und IT-Betrieb hat die Forschung mit zahlreichen Studien und Untersuchungen, die sich mit diesen Themen befassen, erheblich an Bedeutung gewonnen. Mit dem Einzug der künstlichen Intelligenz in diesen Bereich sind viele neue Forschungsarbeiten entstanden, die insbesondere Aspekte der Automatisierung und der Steigerung der Prozesseffizienz beleuchten. Die Fülle der verfügbaren Studien und Erkenntnisse macht es jedoch zunehmend schwierig, einen klaren Überblick über die spezifischen

Möglichkeiten zu gewinnen, die künstliche Intelligenz im Kontext von DevOps und IT Operations bietet.

Die vorliegenden Kapitel der Masterarbeit sollen daher diese Schwierigkeiten überwinden, indem es eine umfassende Zusammenfassung der relevanten Forschungsergebnisse bietet. Dabei sollen nicht nur die Potenziale aufgezeigt, sondern auch die damit verbundenen Herausforderungen und Probleme angesprochen werden. Dadurch wird ein fundiertes Verständnis für die Auswirkungen und die praktische Anwendbarkeit von KI im Bereich DevOps und IT Operations gefördert.

## 1.3 Fragestellung

In diesem Kapitel soll die Forschungsfrage, daraus abgeleitete Hypothesen und die Zielsetzung definiert werden, um eine klare Struktur für die Arbeit zu schaffen und klar zu machen, was im Zuge der Forschung beantwortet wird.

### 1.3.1 Forschungsfrage

Künstliche Intelligenz (KI) oder Artificial Intelligence (AI) bringt mehr und mehr Anwendungsbereiche mit sich. Eine davon ist die Integration im Bereich der Software-Entwicklung. Ein Vorreiter dafür ist der DevOps Bereich, welcher sich mit Organisation und Struktur rund um die Software-Entwicklung beschäftigt. Wie in Kapitel 5 weiter erläutert, gibt es einige Phasen im DevOps Prozess, die sich zur Integration von KI eignen. Da sich viele Möglichkeiten selbst in diesem Bereich ergeben, stellt sich die Frage welche Arten der Integration sich über die Zeit entwickelt haben und welche Herausforderungen diese mit sich bringen.

Daher hat sich folgende Forschungsfrage ergeben:

*„Welche Möglichkeiten der Integration von KI im Bereich DevOps und IT Operations gibt es und welche Herausforderungen bringen diese mit sich?“*

### 1.3.2 Hypothesen

Aus der Forschungsfrage und derzeitigen Forschungen ergeben sich zudem folgende Hypothesen, welche im Laufe der Arbeit durch eine systematische Literaturrecherche erarbeitet und beantwortet werden.

#### 1.3.2.1 H1

**H1<sub>0</sub>:** Durch die Integration von KI im DevOps Prozess entstehen keine neuen Herausforderungen bzw. Probleme in Bezug auf die Bereiche Sicherheit, Ethik und Verantwortlichkeit der KI Anwendungen.

**H1<sub>1</sub>:** Wenn KI im DevOps Prozess integriert wird, entstehen neue Herausforderungen für die Sicherheit, Ethik und Verantwortlichkeit der KI Anwendungen.

### 1.3.2.2 H2

**H2<sub>0</sub>:** Wenn KI für Monitoring im DevOps Prozess eingesetzt wird, hat das keinen signifikanten Einfluss auf die Erkennung und Behebung von Anomalien in Softwareprodukten.

**H2<sub>1</sub>:** Wenn KI für Monitoring im DevOps Prozess wird, hat das einen signifikanten positiven Einfluss auf die Erkennung und Behebung von Anomalien in Softwareprodukten.

### 1.3.2.3 H3

**H3<sub>0</sub>:** Es kann nicht jeder Teil des DevOps Prozesses, wie Planung, Entwicklung, Testen, Bereitstellung und Überwachung, von KI unterstützt werden.

**H3<sub>1</sub>:** Jeder Teil des DevOps Prozesses, wie Planung, Entwicklung, Testen, Bereitstellung und Überwachung, kann von KI unterstützt werden.

## 1.3.3 Zielsetzung

Das Ziel dieser Masterarbeit besteht darin, eine umfassende Übersicht über die Integrationsmöglichkeiten von Künstlicher Intelligenz (KI) im gesamten DevOps-Prozess zu erstellen. Dies umfasst eine detaillierte Analyse der potenziellen Anwendungen von KI-Methoden wie Maschinelles Lernen, Natürliche Sprachverarbeitung und Neuronale Netze in den einzelnen Phasen des DevOps-Prozesses. Dabei werden spezifische Techniken wie Anomaly Detection, Root Cause Analysis, Datenintegration und Predictive ALM betrachtet, um deren mögliche Integration in jede Phase des Prozesses zu untersuchen. Ziel ist es, eine umfassende und strukturierte Übersicht zu schaffen, die die Potenziale und Herausforderungen der KI-Integration in DevOps klar und prägnant darstellt.

## 2 DEVOPS

Da für die Forschungsfrage und im praktischen Part der Arbeit der Begriff und seine Prinzipien unumgänglich sind, befasst sich dieses Kapitel mit seiner Definition und Grundlagen, Automatisierung, Konzepten und Prinzipien. Aber auch Themen wie Sicherheit werden in Bezug zu DevOps erklärt.

### 2.1 Grundlagen

Dieses Kapitel befasst sich mit allen Grundlagen des Begriffs, dazu gehört die Definition, der Einsatz von DevOps und dessen Prinzipien.

#### 2.1.1 Definition

Wie Len Bass (2015), in seinem Buch beschreibt, ist DevOps eine Methode oder eine Reihe von Praktiken, die darauf abzielt, die Zeit zwischen der Änderung eines Systems und dessen Überführung in die Produktion zu verkürzen, während gleichzeitig eine hohe Qualität sichergestellt wird. Zusammengefasst gibt es jedoch 5 Implikationen laut Len Bass (2015), welche bei dieser Definition beachtet werden müssen:

1. Die Qualität von Änderungen an einem System ist wichtig. Es gibt verschiedene Methoden, um die Qualität sicherzustellen, wie automatisierte Tests, Tests mit begrenzten Benutzern oder Überwachung von neuem Code.
2. Die Definition verlangt, dass der Liefermechanismus zuverlässig und wiederholbar ist. Wenn er oft ausfällt oder Fehler bei der Lieferung auftreten, kann dies die Qualität des bereitgestellten Systems beeinträchtigen, z. B. durch verringerte Verfügbarkeit oder Zuverlässigkeit.
3. Es gibt zwei wichtige Zeitpunkte im Entwicklungsprozess: das Einreichen von neuem Code und das Überführen des Codes in die Produktion. Nach der Überführung wird der Code getestet und überwacht. Nachdem er die Tests bestanden hat, wird er als Teil des normalen Produktionssystems betrachtet. Es gibt einen Unterschied zwischen dem Überführen von Code in die Produktion für Tests und Überwachung und dem anschließenden Hochstufen des neuen Codes.
4. Unsere Definition legt den Schwerpunkt auf das Ziel, die Zeit zwischen einem Commit von einem Entwickler und der Bereitstellung in der Produktion zu reduzieren. Die Form der Praktiken oder die Verwendung von Werkzeugen ist dabei nicht vorgegeben. Dies unterscheidet sich von anderen De-

definitionen, die den Fokus auf Kommunikation, Zusammenarbeit und Integration legen, ohne das Ziel zu nennen, oder die Verbindung zwischen DevOps und agilen Methoden oder die verwendeten Werkzeuge betonen, ohne das Ziel von DevOps-Praktiken, die Zeit oder die Qualität zu erwähnen.

- Die Ziele von DevOps-Praktiken beschränken sich nicht nur auf Tests und Bereitstellung. Es ist wichtig, eine Ops-Perspektive in die Anforderungserfassung einzubeziehen und die hohe Qualität des bereitgestellten Systems während seines gesamten Lebenszyklus sicherzustellen. Überwachungspraktiken sind ebenfalls einzubeziehen.

### 2.1.2 Phasen des DevOps Prozesses

Der DevOps-Prozess wird laut Len Bass (2015) in sechs Phasen unterteilt, wie in Abbildung 2.1 ersichtlich.

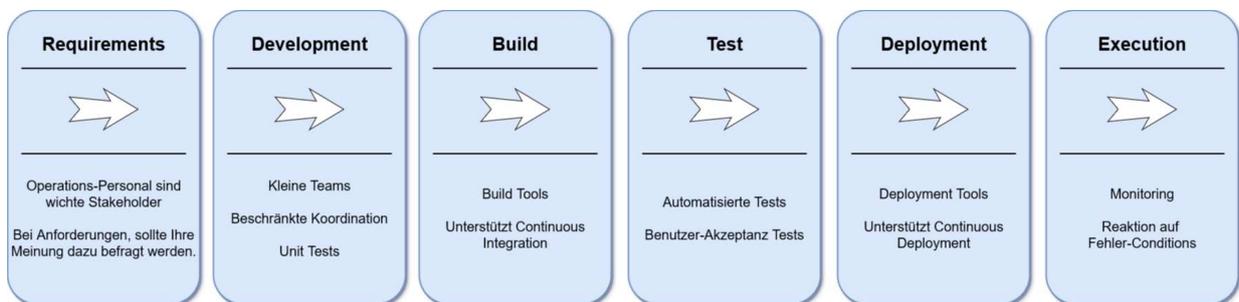


Abbildung 2.1: DevOps Phasen adaptiert aus Len Bass (2015)

Allerdings gibt es mehrere mögliche Bezeichnungen der Phasen, die Aktionen oder Prozesse hinter den Phasen sind dabei aber meist ähnlich oder gleich. Azad und Hyrynsalmi (2023) definiert wie in Abbildung 2.2 ersichtlich, sieben Phasen im DevOps-Prozess, wobei die Execution-Phase von Len Bass (2015) hier auf zwei Phasen aufgeteilt wurde. Des weiteren wurden die Phasen in sogenannte "Continuous Practices" unterteilt. Diese werden im Kapitel 2.6 näher erläutert.

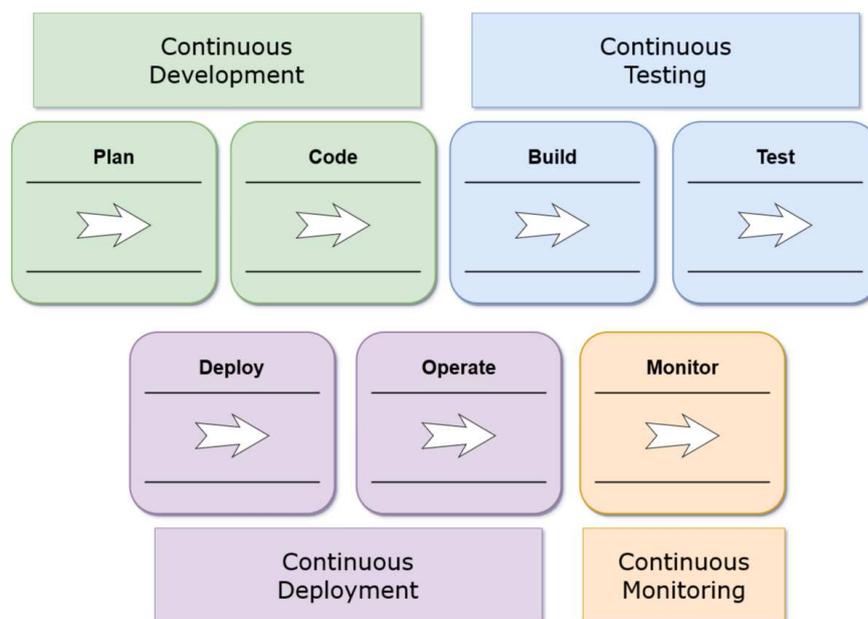


Abbildung 2.2: DevOps Phasen adaptiert aus Azad und Hyrynsalmi (2023)

### 2.1.3 Prinzipien und Werte

In diesem Kapitel werden die im DevOps Bereich definierten Prinzipien und Werte erklärt. Laut Srivastava (2023) helfen die DevOps Prinzipien dabei, den Prozess besser zu verstehen. Dabei gibt es sieben Prinzipien die in Srivastava (2023) näher erklärt werden. Zudem hebt Senapathi et al. (2018) ein Set von Werten hervor, welche in DevOps eine hohe Bedeutung haben. Diese werden im folgenden Abschnitt mit den passenden Prinzipien in Zusammenhang gebracht. Insgesamt zählt Senapathi et al. (2018) in seiner Arbeit folgende 14 Werte auf:

1. Qualifikationen arbeiten von Anfang an zusammen
2. Entwicklungs- und Ops-Rollen arbeiten zusammen
3. Eingebettete Ops
4. Teamverantwortung
5. Team-Fähigkeit
6. Verantwortung des Teams
7. Team-Kontrolle
8. Einfluss des Teams
9. Produktsicht
10. Entwurf, Implementierung und Wartung der Infrastruktur
11. Produktbereitstellung
12. Überwachung nach der Bereitstellung
13. Unterstützende Tools
14. Automatisierung ist eine Faustregel

### 2.1.3.1 Zusammenarbeit

Nach Srivastava (2023) betont DevOps die Zusammenarbeit und das Teamwork zwischen Entwicklungs- und Betriebsteams. Durch das Aufbrechen von Silos und die Förderung der Kommunikation können die Teams während des gesamten Entwicklungs- und Bereitstellungszyklus effektiv zusammenarbeiten.

#### Werte:

- Qualifikationen arbeiten von Anfang an zusammen
- Entwicklungs- und Ops-Rollen arbeiten zusammen
- Eingebettete Ops
- Teamverantwortung
- Team-Fähigkeit
- Verantwortung des Teams
- Team-Kontrolle
- Einfluss des Teams

### 2.1.3.2 Continuous Integration and Delivery

Laut Shahin et al. (2017) ist die kontinuierliche Integration (Continuous Integration, CI) eine Praxis, bei der laufende Arbeiten mehrmals am Tag integriert werden, um sicherzustellen, dass die von den Entwicklern vorgenommenen Codeänderungen häufig integriert und getestet werden. Auf diese Weise können Probleme frühzeitig im Entwicklungsprozess erkannt und behoben werden, wodurch das Risiko von Integrationsproblemen verringert und die Qualität der Software verbessert wird.

Bei der kontinuierlichen Bereitstellung (Continuous Delivery, CD) hingegen geht es nach Shahin et al. (2017) um die Fähigkeit, Software schnell und zuverlässig an Kunden zu liefern, indem die Automatisierung so weit wie möglich unterstützt wird. Dies beinhaltet die Automatisierung des gesamten Softwarebereitstellungsprozesses, von der Erstellung und dem Testen bis hin zur Bereitstellung und Freigabe, um sicherzustellen, dass Softwareänderungen jederzeit mit minimalem Risiko für die Produktion freigegeben werden können.

Shahin et al. (2017) schreibt, dass sowohl CI als auch CD sind Teil einer breiteren Palette von Praktiken, die als kontinuierliche Praktiken bekannt sind. Dazu gehört auch Continuous Deployment (CDep), eine Praxis, bei der Softwareänderungen automatisch für die Produktion bereitgestellt werden, sobald sie alle Tests und Prüfungen bestanden haben.

Dieses Prinzip wird in Kapitel 2.2 noch genauer erläutert.

#### Werte:

- Qualifikationen arbeiten von Anfang an zusammen
- Teamverantwortung
- Team-Fähigkeit
- Verantwortung des Teams

- Team-Kontrolle
- Einfluss des Teams
- Unterstützende Tools
- Automatisierung ist eine Faustregel

### 2.1.3.3 Kundenorientiertes Handeln

Altunel und Say (2021) zeigt auf, wie DevOps im Rahmen des SPSM-Modells kundenorientiert ist, indem agile, produktbasierte Transformation und DevOps-Prinzipien integriert werden. Er unterstreicht die zentrale Rolle des Produktmanagers bei der Sicherstellung des Produkterfolgs, der Festlegung der Vision und der Überwachung der Kundenzufriedenheit. Das Produktentwicklungs- und -betriebsteam spielt eine Schlüsselrolle bei der Bereitstellung und kontinuierlichen Verbesserung des Softwareprodukts unter Anwendung von DevOps-Praktiken. Der Ansatz stellt laut Altunel und Say (2021) den Kundennutzen in den Vordergrund und bezieht Metriken und Feedback während des gesamten Produktlebenszyklus ein.

#### Werte:

- Teamverantwortung
- Team-Fähigkeit
- Verantwortung des Teams
- Einfluss des Teams

### 2.1.3.4 End-to-End-Verantwortung

Nach Faber (2020) führt DevOps einen erweiterten Anwendungsbereich ein, indem Teams für die Gesamtqualität ihrer Dienste verantwortlich gemacht werden. Qualitätssicherungsaktivitäten sind integraler Bestandteil dieser erweiterten Verantwortung. Es ist vernünftig zu behaupten, dass alle Testaufgaben im DevOps-Team angesiedelt sein sollten. Im Idealfall sollten die durchgeführten Tests umfassend sein und den gesamten Umfang des Service des Teams abdecken. Dies kann jedoch eine Herausforderung sein, wenn ein Team einen Dienst mit mehreren miteinander verbundenen Diensten verwaltet. Die Interpretation von „End-to-End“ durch das Team kann sich von der des Kunden unterscheiden.

#### Werte:

- Teamverantwortung
- Verantwortung des Teams
- Team-Kontrolle

### 2.1.3.5 Funktionsübergreifende, autonome Teams

López-Fernández et al. (2022) untersucht die Autonomie innerhalb organisatorischer Teamstrukturen, insbesondere in DevOps-Produktteams. Die Autonomie wird in drei Stufen eingeteilt: hoch (keine externen Abhängigkeiten), mittel (Einsatzabhängigkeiten mit externer Genehmigung) und niedrig (viele Abhängigkeiten,

auch über den Einsatz hinaus). López-Fernández et al. (2022) hebt die Herausforderungen und Unterschiede beim Erreichen von Autonomie hervor und wirft dabei ein Licht auf die Entwicklung von DevOps-Praktiken in verschiedenen Organisationen.

**Werte:**

- Teamverantwortung
- Team-Fähigkeit
- Verantwortung des Teams
- Team-Kontrolle
- Einfluss des Teams

### 2.1.3.6 Kontinuierliche Verbesserung

Nach Ansicht von Dhakad (2023) sind kontinuierliche Tests und kontinuierliche Integration entscheidende Phasen für die Aufrechterhaltung konsistenter Verbesserungen im Code. Diese Verbesserungen werden in der Phase des kontinuierlichen Feedbacks bewertet, in der die Entwickler die Auswirkungen der Änderungen auf das Endprodukt beurteilen. Wichtig ist, dass Kundenfeedback eingeholt und umgehend umgesetzt wird, was letztlich zu positiven Reaktionen und der Veröffentlichung neuer Versionen von Softwareanwendungen führt.

**Werte:**

- Teamverantwortung
- Team-Fähigkeit
- Verantwortung des Teams
- Team-Kontrolle
- Einfluss des Teams

### 2.1.3.7 Automatisierung

Automatisierung ist nach Srivastava (2023) ein grundlegendes Prinzip bei DevOps. Sie beinhaltet die Automatisierung sich wiederholender und manueller Aufgaben, um die Effizienz zu steigern, Fehler zu Fehler zu reduzieren und eine schnellere und zuverlässigere Bereitstellung von Software zu ermöglichen. Dieses Prinzip wird in Kapitel 2.3 noch genauer erläutert.

Karamitsos et al. (2020) hebt die Rolle der Automatisierung bei der kontinuierlichen Bereitstellung hervor. Dazu gehört, dass die Entwickler den Code häufig übermitteln und ein Qualitätssicherungsteam automatisierte Tests durchführt. Automatisierungswerkzeuge helfen bei der Erkennung von Fehlern und Schwachstellen, und Versionskontrollsysteme sorgen für eine nahtlose Integration. Durch diese Automatisierung wird sichergestellt, dass die Code-Integration reibungslos verläuft und die mit umfangreichen Code-Unterschieden verbundenen Herausforderungen im Laufe der Zeit vermieden werden.

**Werte:**

- Automatisierung ist eine Faustregel

## 2.2 Continuous Integration und Continuous Delivery/Deployment

Laut Zampetti et al. (2021) zielt Continuous Integration (CI) auf die Automatisierung des Build-Prozesses ab, konkret auf dedizierte Server, mit dem Ziel der frühzeitigen Fehlerentdeckung sowie einer allgemeinen Qualitätsbewertung und Verbesserung. Aus diesen Gründen ist CI in der Industrie und im Open-Source-Bereich weit verbreitet.

Die Weiterentwicklung, Continuous Delivery/Deployment (CD), hilft dabei, kurze Veröffentlichungszyklen zu erreichen. Frühere empirische Untersuchungen haben gezeigt, wie CI/CD eine frühzeitige Entdeckung von Fehlern ermöglicht, die Produktivität der Entwickler erhöht und die Veröffentlichungszyklen beschleunigt.

### 2.2.1 Grundlagen

Wie in Kapitel 2.1.3 definiert, ist CI und CD ein Prinzip der DevOps-Methodik. Während DevOps als Methodik viele Werte und Prinzipien beinhaltet, ist CI/CD eines dieser Prinzipien, das diese Werte umsetzt und den DevOps-Prozess unterstützt.

### 2.2.2 Phasen

Die einzelnen Phasen von CI/CD bestehen aus den in Abbildung 2.3 ersichtlichen Punkten. Diese Abfolge an Aktivitäten wird CI/CD-Pipeline genannt. Der Start dieser Pipeline, "Build", liegt laut RedHat (2022) beim Entwickler-Team, welches ein neues Feature im Programm einbaut und die Software anschließend mit diesem Feature baut. Nachdem eine Version mit dem neuen Feature vorhanden ist, wird das Feature in dieser Version in der Phase "Test" getestet. Wenn das eingebaute Feature keine Probleme hervorruft und alle Anforderungen gedeckt sind, wird es in der Phase "Merge" in eine Test-Umgebung veröffentlicht. Sollten in dieser Umgebung keine Fehler durch manuelle und/oder automatisierte Tests auftreten, wird das Feature im "Continuous Delivery"-Schritt automatisch in eine vom Produktionssystem idente Umgebung übertragen. Sollten hier ebenfalls keine Fehler auftreten, wird das Feature für die Endbenutzer\*innen in der "Continuous Deployment"-Phase veröffentlicht.

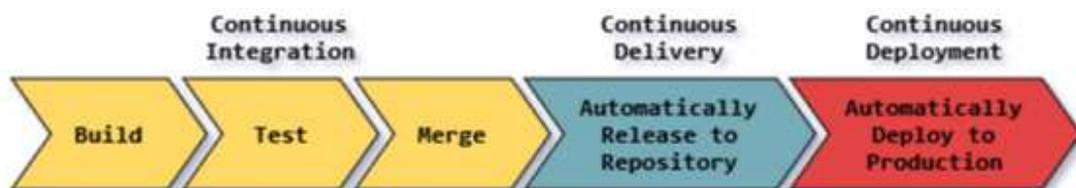


Abbildung 2.3: Vereinfachte CI/CD Pipeline adaptiert von RedHat (2022)

### 2.2.3 Ziele

Fowler und Foemmel (2006) definiert, dass das Ziel für CI ist, schnelles Feedback zu geben und sicherzustellen, dass eine stabile Software vorliegt, die ordnungsgemäß funktioniert und nur wenige Fehler enthält. Um das Ziel zu erreichen, gibt es laut Fowler und Foemmel (2006) mehrere Praktiken, die eine effektive CI ausmachen, darunter die Pflege eines einzigen Sourcecode-Repositorys, die Automatisierung des Build-Prozesses und das Testen in einem Klon der Produktionsumgebung. Außerdem ist es wichtig, den Commit-Build zu beschleunigen und sich von jemandem helfen zu lassen, der bereits Erfahrung mit Continuous Integration hat.

Farley und Humble (2010) definiert in seinem Buch das Ziel für Continuous Delivery: Softwareentwicklungsteams in die Lage zu versetzen, qualitativ hochwertige Software schnell, effizient und mit minimalem Risiko für die Produktion freizugeben. Erreicht wird dies laut Farley und Humble (2010) durch die Automatisierung des gesamten Softwarebereitstellungsprozesses, von der Erstellung und den Tests bis hin zur Bereitstellung und Freigabe, und durch die Gewährleistung, dass die Software stets in einem freigabefähigen Zustand ist. Letztlich geht es darum, den Kunden schneller und häufiger einen Mehrwert zu bieten und gleichzeitig die Zuverlässigkeit, Sicherheit und Wartbarkeit der Software zu verbessern.

### 2.2.4 Vorteile

Die Vorteile von CI sind laut Fowler und Foemmel (2006) folgende:

- **Schnelles Feedback:** Entwickler erhalten schnelles Feedback und können die Probleme daher schneller lösen, bevor sich eine größere Herausforderung daraus entwickelt.
- **Weniger Bugs:** Projekte mit CI tendieren dazu weniger drastische Bugs zu haben, sowohl im Prozess als auch in der Produktion.
- **Erhöhte Zusammenarbeit:** Durch die häufige Bereitstellung neuer Features tendieren Endbenutzer\*innen mehr mit den Entwickler zusammenzuarbeiten und Feedback, sowie Vorschläge zu den Features zu geben.
- **Erhöhtes Vertrauen:** CI erhöht das Vertrauen in das produzierte Software Produkt durch dauerhafte Tests und Integration.
- **Schnellere Time-to-Market:** CI sorgt für die Möglichkeit einer verkürzten Time-to-Market, da Features schneller bereitgestellt werden können.

Die Vorteile von CD hingegen sind laut Farley und Humble (2010) folgende:

- **Definierter Prozess:** Durch die CD wird ein konsistenter und zuverlässiger Prozess geschaffen, der zur Reduzierung von Fehlern und zur Steigerung der Effizienz beiträgt.
- **Kürzere Markteinführungszeit:** Durch die Verkürzung der Zykluszeit ermöglicht die CD eine schnellere Bereitstellung von Funktionen und Fehlerbehebungen für die Benutzer\*innen.

- **Gestärkte Teams:** Die CD-pipeline ermöglicht es Mitarbeiter\*innen, die gewünschte Version der Anwendung in der Umgebung ihrer Wahl selbst bereitzustellen, was zu kürzeren Wartezeiten und höherer Produktivität beitragen kann.
- **Geringeres Risiko:** Die CD reduziert die Anzahl der Änderungen zwischen den einzelnen Versionen, was zu einem geringeren Risiko bei der Veröffentlichung einer bestimmten Version führt.
- **Automatisierung:** Die CD erfordert die Automatisierung des gesamten Erstellungs-, Bereitstellungs-, Test- und Freigabeprozesses, was zur Reduzierung von Fehlern und zur Steigerung der Effizienz beitragen kann.
- **Umfassende Tests:** Die CD fördert die Erstellung eines umfassenden, zuverlässigen Satzes von automatisierten Tests und Systemtests, die in einer produktionsähnlichen Umgebung durchgeführt werden.
- **Kosteneinsparungen:** Allein die Kosteneinsparungen sind die zeitliche Investition wert, die die Einrichtung und Pflege eines solchen Freigabesystems mit sich bringt.

## 2.3 Automatisierung

Laut Srivastava (2023) spielt die Automatisierung eine entscheidende Rolle in der erfolgreichen Implementierung von DevOps. Sie umfasst die Ersetzung fehleranfälliger manueller Aufgaben durch automatisierte Prozesse, was DevOps-Teams ermöglicht, schnelles Feedback zu erhalten und Informationen über ihre Arbeitsabläufe zeitnah zu sammeln. Diese Automatisierung beschleunigt bestehende Prozesse, ermöglicht Teammitgliedern, sich auf aufgaben zu konzentrieren, die menschliches Eingreifen erfordern, und fördert die Nutzung der gleichen Tools während des gesamten Softwareentwicklungszyklus. Die Anpassung der Automatisierung an die Bedürfnisse und Ziele des Teams rationalisiert die Softwareentwicklung, -bereitstellung und -verwaltung und führt zu einer schnelleren Bereitstellung zuverlässiger Software bei geringerem Risiko. Die Automatisierung adressiert Herausforderungen wie manuelle Prozesse, Isolation, Doppelarbeit, Sicherheitsrisiken und Compliance-Probleme.

### 2.3.1 Test Automatisierung

Alégroth (2015) beschreibt in seiner Arbeit die Test-Automatisierung. Dabei wird beschrieben, dass Automatisierte Tests in der industriellen Praxis vor allem zwei Zwecken dienen: der Verbesserung der Softwarequalität und der Senkung der testbezogenen Kosten.

Erstens verbessert es die Softwarequalität, indem es eine schnellere Testausführung ermöglicht, eine schnellere Rückmeldung an die Entwickler erleichtert und eine frühzeitige Fehlererkennung ermöglicht, was letztlich die Entwicklungszeit verkürzt und die Ausbreitung von Fehlern verhindert. Automatisierte Unit-Tests sind in dieser Hinsicht besonders beliebt, da sie schnell ausgeführt werden können und in der agilen Softwareentwicklung effektiv sind. (Alégroth, 2015)

Zweitens werden die Testkosten gesenkt, auch wenn immer noch Kosten für die Anschaffung von Tools, Schulungen, organisatorische Änderungen und Pilotprojekte anfallen. Die Wartung von Testskripten ist ein laufender Kostenfaktor, der jedoch durch bewährte technische Verfahren gemindert werden kann. Die Durchführbarkeit automatisierter Tests hängt von der Bewertung ihrer Kosten, ihres Werts in Bezug auf die Fehlererkennung und ihres ROI im Vergleich zu alternativen Techniken wie manuellen Tests ab. Diese umfassende Bewertung ist für die Bestimmung der Anwendbarkeit und Durchführbarkeit der Testautomatisierung in der Praxis unerlässlich. (Alégroth, 2015)

### **2.3.2 Continuous Testing**

Hollier und Wagner (2017) definiert Continuous Testing als die Integration von Testautomatisierung in den Bereitstellungsprozess im Rahmen von DevOps und die Validierung von Software in realistischen Testumgebungen. Die Service-Virtualisierung spielt eine Rolle bei der Verlagerung des Testens nach links und ermöglicht es den Teams, die Softwarequalität früher zu bewerten, indem sie nicht verfügbare Komponenten simulieren. Dieser Ansatz erleichtert End-to-End-Tests, indem virtuelle Dienste anstelle von realen Diensten für Integrationstests während der Entwicklung verwendet werden. Die Kombination aus Testautomatisierung und Service-Virtualisierung bietet unmittelbares Feedback zur Softwarequalität und ermöglicht eine frühzeitige Problemlösung zu geringeren Kosten. Kontinuierliche Tests ermöglichen den Projektteams die Durchführung von Tests nach Bedarf, was genauere Liefervorhersagen und schnelle, marktreife und qualitativ hochwertige Lösungen gewährleistet.

## **2.4 Kontinuierliches Monitoring und Feedback**

Kontinuierliches Monitoring ist in DevOps nach Charak (2023) ein automatisierter Prozess, der Compliance-Probleme in verschiedenen Stadien des DevOps-Workflows aufdeckt. Dieser Ansatz, der auch als Continuous Control Monitoring (CCM) bezeichnet wird, geht über DevOps hinaus und kann überall dort angewendet werden, wo Aufmerksamkeit erforderlich ist. CCM liefert wichtige Daten für die Entscheidungsfindung, erleichtert die schnelle Fehlererkennung und liefert Feedback zu Problemen, so dass zeitnahe Korrekturmaßnahmen möglich sind. Kontinuierliches Monitoring ist nach Charak (2023) durch anpassungsfähige Tools möglich, die in verschiedenen Umgebungen funktionieren, einschließlich On-Premise-, Cloud- und Container-Setups, und ein kontinuierliches Monitoring der Systeme gewährleisten.

### **2.4.1 Monitoring von Anwendungen und Infrastruktur**

Im Kontext von DevOps beschreibt Thantharate (2023) die Überwachung und Beobachtbarkeit komplexer und verteilter Systeme eine große Herausforderung dar. Ein wichtiges Anliegen ist die Bewältigung der überwältigenden Datenmenge, die Erschöpfung der Warnmeldungen und das Erreichen einer umfassenden Transparenz. Komplexe Systeme erzeugen eine große Menge an Daten, so dass es schwierig ist, zu erkennen, was wichtig ist und was vernachlässigt werden kann. Darüber hinaus können schlecht konfigurierte Überwachungssysteme zu einem Übermaß an Warnmeldungen führen, darunter auch viele, die möglicherweise nicht aussagekräftig sind, was zu einer Ermüdung der Warnmeldungen und der potenziellen

Vernachlässigung wichtiger Warnmeldungen führt. Schließlich erweist es sich als schwierig, einen einheitlichen Überblick über den Zustand und die Leistung des Systems zu erhalten, insbesondere bei komplexen, verteilten Systemen. Die Bewältigung dieser Herausforderungen erfordert einen ganzheitlichen Ansatz mit effektiven Verfahren, effizienten Tools und einer Kultur der Kommunikation und Zusammenarbeit.

### 2.4.2 Bedeutung von Log- und Metrik-Analyse

Thantharate (2023) betont die Bedeutung von Log- und Metrikanalyse im Kontext von IntelligentMonitor, einem intelligenten Überwachungssystem für DevOps-Umgebungen. IntelligentMonitor nutzt maschinelles Lernen und Datenanalysetechniken, um Log- und Metrikdaten zu sammeln und zu analysieren, um wichtige Einblicke in die Systemleistung zu gewinnen. Die Analyse von Log-Dateien und Metriken ermöglicht es IntelligentMonitor, Datenüberlastung zu bewältigen, Alert-Fatigue zu reduzieren und die Sichtbarkeit des Systems zu verbessern. Durch die Überwachung von Metriken wie CPU-Auslastung, Speichernutzung und Netzwerkverkehr kann IntelligentMonitor Engpässe und Flaschenhalse identifizieren und die Leistung von Anwendungen und Systemen optimieren. Die Analyse von Log-Dateien ermöglicht es IntelligentMonitor, Probleme schnell zu identifizieren und zu beheben, indem sie Einblicke in die Ursachen von Fehlern und Problemen bietet. Insgesamt tragen Log- und Metrikanalyse dazu bei, die Qualität und Zuverlässigkeit von Softwareentwicklungs- und Bereitstellungsprozessen zu verbessern.

### 2.4.3 Feedbackschleifen

Im Kontext von DevOps beschreibt Knight (2020) Feedback-Schleifen als komplexe Mechanismen zur Erfassung, Verarbeitung und Verarbeitung von Informationen innerhalb des Softwareentwicklungsprozesses. Sie spielen laut Knight (2020) eine zentrale Rolle bei der Aufrechterhaltung von Effizienz und Geschwindigkeit, indem sie sicherstellen, dass alle Entwicklungsphasen, von der Idee bis zur Bereitstellung, synchronisiert und optimiert werden. Diese Schleifen ermöglichen es Softwareteams, ihre Prozesse anzupassen, zu automatisieren und kontinuierlich zu verbessern, was letztendlich zu einer schnelleren Softwarebereitstellung und verbesserter Produktqualität führt.

### 2.4.4 Key Performance Indicators (KPIs)

Thantharate (2023) beschreibt in seiner Arbeit 5 KPIs, also messbare Werte die beim Infrastruktur Monitoring eingesetzt werden können.

1. **CPU-Auslastung:** Prozentualer Anteil der CPU-Kapazität, der von jeder Komponente genutzt wird
2. **Speicher Nutzung:** Anteil des benutzten Speicher, von jeder Komponente.
3. **Netzwerktraffic:** Datentransfer im Netzwerk zwischen den Komponenten.
4. **Antwort-Zeit:** Zeit die pro Komponente benötigt wird um auf eine Anfrage zu antworten.
5. **Fehler-Rate:** Prozentsatz an in Komponenten ausgelösten Fehlern.

## 2.5 Sicherheit in DevOps

Im Zuge der Sicherheit in DevOps ist der Begriff 'DevSecOps' entstanden. Dieser beschreibt laut Srivastava und Bhosale (2023) die Praxis, Sicherheit von Anfang an in den DevOps-Prozess zu integrieren, anstatt sie als nachträglichen Gedanken zu behandeln.

### 2.5.1 Sicherheit als kontinuierlicher Bestandteil von DevOps

Srivastava und Bhosale (2023) behandelt in seiner Arbeit den Umfang von DevSecOps. Laut Srivastava und Bhosale (2023) umfasst es die Implementierung von Sicherheitspraktiken und -tools während des gesamten Lebenszyklus der Softwareentwicklung, vom Entwurf über die Entwicklung bis hin zu Test und Bereitstellung. Dazu gehören laut Srivastava und Bhosale (2023) der Einsatz automatisierter Sicherheitstests zur frühzeitigen Erkennung von Schwachstellen im Entwicklungsprozess, die Implementierung sicherer Codierungspraktiken und die Gewährleistung, dass die Sicherheit in jeder Phase des Prozesses berücksichtigt wird.

Durch die Integration der Sicherheit in den DevOps-Prozess können Unternehmen laut Srivastava und Bhosale (2023) die allgemeine Sicherheit ihrer Software verbessern und das Risiko von Sicherheitsverletzungen verringern. Dieser Ansatz trägt auch dazu bei, eine Sicherheitskultur innerhalb des Unternehmens zu schaffen, in der jeder für die Sicherheit verantwortlich ist und die Bedeutung der Sicherheit versteht.

### 2.5.2 DevSecOps Integration

Die Integration von DevSecOps ist ein integraler Bestandteil um fortlaufende Sicherheit im DevOps Prozess einzuhalten. DevSecOps deckt laut Srivastava und Bhosale (2023) ein breites Spektrum an Themen ab, die mit der Sicherheit im Softwareentwicklungsprozess zusammenhängen. Einige der wichtigsten Themen in DevSecOps sind, wie in Srivastava und Bhosale (2023) hervorgeht, folgende:

1. **Sichere Coding-Practices:** Dazu gehört das Schreiben von Code, der sicher und frei von Schwachstellen ist, wie z. B. SQL-Injection und Cross-Site-Scripting (XSS).
2. **Kontinuierliche Sicherheitstests:** Dazu gehört der Einsatz automatisierter Sicherheitstests, um Schwachstellen frühzeitig im Entwicklungsprozess zu erkennen.
3. **Zugriffskontrolle und Identitätsmanagement:** Dies beinhaltet die Implementierung geeigneter Zugangskontrollen und Identitätsmanagementpraktiken, um sicherzustellen, dass nur autorisierte Benutzer Zugang zu sensiblen Daten und Systemen haben.
4. **Compliance und Auditing:** Hier geht es darum, sicherzustellen, dass der Softwareentwicklungsprozess den gesetzlichen Anforderungen und Industriestandards entspricht.
5. **Sicherheit der Lieferkette:** Dies beinhaltet die Überprüfung von Drittanbietern und die Aufrechterhaltung einer genau definierten Lieferkette für Container und Bibliotheken.

6. **Sicherheit der Orchestrierung:** Dazu gehören die Implementierung sicherer Kubernetes-Konfigurationen und die Einhaltung bewährter Verfahren für die Zugriffskontrolle.
7. **Sicherheit von serverlosen Funktionen:** Dazu gehören die Förderung sicherer Codierungspraktiken und die Gewährleistung, dass serverlose Funktionen mit einer ordnungsgemäßen Eingabevalidierung und einem Schutz gegen gängige Angriffe wie Injection geschrieben werden.

### 2.5.3 Identifizierung und Behebung von Sicherheitslücken in DevOps

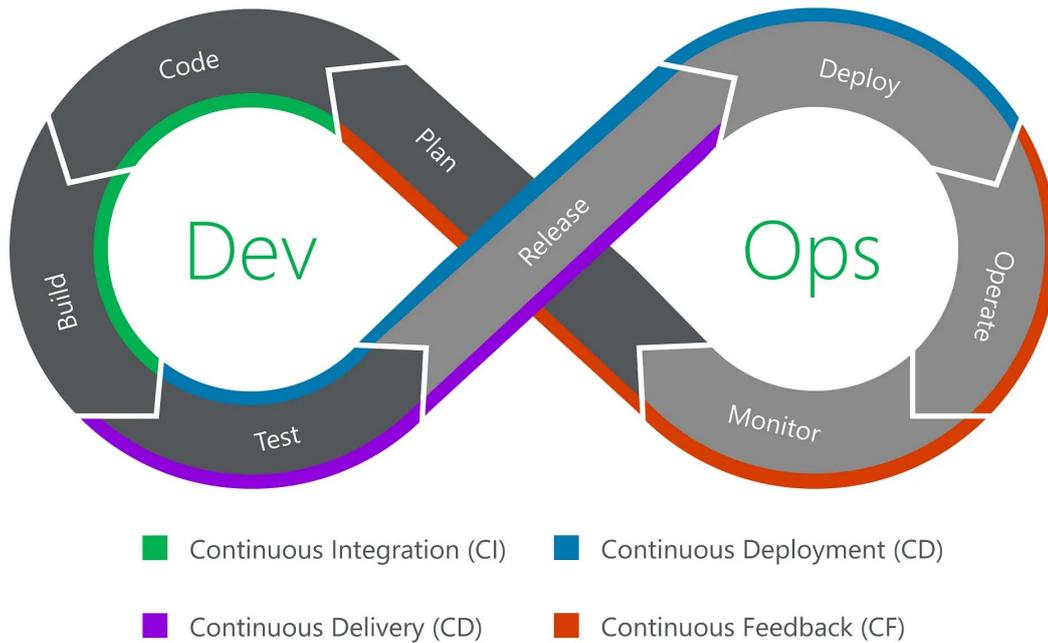
Wie sich aus Srivastava und Bhosale (2023) ableiten lässt, können folgende Schritte unternommen werden im Sicherheitslücken zu erkennen zu beheben, oder präventiv zu handeln:

1. **Bewerten von der aktuellen Sicherheitslage:** Durchführung einer gründliche Bewertung der derzeitigen Sicherheitspraktiken und Ermittlung von Bereichen, in denen die Sicherheit mangelhaft ist oder verbessert werden muss.
2. **Entwickeln einer DevSecOps-Strategie:** Entwickeln einer Strategie für die Integration von Sicherheit im DevOps-Prozess. Dabei sollten die Sicherheitstools und -praktiken, welche im Einsatz sind, sowie die Rollen und Verantwortlichkeiten der Teammitglieder festgelegt werden.
3. **Implementieren von Sicherheitstools und -praktiken:** Dies kann die Verwendung automatisierter Sicherheitstests, die Implementierung sicherer Kodierungsverfahren und die Sicherstellung, dass die Sicherheit in jeder Phase des Prozesses berücksichtigt wird, beinhalten.
4. **Teammitglieder schulen:** Informieren von Teammitgliedern über die Bedeutung der Sicherheit und über die Umsetzung sicherer Coding-Practices.
5. **Fortschritte überwachen und messen:** Überwachen der Fortschritte und messen der Wirksamkeit der DevSecOps-Strategie. Dies kann die Durchführung regelmäßiger Sicherheitsaudits und Konformitätsprüfungen beinhalten, um sicherzustellen, dass die Umgebung den gesetzlichen Anforderungen und Branchenstandards entspricht.
6. **Kontinuierliche Verbesserung:** Indem das Feedback von Teammitgliedern und Stakeholdern einbezogen wird und sich an die neuesten Sicherheitstrends und Best Practices gehalten wird.

## 2.6 Continuous Practices

In diesem Kapitel werden alle relevanten Continuous Practices und deren Phasen erklärt.

Die wichtigsten Practices und deren Phasen sind nach Pennington (2019) in Abbildung 2.4 ersichtlich. In der Literatur haben diese häufig unterschiedliche Bezeichnungen, beinhalten aber meist einen ähnlicher oder gleichen Prozess.



**Abbildung 2.4:** Acht CI/CD Phasen (Pennington, 2019)

## 2.6.1 Continuous Integration

Pittet (2023) beschreibt, bei der kontinuierlichen Integration führen die Entwickler die Änderungen häufig in die Versionskontrolle ein und validieren sie durch die Erstellung von Builds und die Durchführung automatisierter Tests. Dieser Ansatz verhindert, nach Pittet (2023), Integrationsprobleme, die auftreten können, wenn Änderungen erst am Tag der Veröffentlichung zusammengeführt werden. Die Testautomatisierung ist ein wichtiger Schwerpunkt, um sicherzustellen, dass die Anwendung bei jeder Integration neuer Commits in die Versionskontrolle funktionsfähig bleibt.

### 2.6.1.1 Plan

In der Planungsphase der Produktentwicklung werden Anforderungen und Feedback von Interessengruppen und Kunden gesammelt, um eine Produkt-Roadmap zu erstellen. Diese Roadmap dient als Leitfaden für die künftige Entwicklung und kann mithilfe von Ticket-Management-Systemen wie Jira, Azure DevOps oder Asana nachverfolgt werden. Die Roadmap ist in Epics, Features und User Stories unterteilt und bildet ein Backlog mit Aufgaben, die sich an den Kundenanforderungen orientieren. Diese Aufgaben werden dann für die Planung von Sprints und die Zuweisung von Aufgaben an das Team für die Entwicklung verwendet. (Pennington, 2019)

### 2.6.1.2 Commit

Im Zusammenhang mit der CI beinhaltet die Commit-Phase, dass die Entwickler ihre Arbeit häufig in ein gemeinsames Repository integrieren. Diese Phase ist entscheidend, da sie eine frühzeitige Erkennung

von Integrationsproblemen ermöglicht und den Aufwand für die Behebung von Fehlern in späteren Phasen verringert. (Baltes et al., 2018)

### **2.6.1.3 Build**

In der Build-Phase von CI geht es um die Umwandlung von Software-Artefakten (z. B. Quellcode) in Ergebnisse wie ausführbare Dateien und Bibliotheken. Diese Phase ist so konzipiert, dass sie häufig und schnell ausgeführt wird, um sicherzustellen, dass Änderungen an der Codebasis den Build nicht beeinträchtigen. (Ghaleb et al., 2019)

## **2.6.2 Continuous Delivery**

Continuous Delivery überschneidet sich wie in Abbildung 2.4 zu sehen ist, stark mit Continuous Deployment. Continuous Delivery teilt sich die meisten Phasen mit Continuous Deployment, bis auf die "Test" Phase.

### **2.6.2.1 Test**

Die Testphase in der CD umfasst effiziente Regressionstests, um die Softwarequalität zu gewährleisten, ohne die CI-Builds erheblich zu verzögern. Techniken wie Test Case Prioritization (TCP) werden eingesetzt, um die Ausführung von Testfällen zu priorisieren und Fehler so früh wie möglich zu erkennen. (Yaraghi et al., 2023)

## **2.6.3 Continuous Deployment**

In dieser Phase geht es um kontinuierliches Deployment. Ein Deployment ist laut Bryan (2021) ein wesentlicher Prozess, der darauf abzielt, ein Artefakt in eine Umgebung zu bringen. Je mehr ein Deployment-Prozess automatisiert werden kann, desto wiederholbarer, zuverlässiger und skalierbarer wird der Verteilungsprozess.

### **2.6.3.1 Release**

Die "Release"-Phase in DevOps ist die Phase, in der der "Ops"-Teil von DevOps beginnt. In der Release-Phase geht es in erster Linie darum, sicherzustellen, dass die Infrastruktur und die Umgebung für einen erfolgreichen Start der aktualisierten Software eingerichtet sind<sup>1</sup>. Dies kann das Hoch- oder Herunterfahren von Servern, die Aktualisierung von Betriebssystemkonfigurationen oder die Einrichtung einer anderen notwendigen Infrastruktur zur Unterstützung der Anwendung umfassen. Zu diesem Zeitpunkt sollte der aktualisierte Code gründlich getestet und bestätigt worden sein, dass er nach bestem Wissen und Gewissen des Teams funktioniert. (PlanetScale, 2023b)

### **2.6.3.2 Deploy**

In der "Deploy"-Phase von DevOps werden neue Änderungen in die Produktion eingeführt, was den Höhepunkt der bisherigen Arbeit darstellt. Das Betriebsteam kopiert getestete Release-Artefakte auf die Produktionsserver. Bei einer Blue/Green-Strategie wird der Datenverkehr auf den Staging-Server umgeleitet, und die aktuelle Produktionsumgebung bereitet den nächsten Anwendungszyklus vor. Offene Bereitstellungsanfragen sollten geschlossen und auf den Produktionszweig übertragen werden. Schemaänderungen und neuer Code sollten in die Produktion einfließen, und für den Fall katastrophaler Probleme beginnt ein 30-Minuten-Fenster für die Schema-Rückkehr. (PlanetScale, 2023a)

## **2.6.4 Continuous Feedback**

Nach Krancher et al. (2018) befürwortet Continuous Feedback einen Wechsel von seltenem und verzögertem Feedback zu einer unmittelbareren und kontinuierlichen Feedbackschleife, die Softwareentwicklungsteams dabei hilft, während des gesamten Entwicklungsprozesses effektiver zu lernen und sich anzupassen.

### **2.6.4.1 Operate**

In der Phase "Operate" von DevOps übernehmen DevOps-Teams Betriebsaufgaben, darunter die Entwicklung und den Betrieb der von ihnen benötigten Software. Dies fördert das Prinzip des "dogfooding" und ermöglicht einen Self-Service-Ansatz zur Entlastung von ITSM-Prozessen, wodurch Raum für agile Methoden geschaffen wird. (Halstenberg et al., 2020)

### **2.6.4.2 Monitor**

Die Phase "Monitoring" beinhaltet die Integration von Monitoring in die IT-Services, was dem DevOps-Team ermöglicht, Betriebsprobleme zu analysieren und Fehlerbehebungen durchzuführen. Das DevOps-Team betreibt und überwacht seine IT-Services und erhält wertvolles Feedback zu seiner Arbeit, bevorzugt dabei einfache Tools wie ein etabliertes Chattool („ChatOps“) gegenüber komplexen ITSM-Tools. (Halstenberg et al., 2020)

## **3 IT OPERATIONS**

Dieses Kapitel definiert IT Operations im Bezug auf diese Masterarbeit. Des weiteren wird der Bezug zu DevOps erklärt und Zusammenhänge mit der Forschungsfrage Richtung Automatisierung und Herausforderungen dargelegt.

### **3.1 Definition**

Nielsen et al. (2017) definiert IT Operations als ein Team, das für die Verwaltung und Wartung der IT-Infrastruktur und der Systeme zuständig ist, die den Geschäftsbetrieb eines Unternehmens unterstützen. Laut Nielsen et al. (2017) sind sie dafür verantwortlich, dass die Systeme verfügbar, zuverlässig und leistungsfähig sind und dass alle Probleme schnell gelöst werden, um Ausfallzeiten und Störungen des Geschäftsbetriebs zu minimieren. Im Zusammenhang mit der Arbeit Nielsen et al. (2017) ist der IT-Betrieb einer der wichtigsten Beteiligten am IT-Entwicklungsprozess, und die Zusammenarbeit zwischen Entwicklungs- und Betriebsteams ist für die Bereitstellung qualitativ hochwertiger Software, die den Anforderungen des Unternehmens entspricht, unerlässlich.

### **3.2 Infrastruktur**

Die Infrastruktur ist ein essenzieller Bestandteil des IT Operations. Die Aufgabe des IT-Betriebs liegt darin die Infrastruktur bereitzustellen. Diese sollte verfügbar, sicher, zuverlässig und konsistent sein.

Aebi (2004) definiert in seinem Buch eine Infrastruktur so, dass diese über einen längeren Zeitraum eine Vielzahl von Dienstleistungen für viele Menschen anbietet. Sie ist laut Aebi (2004) ein soziotechnisches System. In diesem Kontext bezieht sich IT-Infrastruktur auf die Gesamtheit von Informationstechnologie-Ressourcen in einem Unternehmen, einschließlich der benötigten personellen und materiellen Ressourcen für deren Betrieb. Gelegentlich wird dies auch als "Informatikmittel" bezeichnet, obwohl dieser Begriff weniger präzise ist.

Die Komponenten der Infrastruktur werden von Aebi (2004) in der Abbildung 3.1 dargestellt.

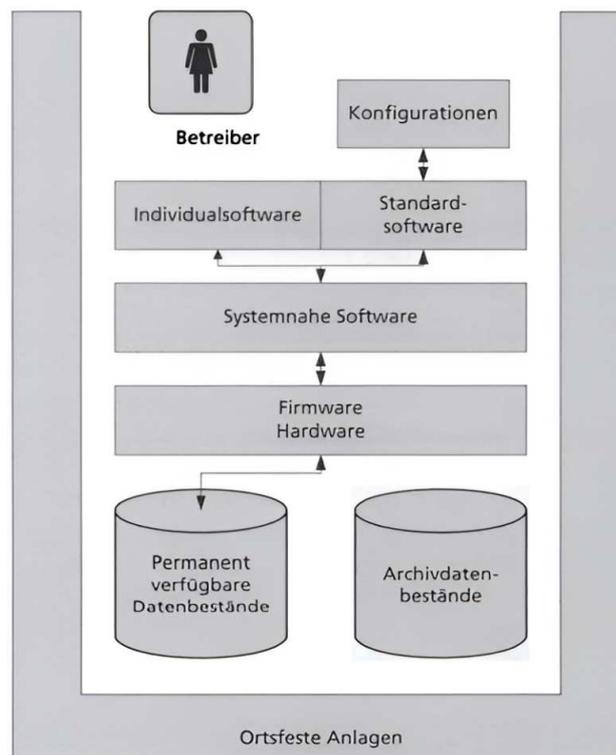


Abbildung 3.1: Komponenten der Infrastruktur nach Aebi (2004)

### 3.3 Prozesse und Praktiken

Prozesse und Praktiken überschneiden sich stark mit dem DevOps Prozess. Nielsen et al. (2017) erwähnt in seiner Arbeit mehrere Praktiken welche Bestandteil von ITOps sind. Einige davon sind wie folgt:

1. Verwendung von Best-Practice-Frameworks wie Capability Maturity Model Integration (CMMI) und IT Infrastructure Library (ITIL) zur Integration von Prozessen und zur Erleichterung der Kommunikation zwischen Teams.
2. Frühzeitige Einbindung von IT Operations in den Entwicklungsprozess, um betriebliche Überlegungen und Anforderungen zu berücksichtigen.
3. Einsatz von Virtualisierungstools wie Vagrant, um produktionsähnliche Entwicklungsumgebungen zu verwalten und das Verhalten, die Funktionalität und die Leistung des Produktionssystems zu simulieren.
4. Verwendung von Versionskontrollwerkzeugen wie Git oder Subversion zur Verwaltung von Codeänderungen und zur Synchronisierung von Umgebungen.
5. Verwendung von Konfigurationsmanagement-Tools wie Puppet zur Beschreibung und Ausführung des gewünschten Zustands einer Umgebung.
6. Einen hohen Fokus auf das Anforderungsmanagement durch enge Beziehungen zu den Benutzern, um deren Bedürfnisse zu ermitteln und schnell auf ihr Feedback zu reagieren.

### 3.3.1 ITIL Framework

Wie in der Aufzählung aus Kapitel 3.3 erwähnt ist der Einsatz von ITIL laut Nielsen et al. (2017) eine gängige Praxis.

Marrone und Kolbe (2011) definiert ITIL (Information Technology Infrastructure Library) als eine Sammlung von Best Practices für das IT-Service-Management (ITSM), die einen Rahmen für die Bereitstellung von IT-Services bietet. Marrone und Kolbe (2011) erklärt, dass ITIL wichtig ist, weil schätzungsweise 70 bis 80 % der Ausgaben einer IT-Organisation auf IT-Services entfallen und ITIL Organisationen dabei helfen kann, die Qualität der IT-Services zu verbessern, Kosten zu senken und die Kundenzufriedenheit zu erhöhen. In der Arbeit von Marrone und Kolbe (2011) wird auch darauf hingewiesen, dass ITIL das am weitesten verbreitete ITSM-Framework ist und dass viele Organisationen es übernommen haben, um ihre IT-Serviceleistungen zu verbessern.

### 3.3.2 DevOps Verbindung

Wie in der Aufzählung ersichtlich überschneiden sich einige Punkte mit den DevOps Praktiken. Das liegt daran, dass laut Nielsen et al. (2017) ITOps ein integraler Bestandteil von DevOps ist. DevOps vereint die Entwicklung von Software mit der Betriebnahme etwaiger Hardware, Netzwerken, etc. die dafür erforderlich sind, also "Development" mit "Operations".

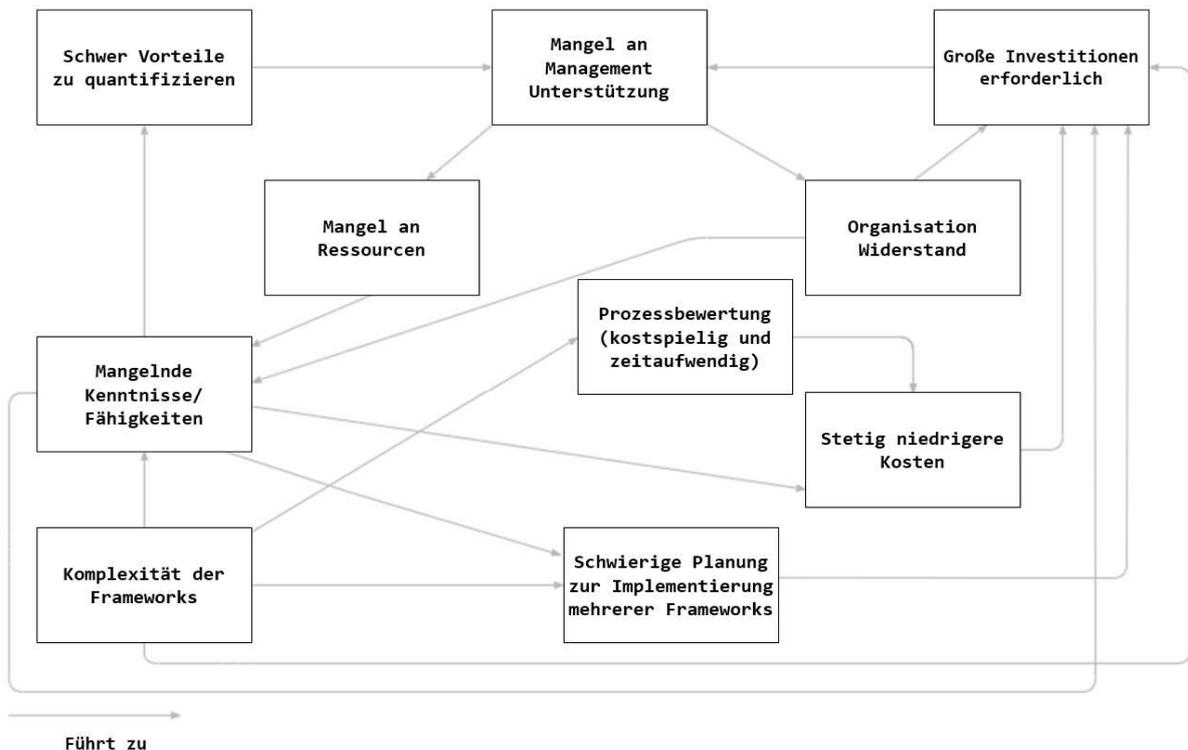
Nielsen et al. (2017) beschreibt die Schnittstelle zwischen ITOps und DevOps ist der Ort, an dem die beiden Sektionen oder Teams zusammenarbeiten, um sicherzustellen, dass die Software so entwickelt, getestet, bereitgestellt und gewartet wird, dass sie den Anforderungen des Unternehmens und der Endbenutzer entspricht und gleichzeitig zuverlässig, leistungsfähig und skalierbar ist.

## 3.4 Herausforderungen

Die Arbeit Serrano et al. (2021) beschreibt unter anderem die Probleme im IT Service Management (ITSM). Da ITSM ein Bestandteil von IT Operations ist, gelten die gefundenen Herausforderungen ebenfalls als Problem im IT Operations Bereich. Anhand einer Literatur Recherche kam die Arbeit zur Erkenntnis, dass folgende drei Punkte Herausforderungen darstellen.

1. **Organisatorischer Widerstand:** Der Widerstand der Mitarbeiter gegen Veränderungen ist die am häufigsten genannte Herausforderung, die auf die Organisationskultur zurückzuführen ist.
2. **Komplexität der Frameworks:** Die Komplexität von IT-Service-Management-Frameworks bezieht sich auf die Tatsache, dass diese Frameworks schwer zu verstehen und zu implementieren sein können, insbesondere für Organisationen, die nicht über angemessene Kenntnisse oder Fähigkeiten verfügen. In den Rahmenwerken wird kein Implementierungsprozess beschrieben, was die Einführung für Unternehmen schwierig machen kann.
3. **Fehlende Kenntnisse/Fähigkeiten:** Unternehmen haben bei der Implementierung von IT-Service-Management-Frameworks möglicherweise mit mangelnden Kenntnissen und Fähigkeiten zu kämpfen.

Desweiteren beschreibt Serrano et al. (2021) auch die Zusammenhänge und Auswirkungen der Herausforderungen. In Abbildung 3.2 sieht man etwa, dass viele Herausforderungen sich gegenseitig verschlimmern. Daher führt eine hohe Komplexität des Frameworks laut Serrano et al. (2021) zu einem Mangel an Skills. Da eine Person benötigt wird, welche sich bestenfalls gut mit dem zu implementierenden Framework auskennt, da dies eine zu hohe Komplexität mit sich bringt.



**Abbildung 3.2:** Zusammenhänge der ITSM Herausforderungen übersetzt aus Serrano et al. (2021)

## 3.5 Automatisierung

Die Automatisierung von IT Operations ist nicht nur für die Forschungsfrage relevant sondern auch in der Praxis ein wichtiges Thema. Die Arbeit von Brown und Hellerstein (2005) beschäftigt sich intensiv mit der Automatisierung im IT Operations Bereich und wurde bereits im Jahr 2005 veröffentlicht. Brown und Hellerstein (2005) beleuchtet kritisch das angebliche Kostenersparnis von Automatisierung und erklärt, dass Automatisierung nicht immer Kosten senkt. Er nennt hierfür die drei folgenden Argumente.

### 3.5.1 Weitere Prozesse

Die Einführung von Automatisierung führt zu zusätzlichen Prozessen, um diese Automatisierung zu implementieren und aufrechtzuerhalten, wie beim Vergleich manueller und automatisierter Softwareverteilungsprozesse. (Brown & Hellerstein, 2005)

### **3.5.2 Strukturierte Daten**

Automatisierung erfordert strukturierte Eingaben (z. B. Pakete für ein Softwareverteilungssystem), die zusätzliche anfängliche (feste) Kosten für Design, Implementierung und Tests im Vergleich zu manuellen Prozessen verursachen. Diese festen Kosten sind in IT-Operationen von großer Bedeutung, da viele Prozesse eine begrenzte Lebensdauer haben. Beispielsweise wird eine Software auf nur einer begrenzten Anzahl von Zielsystemen installiert. Tatsächlich ergaben Studien zur automatisierten Softwareverteilung in mehreren Rechenzentren, dass 25% der Softwarepakete auf weniger als 15 Servern installiert waren. Brown und Hellerstein, 2005

### **3.5.3 Komplexität der Fehlererkennung**

Das Erkennen und Beheben von Fehlern in einem automatisierten Prozess ist erheblich komplizierter als in einem manuellen Prozess. Die Daten zur Softwareverteilung deuten darauf hin, dass Fehler in der Automatisierung häufig auftreten - in den untersuchten Rechenzentren scheiterten 19% der angeforderten Installationen. (Brown & Hellerstein, 2005)

### **3.5.4 Einsatz von KI**

Andererseits erklärt Mohammad (2019) in seiner Arbeit, dass der Einsatz von KI zur Automatisierung einige Vorteile mit sich bringt. Mohammad (2019) erwähnt dabei folgende Vorteile:

1. Betriebliches Kostenmanagement
2. Steigerung der Produktivität
3. Gewährleistung hoher Verfügbarkeit
4. Erhöhung der Verlässlichkeit
5. Optimierung der Leistung

#### **3.5.4.1 Barrieren**

Zu den Hindernissen und Fallstricken, die es zu überwinden gilt, gehören laut Mohammad (2019) Fragen zu Geld, Zeit, Code und sogar Personal. Die Vorteile der Automatisierung, wie zum Beispiel die erhöhte Produktivität und Zuverlässigkeit, können Unternehmen jedoch davon überzeugen, mit den Automatisierungsaufgaben fortzufahren und verdrängen im Allgemeinen die Notwendigkeit Automatisierungsprojekte zu rechtfertigen.

## 4 KÜNSTLICHE INTELLIGENZ

In diesem Kapitel wird Künstliche Intelligenz und alle Grundlagen, welche zur Beantwortung der Forschungsfrage relevant sind, erklärt. Beginnend mit der Definition bis hin zum Zukunftsausblick.

### 4.1 Grundlagen

Die Grundlage der KI ist es laut Teich (2020) das Ziel, menschliche Intelligenz und menschliches Verhalten durch Algorithmen und Computerprogramme zu imitieren. Dabei werden verschiedene Techniken und Methoden eingesetzt, wie zum Beispiel maschinelles Lernen, neuronale Netze, Expertensysteme und Entscheidungsbäume.

#### 4.1.1 Definition

Der Begriff "Künstliche Intelligenz" (KI) wird von McCarthy et al. (1955) geprägt, der damit das Ziel verfolgte, jeden Aspekt von Lernen und andere Merkmale von Intelligenz so genau zu beschreiben, dass eine Maschine konstruiert werden kann, um sie nachzuahmen. Turing (1950) hat definiert, dass eine Maschine dann als KI gilt, wenn Menschen nicht erkennen können, ob es sich um einen Menschen oder eine Maschine handelt.

#### 4.1.2 KI-Methoden

Wie bereits in Kapitel 4.1 erwähnt, gibt es laut Teich (2020) mehrere Methoden der KI. Teich (2020) definiert dabei folgende Methoden:

1. **Maschinelles Lernen:** Die Umsetzung von Lernen durch Software, was auch der Ursprung der Beschäftigung mit KI ist.
2. **Neuronale Netze:** Ein Modell, das auf der Funktionsweise des menschlichen Gehirns basiert und zur Mustererkennung und Vorhersage eingesetzt wird.
3. **Expertensysteme:** Ein System, das auf dem Wissen von Experten basiert und zur Entscheidungsfindung eingesetzt wird.
4. **Entscheidungsbäume:** Ein Modell, das Entscheidungen auf der Grundlage von Ja/Nein-Fragen trifft.
5. **Natürliche Sprachverarbeitung:** Die Fähigkeit von Computern, menschliche Sprache zu verstehen und zu verarbeiten.

6. **Computer Vision:** Die Fähigkeit von Computern, Bilder und Videos zu verstehen und zu interpretieren.

### 4.1.3 Anwendungsbereiche von KI

Der Anwendungsbereich von KI ist enorm und wird sich in den nächsten Jahren ausbreiten. Bereits heute findet KI laut Pannu (2015) unter anderem Einzug in die Bereiche Wissenschaft, Technik, Wirtschaft, Medizin, Wettervorhersage, Buchhaltungsdatenbanken und Computerspiele. Pannu (2015) untersucht auch die Anwendung von KI bei Systemen zur Erkennung von Eindringlingen in Netzwerken, Stromsystemstabilisatoren und medizinische Bildklassifizierung.

## 4.2 Kategorien von KI

Wiles (2022) definiert in seinem Artikel vier Kategorien in die KI unterteilt werden kann. Diese werden im Anschluss genauer erklärt und definiert.

### 4.2.1 Datenzentrierte KI

Wiles (2022) schreibt in seinem Artikel, dass in der datenzentrierten KI der Fokus auf der Verbesserung der Daten für das Training von Algorithmen liegt, während traditionelle Ansätze sich auf die Optimierung der KI-Modelle selbst konzentrierten. Dies kann laut Wiles (2022) herkömmliches Datenmanagement stören, aber Organisationen können klassische Datenmanagement-Ideen erweitern, indem sie bequeme Funktionen für KI-Entwickler hinzufügen und KI zur Verbesserung der Data Governance, Persistenz, Integration und Datenqualität nutzen.

Des Weiteren führt Wiles (2022) an, dass Innovationsbereiche in der datenzentrierten KI synthetische Daten, Wissensgraphen, Datenbeschriftung und -kommentierung umfassen. Synthetische Daten sind künstlich erzeugte Daten, die die Verwendung personenbezogener Daten vermeiden und die Kosten sowie die Entwicklungszeit von Maschinellem Lernen (ML) reduzieren können, während sie die ML-Leistung verbessern. Wiles (2022) prognostiziert eine verstärkte Einführung von synthetischen Daten aufgrund dieser Vorteile.

### 4.2.2 Modellzentrierte KI

Trotz des datenzentrierten Ansatzes bleiben laut Wiles (2022) die Überwachung von KI-Modellen und Innovationen wie zusammengesetzte KI und kausale KI wichtig. Zusammengesetzte KI kombiniert verschiedene Techniken, um effizientes Lernen und erweiterte Wissensrepräsentation zu ermöglichen. Dies könnte in den nächsten zwei bis fünf Jahren verstärkt genutzt werden, wodurch neue Geschäftsmodelle und Branchendynamiken entstehen. Vorteile umfassen den Zugang zu KI für Organisationen ohne große Datensätze und die Erweiterung von KI-Anwendungsbereichen. Kausale KI nutzt Techniken wie Kausaldiagramme zur besseren Entscheidungsfindung, obwohl ihre breite Akzeptanz noch fünf bis zehn Jahre dauern könnte. Vorteile sind Effizienzsteigerung, größere Entscheidungsfreiheit und Stabilität in sich ändernden Umgebungen.

### **4.2.3 Anwendungszentrierte KI**

Innovationen im Bereich der Künstlichen Intelligenz umfassen nach Wiles (2022) AI-Engineering, Decision Intelligence, operative KI-Systeme, ModelOps, AI-Cloud-Services, intelligente Roboter, natürliche Sprachverarbeitung (NLP), autonome Fahrzeuge, intelligente Anwendungen und Computer Vision. Besonders hervorzuheben sind laut Wiles (2022) Decision Intelligence und Edge-AI, die voraussichtlich in den nächsten zwei bis fünf Jahren weit verbreitet sein werden und erhebliche Vorteile für Unternehmen bieten. Decision Intelligence zielt darauf ab, die Entscheidungsfindung zu optimieren, indem Entscheidungen transparenter, überprüfbarer und weniger unvorhersehbar werden. Edge-AI bezieht sich auf den Einsatz von KI in IoT-Geräten und bietet Unternehmen verbesserte Effizienz, Kundenzufriedenheit, geringere Latenzzeiten, niedrigere Konnektivitätskosten und eine zuverlässige Verfügbarkeit der Lösungen unabhängig von der Netzwerkkonnektivität.

### **4.2.4 Menschenzentrierte KI**

Zu den Innovationen gehören laut Wiles (2022) KI-Vertrauens-, Risiko- und Sicherheitsmanagement (TRISM), verantwortungsvolle KI, digitale Ethik und KI-Entwickler- und Schulungskits. Verantwortungsvolle KI ist entscheidend, um gute Ergebnisse bei der Unterstützung von Entscheidungen durch KI zu gewährleisten und Konflikte zwischen Werten und Risiken zu lösen. Sie umfasst nach Wiles (2022) verschiedene ethische und geschäftliche Aspekte wie Vertrauen, Transparenz, Fairness, Datenschutz und Compliance. Obwohl verantwortungsvolle KI noch einige Jahre bis zur weit verbreiteten Akzeptanz benötigt, wird sie langfristig die Wirtschaft beeinflussen. Ebenfalls in den nächsten Jahren relevant ist die digitale Ethik, die die Wertesysteme und moralischen Grundsätze in der elektronischen Interaktion betrifft und besonders im Kontext von Datenschutz und Voreingenommenheit relevant ist. Wiles (2022) sagt, dass Organisationen die digitale Ethik in ihre KI-Strategien integrieren, um Einfluss und Ansehen bei verschiedenen Interessengruppen zu stärken.

## **4.3 Ethik und Verantwortung**

Hermansyah et al. (2023) schreibt über den dynamischen Bereich der Ethik von KI, als Reaktion auf den zunehmenden Einsatz von KI in verschiedenen Sektoren rasch weiterentwickelt. Hermansyah et al. (2023) zeigt zwei Hauptperspektiven innerhalb der KI-Ethik auf.

Die eine dreht sich um die menschliche Interaktion mit KI und umfasst Fragen wie Datenschutz, Datennutzung und Verantwortung der Nutzer (Roboterethik).

Die zweite Perspektive konzentriert sich auf ethische Erwägungen, die das Verhalten der KI selbst betreffen, und befasst sich mit Fragen wie der Verhinderung von Diskriminierung, dem Schutz der Umwelt und der Vermeidung des Potenzials der KI, die menschliche Intelligenz zu übertreffen (Maschinenethik).

Beide Perspektiven unterstreichen laut Hermansyah et al. (2023) die Notwendigkeit der Zusammenarbeit zwischen Ethikern und Technologieexperten während des gesamten KI-Entwicklungsprozesses. Durch diese Zusammenarbeit wird sichergestellt, dass KI-Systeme ethische und moralische Standards einhalten und

das Risiko negativer Auswirkungen auf Mensch und Umwelt minimiert wird. Letztlich betont Hermansyah et al. (2023), dass eine verantwortungsvolle KI-Entwicklung sowohl die Roboter- als auch die Maschinenethik umfassen sollte, um sicherzustellen, dass die KI-Technologie mit ethischen Grundsätzen übereinstimmt und weder den Menschen noch dem Ökosystem schadet.

### **4.3.1 Herausforderungen und Bedenken im Umgang mit KI**

Durch die wachsende Erwartungshaltung der KI, siehe Abbildung 4.1, entstehen stets neue Bedenken und Herausforderungen. Volkmar et al. (2021) fasst dabei folgende KI-Herausforderungen in einer Grafik zusammen:

1. Akzeptanz
2. Daten
3. Ethik
4. Umsetzung
5. Interaktion
6. Wissen
7. Strategie
8. Unternehmenskultur
9. Vertrauen

## **4.4 Zukunftsausblick und Forschungstrends in der KI**

In diesem Unterkapitel geht es um die Erwartungen und möglichen zukünftige Entwicklungen von KI. Desweiteren geht es um Herausforderungen und Chancen in der Zukunft mit KI-Systemen.

### **4.4.1 Aktuelle Entwicklungen und Trends in der KI**

KI ist derzeit ein weiterverbreitetes Thema und erzeugt derzeit viele Innovationsmöglichkeiten. Wie in Abbildung 4.1 gezeigt, sind viele Themen im Bereich KI laut Perri (2023) noch am Anfang, einige fortgeschrittener und ein paar wenige steigen Richtung Erwartungs-Plateau. Laut Perri (2023) sind Themen wie Neuro-Symbolische KI Multiagentensysteme, First-Principle-KI, Automatische Systeme und viele mehr, noch am Anfang ihrer Erwartungshaltung und sind sogenannte Innovationsauslöser. Das Thema KI Allgemein wird von Perri (2023) genau zwischen den zwei Stufen "Innovationsauslöser" und "Gipfel der überzogenen Erwartung", was auf eine enorme Steigerung der Erwartungshaltung im Vergleich zum Vorjahr hinweist Wiles (2022). Das Thema "Computer Vision" zum Beispiel ist allerdings schon fast am "Plateau der Produktivität", obwohl dieses erst weniger als 2 Jahre alt ist.

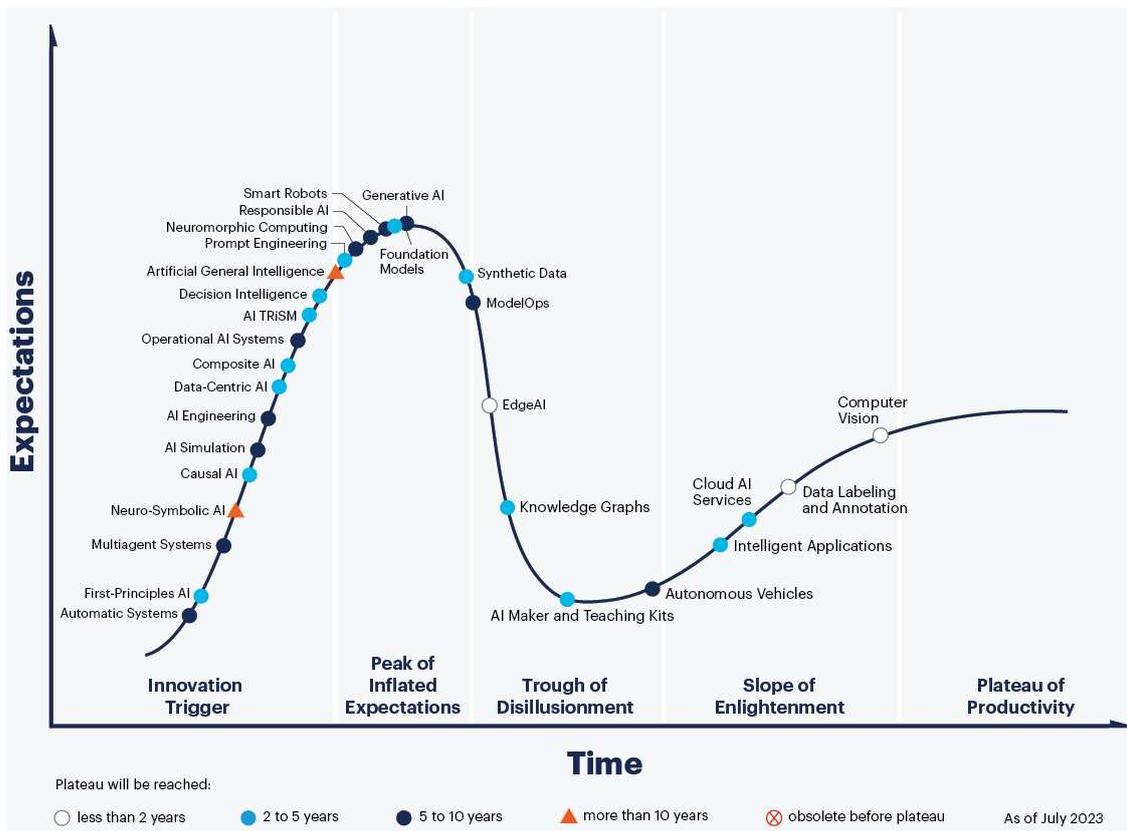


Abbildung 4.1: HypeCycle für KI (Wiles, 2022)

#### 4.4.2 Herausforderungen und Chancen für die Zukunft der KI

Laut Bharadiya (2023) gehören zu den Chancen eine bessere Entscheidungsfindung, ein besseres Kundenerlebnis und eine höhere betriebliche Effizienz. Jedoch wird von Bharadiya (2023) auch darauf hingewiesen, dass es einige Herausforderungen gibt, die adressiert werden müssen. Einerseits sind das laut Bharadiya (2023) Datenschutz- und Sicherheitsbedenken, der Bedarf an speziellen Fähigkeiten und Fachwissen und das Potenzial für Verzerrungen bei der algorithmischen Entscheidungsfindung. Insgesamt empfiehlt Bharadiya (2023), dass Unternehmen diese Herausforderungen und Chancen sorgfältig abwägen sollten, wenn sie das Potenzial von maschinellem Lernen und KI in ihrem Betrieb erkunden.

Laut Bharadiya (2023) ist der Datenschutz ein Problem, wenn es um die Integration von maschinellem Lernen und KI in die Business Intelligence geht. Dies liegt daran, dass KI-Modelle auf riesige Datenmengen angewiesen sind, die oft sensible und persönliche Informationen enthalten. Ethische Überlegungen unterstreichen die Bedeutung des Schutzes der Privatsphäre der Nutzer und der Einhaltung von Datenschutzbestimmungen. Unternehmen müssen robuste Sicherheitsmaßnahmen, Datenanonymisierungstechniken und sichere Datenverarbeitungspraktiken implementieren, um Benutzerinformationen zu schützen. Andernfalls kann es laut Bharadiya (2023) zu Verletzungen der Privatsphäre und zum Verlust des Vertrauens der Kunden kommen, was erhebliche negative Auswirkungen auf den Ruf und den Gewinn eines Unternehmens haben kann.

#### **4.4.3 Ausblick auf mögliche Anwendungen und Forschungsrichtungen in der KI**

Rathore (2023) schreibt in seiner Arbeit, dass die Aussichten für potenzielle Anwendungen und Forschungsrichtungen im Bereich der KI vielversprechend sind. Laut Rathore (2023) haben KI-Tools wie ChatGPT das Potenzial, verschiedene Bereiche zu revolutionieren, darunter Bildung, Kundendienst und akademisches Schreiben. Im Bildungsbereich kann ChatGPT interaktives Lernen erleichtern, die Zusammenarbeit zwischen Schülern und Lehrern unterstützen und personalisierte Lehrmaterialien bereitstellen. Im Kundendienst kann ChatGPT die Reaktionszeiten und die Kundenzufriedenheit verbessern. Darüber hinaus können KI-Tools in verschiedenen Branchen eingesetzt werden, um Aufgaben zu automatisieren, intelligente Entscheidungen zu treffen und Erkenntnisse aus Daten zu gewinnen. In dem Dokument wird auch das Potenzial von KI für die prädiktive Analyse erwähnt, bei der KI-Algorithmen zur Vorhersage von Ergebnissen und zur Abgabe genauerer Empfehlungen eingesetzt werden können. Insgesamt hebt das Dokument die potenziellen Vorteile und Möglichkeiten von KI-Tools hervor und unterstreicht die Notwendigkeit weiterer Forschung und Erkundung in diesem Bereich. (Rathore, 2023)

## 5 AIOPS

Dieses Kapitel befasst sich mit der Definition und den Bestandteilen von AIOps. Im weiteren Verlauf wird auf die Definition, die unterschiedlichen Ausprägungen, die Vorteile und Herausforderungen eingegangen.

### 5.1 Definition

Moore (2019) definiert AIOps wie folgt. AIOps verwendet maschinelles Lernen (ML) und Datenwissenschaft an, um die Herausforderungen des IT-Betriebs zu bewältigen. AIOps-Plattformen integrieren Big-Data- und ML-Funktionen, um wichtige Aufgaben des IT-Betriebs zu verbessern und teilweise zu ersetzen, zum Beispiel die Überwachung von Verfügbarkeit und Leistung, die Analyse von Ereignissen und die Verwaltung von IT-Diensten bei gleichzeitiger Automatisierung von Prozessen. Diese Plattformen nehmen die ständig wachsenden und vielfältigen IT-Daten auf, analysieren sie und stellen sie in wertvoller Weise dar.

### 5.2 Vorteile und Herausforderungen

In Artikel von Y. Li, Jiang, Li, Hassan et al. (2020) werden mehrere Herausforderungen im Zusammenhang mit AIOps-Lösungen genannt. Allgemein wird in der Studie definiert, dass die erfolgreiche Einführung von AIOps-Lösungen mehr als nur ein leistungsfähiges ML-Modell erfordert. AIOps-Lösungen müssen laut Y. Li, Jiang, Li, Hassan et al. (2020) auch vertrauenswürdig, interpretierbar, wartbar, skalierbar und im Kontext evaluierbar sein. Darüber hinaus müssen AIOps-Lösungen jahrelanges, in der Praxis erprobtes und von Ingenieuren erprobtes Fachwissen in ihre ML-Modelle einfließen lassen, interpretierbar sein und minimale Wartung und Feinabstimmung erfordern. In der Studie geht es darum, die Fehlerrate von Nodes in einem Netzwerk besser vorherzusagen über ein Machine Learning Modell. Auf diesen Kontext bezogen, gab es auch einige Herausforderungen bezogen auf die extremen Mengen an Daten und des komplexen Fehlerbilds, dieser ausfallenden Nodes.

AIOps bringt viele Vorteile mit sich, darunter beschreibt Linthicum (2021) folgende Punkte:

1. **Geschäftliche Agilität:** AIOps verbessert die geschäftliche Agilität für Skalierbarkeit und Anpassungsfähigkeit.
2. **Verlängerte Hardware- und Cloud-Lebensdauer:** AIOps reduziert die Häufigkeit des Austauschs von Hardware und Cloud-Services.
3. **Kostenvorhersage:** AIOps sagt die Betriebskosten voraus und passt sie an die Budgetvorgaben an.

4. **Einhaltung gesetzlicher Vorschriften:** AIOps stellt die Einhaltung von Vorschriften sicher und verhindert so Geldstrafen.

Kerravala (2021) fügt dieser Liste noch weitere folgende Punkte hinzu:

1. **Kostenreduzierung:** AIOps senkt die Kosten durch Automatisierung und effiziente Ressourcenzuweisung.
2. **Netzwerk-Effizienz:** Durch die Integration von AIOps mit SD-WAN wird die Netzwerkverwaltung zentralisiert.
3. **IT-Transformation:** AIOps ermöglicht Unternehmen den Übergang zu selbstheilenden und selbstverwalteten Netzwerken.
4. **SASE-Integration:** AIOps lässt sich gut in eine SASE-Architektur für moderne Unternehmensnetzwerke integrieren.
5. **Cloud-Migration:** AIOps hilft bei der digitalen Transformation von Netzwerken während der Cloud-Migration.

Die Allgemeinen Challenges werden von Y. Li, Jiang, Li, Hassan et al. (2020) wie folgt definiert:

1. **Vertrauenswürdigkeit:** AIOps-Lösungen müssen jahrelanges, in der Praxis erprobtes Fachwissen von Ingenieuren in ihre ML-Modelle einfließen lassen, anstatt einfach hochentwickelte Modelle auf Rohdaten anzuwenden.
2. **Auslegbarkeit:** AIOps-Lösungen müssen interpretierbar sein, auch wenn dies auf Kosten der Leistung geht. Solche interpretierbaren Modelle ermöglichen es DevOps-Ingenieuren, über Modellempfehlungen nachzudenken, die Unterstützung des oberen Managements gewinnen und, was noch wichtiger ist, DevOps-Ingenieure in die Lage versetzen, den Status Quo zu verbessern (z. B. durch Verbesserung und Optimierung ihrer Überwachungslösungen).
3. **Wartbarkeit:** AIOps-Lösungen sollten minimale Wartung und Feinabstimmung erfordern, da DevOps-Ingenieure in der Regel keine ML-Experten sind und oft bereits durch zahlreiche unternehmensweite ML-Initiativen belastet sind.
4. **Skalierbarkeit:** AIOps-Lösungen müssen skalierbar und effizient sein, da sie die Überwachungsdaten von Tausenden bis Millionen von Knoten analysieren müssen.
5. **Bewertung:** AIOps-Lösungen müssen in einem Kontext bewertet werden, der dem tatsächlichen Produktionseinsatz ähnelt. Herkömmliche ML-Bewertungstechniken (z. B. Kreuzvalidierung) sind selten anwendbar, da sie die Besonderheiten des realen Lebens nicht berücksichtigen.

## 5.3 Techniken

In diesem Kapitel wird auf Techniken der AIOps-Integration eingegangen, um eine Grundlage für den weiteren Verlauf und den praktischen Teil der Arbeit zu schaffen.

Es gibt viele Möglichkeiten, die Methoden des Einsatzes von AIOps zu kategorisieren. Aus der Studie Dang et al. (2019) lassen sich fünf Möglichkeiten für den Einsatz von AIOps ableiten.

1. **Datenintegration:** AIOps nutzt Daten aus unterschiedlichen Quellen wie Protokolle, Metriken und Ereignisse, um Systemleistung und Probleme zu überwachen.
2. **Auto Triggers:** AIOps automatisiert Aktionen basierend auf Regeln, um menschliche Eingriffe in IT-Operationen zu verringern und die Effizienz zu steigern.
3. **Predictive ALM:** Diese Methode beschleunigt die Softwarebereitstellung und verbessert Qualität und Sicherheit durch Datenintegration und vorausschauendes Agile Lifecycle Management.
4. **Anomaly Detection:** Hier werden ungewöhnliche Muster in Systemdaten erkannt, um Probleme wie Sicherheitsbedrohungen frühzeitig aufzudecken.
5. **Root Cause Analysis:** Diese Technik hilft, die Ursache von Problemen durch die Analyse von Systemdaten zu ermitteln, was zu schnellerer Problemlösung und erhöhter Dienstverfügbarkeit führt.

## **6 METHODIK**

### **6.1 Wahl der Methode**

Die systematische Literaturrecherche bietet die Möglichkeit eine Vielzahl an Literatur und deren Daten zu analysieren und eine Auswertung über gewisse Inhalte der Arbeiten zu generieren. Dies verschafft die Möglichkeit, Forschungsfelder zu analysieren und einen Überblick zu schaffen. Im Falle der Forschungsfrage bietet sie Möglichkeit, die verschiedenen Integrationsarten zu erfassen und bezüglich Häufigkeit, Verbreitung und Herausforderungen analysieren zu können.

### **6.2 Arten**

Bei der Literaturrecherche gibt es zwei vertretene Arten, zwischen denen unterschieden werden muss. Diese zwei Arten werden folgend erklärt und sich für eine Art entschieden.

#### **6.2.1 Unsystematische Literaturrecherche**

Eine unsystematische Literaturrecherche, auch bekannt als unsystematische oder zielgerichtete Literaturrecherche, ist eine Art von Literaturrecherche, die Flexibilität bietet, um weitergehende Fragen zu behandeln und neue Erkenntnisse zu gewinnen, jedoch wenig Gewähr für eine ausgewogene Perspektive auf das Thema bietet (Cook, 2019). Sie unterscheidet sich von einer systematischen Literaturübersicht, bei der strenge Methoden zur Auswahl von Artikeln und zur Datenextraktion angewandt werden, um ein konzentriertes, tiefgehendes Licht auf einen relativ engen Forschungsbereich zu werfen (Cook, 2019).

#### **6.2.2 Systematische Literaturrecherche**

Eine systematische Überprüfung ist eine methodisch strenge Forschungsmethode, die dazu dient, die vorhandene Literatur zu einem spezifischen Problem zu organisieren und zu bewerten (Hanson-Abromeit, 2014). Sie liefert im Vergleich zu einer herkömmlichen Literaturrecherche ein umfassenderes und präziseres Verständnis einer Forschungsfrage oder eines Themas von Interesse, indem sie standardisierte Methoden und Richtlinien für die Suche, Filterung, Überprüfung, Kritik, Interpretation, Synthese und Berichterstattung von Ergebnissen aus mehreren Veröffentlichungen anwendet (Pati, 2018).

## 6.3 Systematische Literaturrecherche

Wie in Abbildung 6.1 ersichtlich umfasst die systematische Literaturrecherche fünf Schritte, die durchlaufen werden müssen:

- Definition des Überprüfungsrahmens
- Konzeptionalisierung des Themas
- Literatursuche
- Literaturanalyse und -synthese
- Forschungsplan

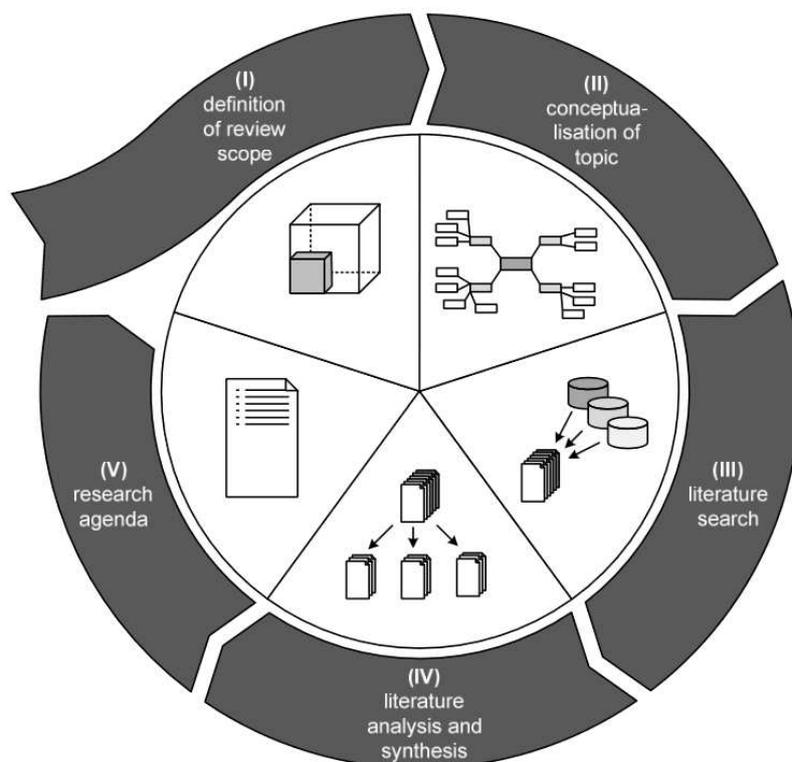


Abbildung 6.1: 5 Phasen systematische Literaturrecherche (Brocke et al., 2009)

### 6.3.1 Anwendungsbereich

Brocke et al. (2009) schlägt in seiner Arbeit vor, die Cooper'sche Taxonomie zu verwenden, um den Umfang einer Literaturübersicht zu definieren. Coopers Taxonomie hat sechs Merkmale, darunter Fokus, Ziel und Organisation. Der Fokus einer Literaturübersicht ist das, was für den Rezensenten am wichtigsten ist, und das Ziel ist es, die Ergebnisse zusammenzufassen, zu kritisieren oder zu integrieren. Des weiteren führt Brocke et al. (2009) die Punkte Perspektive, Zielgruppe und Umfang auf. Die Perspektive einer Rezension spiegelt wider, ob eine bestimmte Position vertreten wird oder nicht. Die Zielgruppe bestimmt insbesondere die Zielgruppe der Forschung, welche davon einen Nutzen tragen. Der Umfang der Literatur zu einem Thema beschreibt in welchem Ausmaß ein Bereich abgedeckt wird.

Brocke et al. (2009) definiert in seiner Arbeit in Abbildung 6.2 ersichtliche Auswahl-Möglichkeit. Diese werden in Kapitel 6.4.2 genauer definiert und erläutert.

Characteristic		Categories			
(1)	focus	research outcomes	research methods	theories	applications
(2)	goal	integration	criticism		central issues
(3)	organisation	historical	conceptual		methodological
(4)	perspective	neutral representation		espousal of position	
(5)	audience	specialised scholars	general scholars	practitioners/politicians	general public
(6)	coverage	exhaustive	exhaustive and selective	representative	central/pivotal

**Abbildung 6.2:** Charakteristik - Auswahl (Brocke et al., 2009)

### 6.3.2 Konzeptualisierung

Beim zweiten Kapitel "Konzeptualisierung des Themas" geht es laut Brocke et al. (2009) darum, eine umfassende Vorstellung davon zu vermitteln, was über das Thema bekannt ist und in welchen Bereichen möglicherweise noch Wissen benötigt wird, um mit einer Literatursuche zu beginnen.

In der von von Brocke et al. (2009) beschriebenen Phase der "Konzeptualisierung des Themas" ist es entscheidend, die Forschungsfrage zu präzisieren und relevante Schlüsselbegriffe zu identifizieren. Mit diesen Begriffen wird dann eine systematische Suche in den ausgewählten Datenbanken durchgeführt. Es ist wichtig, die Suchstrategie klar zu definieren und zu dokumentieren, um die Nachvollziehbarkeit und Wiederholbarkeit der Recherche zu gewährleisten. Außerdem sollte in dieser Phase eine vorläufige Struktur oder ein Rahmen für die Analyse und Synthese der Literatur entwickelt werden. Dieser Rahmen kann sich im Laufe des Forschungsprozesses weiterentwickeln und anpassen, bietet aber eine grundlegende Basis für eine systematische Literaturlauswertung.

### 6.3.3 Literatursuche

Beim dritten Kapitel "Literatursuche" geht es laut Brocke et al. (2009) um den Prozess der Suche nach wissenschaftlicher Literatur zu einem bestimmten Thema, der die Abfrage wissenschaftlicher Datenbanken mit Hilfe von Schlüsselwörtern, die Rückwärts- oder Vorwärtssuche auf der Grundlage relevanter Artikel und die fortlaufende Auswertung von Quellen umfasst. Wichtig dabei ist auch, dass die Datenbanken und Schlüsselwörter transparent festgelegt und definiert werden. Der Ablauf der Literatursuche ist in Abbildung 6.3 ersichtlich und wird in einer adaptierten Form durchgeführt. Die Adaption ist die Erweiterung durch den Schritt, dass bei der manuellen Suche die Quellen bereits gefundener Arbeiten ebenfalls nach nützlichen Quellen durchsucht werden.

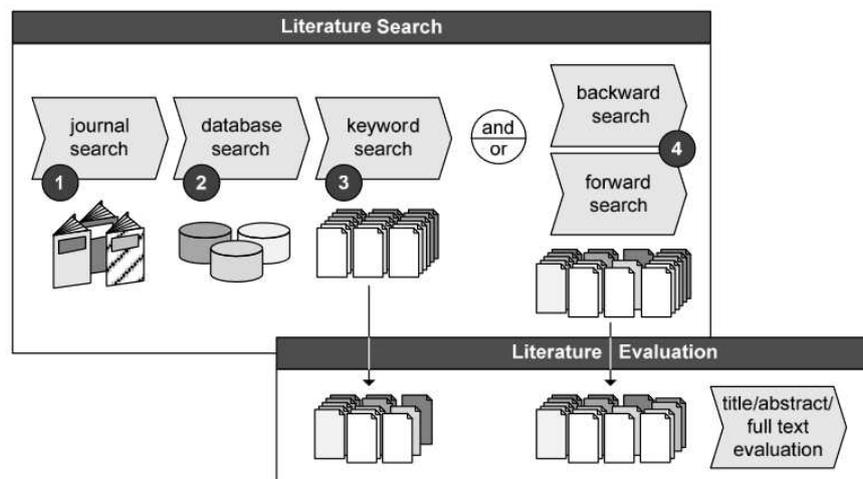


Abbildung 6.3: Prozess der Literatursuche (Brocke et al., 2009)

### 6.3.4 Literaturanalyse und -synthese

Laut Brocke et al. (2009) geht es im vierten Kapitels "Literaturanalyse und -synthese" um den Prozess der Überprüfung und Zusammenfassung vorhandener Literatur zu einem bestimmten Thema, einschließlich der Identifizierung von Schlüsselkonzepten, der Zusammenfassung, Kritik und/oder Integration von Ergebnissen und der Einordnung der Überprüfung in eine historische, konzeptionelle oder methodologische Struktur.

Webster und Watson (2002) haben hierfür eine Matrixdarstellung definiert, wie in Abbildung 6.4 ersichtlich. In dieser werden die analysierten Arbeiten als Reihen eingetragen und die selbst-definierten Eigenschaften, die diese erfüllen können als Spalten definiert. Somit könnte man für die einzelnen erfassten Arbeiten Eigenschaften wie "Datenbank", "Erscheinungsjahr" oder auch "Integrationsarten von KI im DevOps Prozess" definieren. Dadurch hat man eine gute Übersicht über die analysierten Arbeiten und kann diese im Bezug auf die Forschungsfrage gut auswerten.

Articles	Concepts														
	A			B			C			D			...		
	O	G	I	O	G	I	O	G	I	O	G	I	O	G	I
unit of analysis															
1					X				X						X
2	X				X	X		X							
...								X	X			X			

Abbildung 6.4: Matrixdarstellung Literaturanalyse (Webster & Watson, 2002)

### 6.3.5 Forschungsagenda

Beim Thema "Forschungsagenda" geht es nach Brocke et al. (2009) darum, auf der Grundlage der Literatursynthese eine Reihe von schärferen und aufschlussreicheren Fragen für die künftige Forschung zu entwickeln, die die Grundlage für die Erweiterung der Überprüfung bilden. Die Forschungsagenda kann auf der Grundlage der vorgeschlagenen Konzeptmatrix entwickelt werden und hebt häufig Forschungsbereiche hervor, welche einen Mangel an Forschungen aufweist.

## **6.4 Durchführung der systematischen Literaturrecherche**

Im folgenden Kapitel wird die Durchführung der systematischen Literaturrecherche dokumentiert und erfasst. Dabei wird der Rahmen und jegliche Definitionen laut Brocke et al. (2009) dokumentiert und eine klare Vorgehensweise erfasst.

### **6.4.1 Iterationen**

Im Zuge der Durchführung der systematischen Literaturrecherche wurden einige Iterationen durchlaufen, da durch neue Erkenntnisse die Recherche angepasst werden musste. In diesem Abschnitt werden die Änderungen und jeweiligen Gründe näher erläutert um Erkenntnisse zu vermitteln und mögliche Unklarheiten bestmöglich, für zukünftige Forschungen, zu beseitigen.

#### **6.4.1.1 Iteration 1**

In dieser Iteration wurde als Hauptsuchbegriff "AIOps" und "MLOps" definiert. Der Begriff "MLOps" wurde jedoch im Laufe der Iteration entfernt, da er viel unpassende Literatur mit sich gebracht hat. "MLOps" definiert meist die Adaption von DevOps auf Machine Learning Modelle und nicht, wie zuerst angenommen, die Integration von Machine Learning in DevOps.

#### **6.4.1.2 Iteration 2**

Aufgrund fehlender Weite im Bereich der Literatur wurde in dieser Iteration die Manuelle Suche nach Literatur in Fachjournals und Datenbanken ergänzt. Da dies auch dem Vorgehen der Literatursuche von Brocke et al. (2009) entspricht. Ebenfalls sorgt dieser Schritt dafür, die Qualität der Arbeiten zu erhöhen und nicht nur Arbeiten aus der Google Scholar Datenbank zu beachten.

#### **6.4.1.3 Iteration 3**

Im Laufe dieser Iteration wurden die Fragen für den Fragebogen für die Literatur in Kapitel 6.4.5 angepasst und die Antwortmöglichkeiten an den ausgearbeiteten Theorieteil anzupassen. Zudem wurde ergänzt für welchen Teil der Arbeit die Fragen relevant sind. Dies schafft einen roten Faden in der Arbeit und sorgt für Nachvollziehbarkeit.

### **6.4.2 Definieren des Überprüfungsrahmens**

In diesem Kapitel wird der Überprüfungsrahmen der in Kapitel 6 theoretisch erklärt wurde, für die Vorgehensweise dieser Arbeit definiert. Dabei wird sich auch wie im theoretischen Kapitel an die Forschung von Brocke et al. (2009) und Cook (2019) gehalten.

#### **6.4.2.1 Fokus**

Der Fokus dieser Arbeit und dieser Literaturrecherche liegt auf den Ergebnissen der zusammengetragenen Forschungen. Insbesondere die Aspekte der gewählten Integrationsarten und wenn erforscht deren daraus resultierenden Herausforderungen.

#### **6.4.2.2 Ziel**

Das Ziel dieser Arbeit liegt bei der Integration der gefundenen Ergebnisse in die gestellte Forschungsfrage und Hypothesen. Anhand einer Analyse der Forschungsergebnisse der recherchierten Arbeiten soll sich eine Antwort für die Forschungsfrage ableiten lassen.

#### **6.4.2.3 Organisation**

Die Organisation dieser Arbeit trifft am ehesten auf konzeptuell zu, da Cooper (1988) dies als Fragen, die künftige Bestrebungen beherrschen sollten beschreibt. Da künstliche Intelligenz viele weitere Forschungen mit sich bringen wird sowie das Zusammenspiel mit DevOps wird, könnte diese Arbeit zu künftigen Forschungen beitragen.

#### **6.4.2.4 Perspektive**

Diese Arbeit wird aus einer neutralen Perspektive erarbeitet um eine allgemeingültige und unbeeinflusste Antwort auf die Forschungsfrage zu finden.

#### **6.4.2.5 Zielgruppe**

Zielgruppe dieser Arbeit sind Fachleute, welche Interesse am Thema der KI Integration im IT Operations und DevOps Bereich haben oder vorhaben eine passende Art der Integration zu finden.

#### **6.4.2.6 Umfang**

Der Umfang der Recherche soll repräsentativ sein um eine große Menge an Arbeiten abzudecken und die Arten der Integrationen im IT Operations und DevOps Bereich gut widerzuspiegeln. Allerdings werden durch gewisse Qualitätskriterien durchaus Arbeiten ausgeschlossen, oder falls diese gar unpassend für die Beantwortung der Forschungsfrage sind.

### **6.4.3 Konzeptualisierung des Themas**

Das Konzept wurde mithilfe einer Mindmap erstellt um einen Überblick über Teile der Masterarbeit zu bekommen und diese zu definieren. So wurde im Zuge dieser Arbeit eine Mindmap erstellt welche einen

Teil beinhaltet wo die Vorgehensweise des SLR definiert wurde. Wie in Abbildung 6.5 ersichtlich enthält dieser Teil der Mindmap die Themen: Suchdatenbanken, Suchbegriffe/-phrasen und Einschlusskriterien.

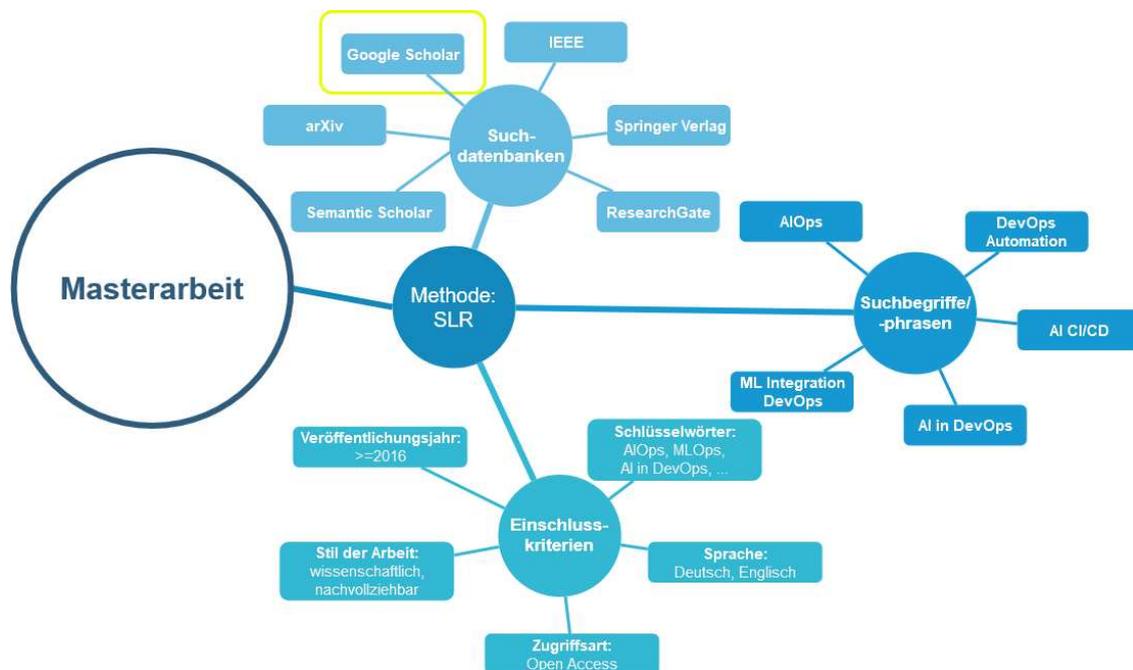


Abbildung 6.5: SLR Ausschnitt aus der Konzept-Mindmap

#### 6.4.3.1 Suchdatenbanken

In der Mindmap zur Auswahl gestellten Suchdatenbanken sind: Emerald Insight, Google Scholar, IEEE, Springer Verlag, ResearchGate. Um die Anzahl der Arbeiten und doppelte Ergebnisse zu reduzieren, beschränkt sich die Arbeit auf Google Scholar. Da zudem über Google Scholar arbeiten von anderen Herausgebern wie Springer, IEEE, ResearchGate gefunden werden können.

#### 6.4.3.2 Tools zur Suche

Bei der Literatursuche wird die Software "Publish or Perish" Anne-Wil Harzing (o. D.) verwendet, welche es ermöglicht, eine Google Scholar Suche abzusetzen und die Ergebnisse anschließend beliebig zu exportieren. Im Falle der Arbeit werden die Ergebnisse als ".csv" exportiert. Das Ergebnis dieser Suche sieht dann wie in Abbildung 6.6 ersichtlich aus.

The screenshot shows the 'Harzing's Publish or Perish' software interface. The main window displays a Google Scholar search for 'AIOps' with the following search criteria: Authors (empty), Years (2016 - 2023), Publication name (empty), ISSN (empty), Title words (AIOps), and Keywords (empty). The search results are displayed in a table with columns for Cites, Per year, Rank, Authors, and Title. The top result is 'AIOps: real-world challenges and research innova' with 170 citations and a per-year average of 716. The interface also includes a 'Citation metrics' section on the right, showing various metrics such as Publication years (2017-2023), Citation years (6 (2017-2023)), Papers (170), Citations (716), Cites/year (119.33), Cites/paper (4.21), Cites/author (315.61), Papers/author (95.25), Authors/paper (2.74), h-index (13), g-index (23), hI,norm (8), hI,annual (1.33), hA-index (9), and Papers with ACC >= 1,2,5,10,20 (60,36,17,7,3). The 'Paper details' section on the right provides options to 'Copy Results', 'Save Results', and 'Copy Paper Details'.

Search terms	Source	Papers	Cites	Cites/year
Intelligent Software Engineering [title] from 201...	Google Scholar	0	0	0.0
CI/CD with Artificial Intelligence [title] from 201...	Google Scholar	0	0	0.0
Generative AI [title], Automation DevOps from 2...	Google Scholar	0	0	0.0
DevOps AI [title] from 2016 to 2023, no patents	Google Scholar	19	179	25.5
DevOps [title], AIOps AI from 2016 to 2023, no ...	Google Scholar	39	426	60.8
AIOps [title] from 2016 to 2023, no patents	Google Scholar	170	716	119.3

Cites	Per year	Rank	Authors	Title
141	35.25	3	Y Dang, Q Lin, P Hu...	AIOps: real-world challenges and research innova
47	15.67	6	Y Li, ZM Jiang, H Li, ...	Predicting node failures in an ultra-large-scale cl
40	10.00	9	A Masood, A Hash...	AIOps: predictive analytics & machine learning in
34	8.50	91	Y Dang, Q Lin, P Hu...	AIOps: real-world challenges and research innova
27	27.00	7	AR Yeruva	Monitoring Data Center Site Infrastructure Using
26	6.50	10	A Levin, S Garion, E...	AIOps for a cloud object storage service
25	12.50	5	P Notaro, J Cardoso...	A survey of aiops methods for failure managemen
22	22.00	8	AR Yeruva	Providing & Personalized Healthcare Service To TI

**Abbildung 6.6:** Ausschnitt einer Google Scholar Suchanfrage über "Publish or Perish" von Anne-Wil Harzing (o. D.)

Es ermöglicht außerdem eine einfache und schnelle Übersicht über vorhandene Arbeiten und deren Grundinformationen wie:

- Anzahl der Zitate
- Zitate pro Jahr
- Rang
- Autor\*innen
- Titel
- Veröffentlichungsjahr
- Ort der Publikation
- Herausgeber
- Art/Typ (des Dokuments)

### 6.4.3.3 Suchbegriffe/-phrasen

Um passende Arbeiten aus den Datenbanken zu fixiert sich diese Arbeit auf wissenschaftliche Unterlagen, welche unter anderem über folgende Begriffe und deren Verkettung zurückgeliefert werden:

- AIOps
- Artificial Intelligence (AI)
- Language Model
- NLP
- Deep Learning
- DevOpsML

- DevOps
- Software Engineering
- Software Testing
- Automation
- Challenges

Die Kombination dieser Suchbegriffe sollen dabei helfen ausschließlich Arbeiten zu finden, die die Themen DevOps und KI vereinen.

#### 6.4.3.4 Ein- und Ausschlusskriterien

Als Einschlusskriterien werden folgende Punkte definiert:

- **Veröffentlichungsjahr:**  $\geq 2016$
- **Schlüsselwörter im Titel:** In Kapitel 6.4.3.3 genannt
- **Stil der Arbeit:** wissenschaftlich, nachvollziehbar
- **Zugriffsart:** Freier Zugriff
- **Sprache:** Deutsch, Englisch

Ausgeschlossen werden alle Arbeiten, welche diese Kriterien nicht erfüllen. Somit sind die Ausschlusskriterien genau gegengleich zu den Einschlusskriterien. Um bereits beim stellen der Suchabfragen irrelevante Arbeiten auszufiltern, sehen die für eine Einschränkungen relevanten Suchabfragen in "Publish or Perish" unter anderem wie in der Abbildung 6.7 aus.

The image shows a screenshot of the Google Scholar search interface. The search criteria are as follows: Authors: (empty), Years: 2016 - 2023, Publication name: (empty), ISSN: (empty), Title words: AI Ops, Keywords: (empty). The maximum number of results is set to 1000. The 'Include' options are checked for CITATION records and Patents. The interface includes buttons for Search, Search Direct, Clear All, Revert, and New, along with a Help link.

Abbildung 6.7: "Publish or Perish" Suchdefinition AI Ops

#### 6.4.4 Literatursuche

Der Ablauf der Literatursuche hält sich an die aktualisierte Form des PRISMA Schemas von Page et al. (2021). Page et al. (2021) hat in seiner Arbeit den Prozess von Prisma erweitert und angepasst, daraus hat sich der Prozess in Abbildung 6.8 ergeben, genannt wird die aktualisierte Version PRISMA 2020. PRISMA 2020 besteht außerdem aus einer Item-Checkliste, einer Abstract-Checkliste und aktualisierte Übersichten. Die Item-Checkliste liefert 27 Punkte zur Analyse von in der Recherche gefundenen Arbeiten, wobei die Abstract-Checklist 12 Punkte liefert um die eigene Arbeit korrekt zu strukturieren und aufzubauen.

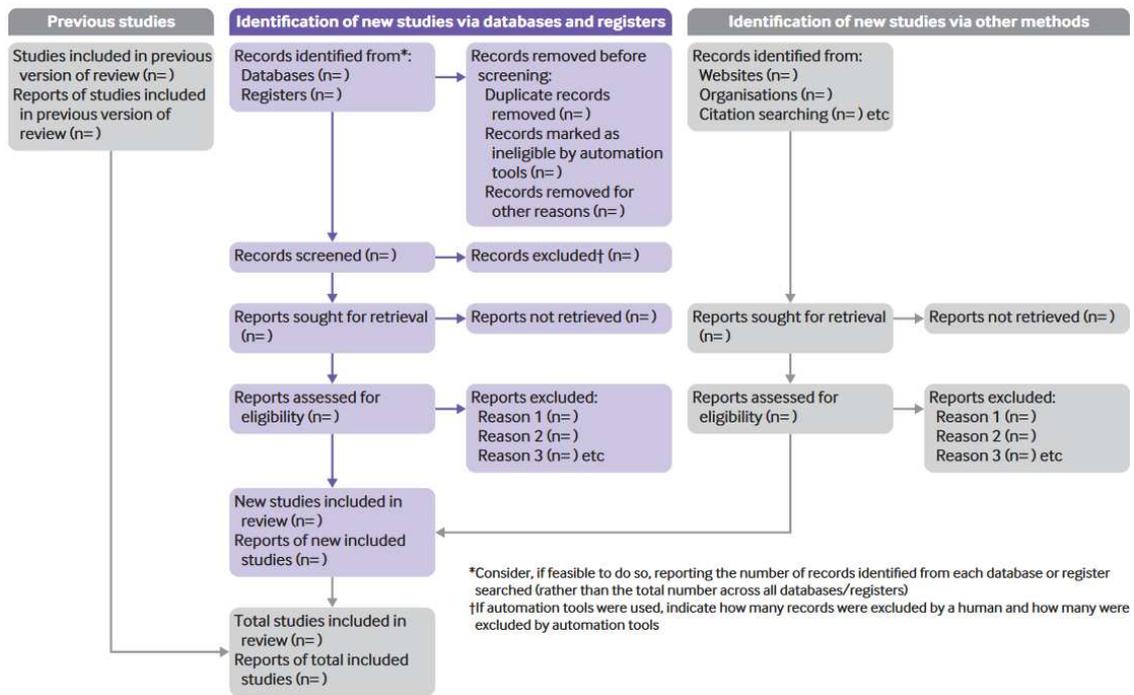


Abbildung 6.8: Aktualisierter PRISMA Prozess von Page et al. (2021)

In der Arbeit von Page et al. (2021) wird außerdem betont, dass alle grau gefärbten Elemente der Grafik optional sind und bei fehlender Notwendigkeit entfernt werden können. Da es noch keine vorherigen Versionen dieser Recherche gibt und auch keine weiteren Methoden im Laufe der Studien verwendet werden, hat sich ein angepasster Prozess ergeben, welcher in Abbildung 6.9 ersichtlich ist. Des Weiteren wurden die einzelnen Aktivitäten im Prozess einer Phase zugeteilt, welche in folgenden Kapitel durchgeführt werden.

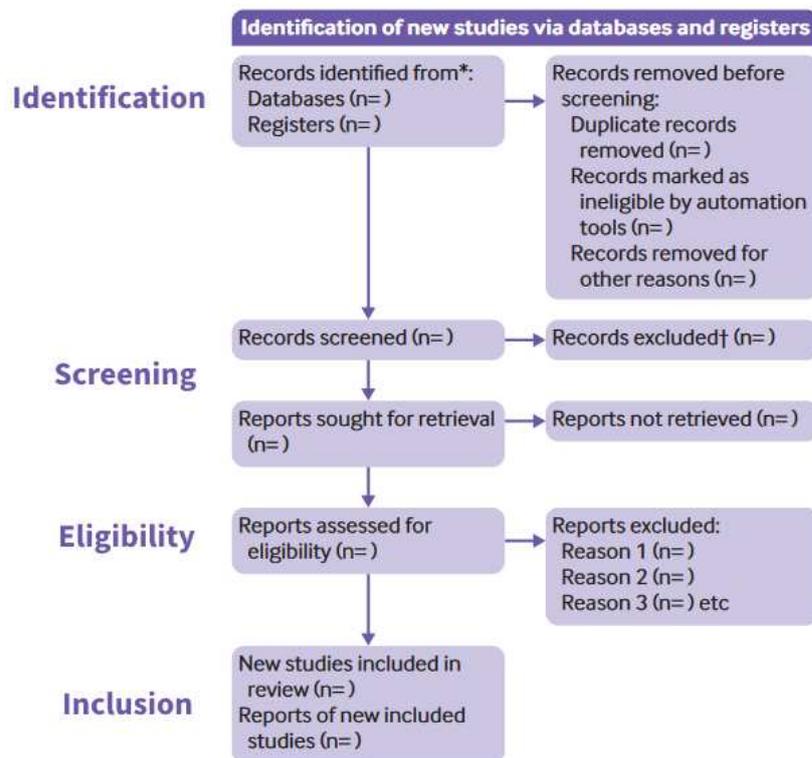


Abbildung 6.9: Auf diese Masterarbeit angepasster PRISMA Prozess von Page et al. (2021)

#### 6.4.4.1 Identification Phase

Laut Page et al. (2021) werden in dieser Phase relevante Studien durch eine umfassende Suche in mehreren Datenbanken und anderen Quellen identifiziert. Diese Arbeit verfolgt dabei zwei Suchstrategien. Die erste beschreibt eine manuelle Suche und die zweite eine Software unterstützte Literatursuche.

##### Manuelle Literatursuche

Die manuelle Literatursuche wurde über folgende Suchdatenbanken durchgeführt:

- arXiv
- ResearchGate
- Semantic Scholar
- Springer
- IEEE

Zusätzlich zur Suche über diese Datenbanken wurden die Quellen von bereits vorhandenen Arbeiten analysiert und taugliche Quellen in ein Tabellenkalkulationsprogramm, zur weiteren verarbeiten, übertragen.

##### Software unterstützte Literatursuche

Insgesamt ergeben sich, mithilfe der Suche aus Abbildung 6.7 über Google Scholar, **224 Ergebnisse** Arbeiten. Die Software "Publish or Perish" bietet die Möglichkeit im Anschluss die Ergebnisse zu exportieren. Die Ergebnisse werden daher nun in eine ".csv" Datei exportiert und über das Cloud-Tabellenkalkulationsprogramm "Google Sheets" importiert.

Desweiteren wurden manuell weitere Arbeiten gesucht, insgesamt wurden durch dieses Vorgehen weitere **25 Arbeiten** hinzugefügt.

Da diese Darstellung nicht übersichtlich ist und irrelevante Daten enthält, wird sie aufbereitet. Im Laufe der Formatierung wurden folgende Spalten entfernt, da diese zur Kontrolle der Einschlusskriterien und Übersicht nicht relevant sind:

- ISSN - Leer
- CitationURL - Leer
- Volume - Leer
- Issue - Leer
- StartPage - Leer
- EndPage - Leer
- QueryDate - Erfasst über Kommentar
- CitesPerAuthor - Irrelevant
- FullTextURL - Irrelevant
- RelatedURL - Irrelevant
- CitesURL - Irrelevant
- Source - Irrelevant / Inkonsistent

Desweiteren wurden Einträge entfernt welche..

- ..keinen Link "ArticleUrl" aufweisen: 58 Einträge
- ..einen nicht Deutsch oder Englisch sprachigen Abstract oder Titel haben: 9 Einträge
- ..den gleichen Titel tragen: 10 Einträge

Somit bleiben **147 Ergebnisse** am Ende der Identification Phase.

#### **6.4.4.2 Screening Phase**

Laut Page et al. (2021) werden in dieser Phase der Titel und Abstract der identifizierten Studien analysiert, um ihre Relevanz für die Forschungsfrage zu ermitteln und gegebenenfalls auszuschließen. Bei den Arbeiten wurde spezifisch auf die Punkte in Abbildung 6.10 geachtet. Wenn jeder dieser Punkte erfüllt, also mit "Ja" beantwortet werden konnte, wurde die Arbeit in die nächste Phase übernommen. Die 25 Manuell hinzugefügten Arbeiten wurden erst mit Ende dieser Phase in die Auswertung mitaufgenommen, da diese keine Identification und Screening Phase nötig hatten. Durch das manuelle filtern und herausuchen der Arbeiten wurde die Identification und Screening Phase bereits beim Suchen durchgeführt.

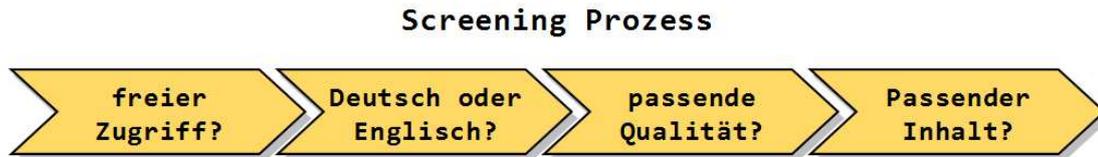


Abbildung 6.10: Prozess Screening Phase

Im Zuge dieses Prozesses wurden insgesamt 86 Arbeiten der 147 Verbleibenden ausgeschlossen, somit bleiben, inklusive der manuell gesuchten Arbeiten, 85 Arbeiten für die Eligibility Phase über. Dabei wurden die meisten Arbeiten durch fehlenden Zugriff ausgeschlossen, wie in Diagramm 6.11 zu erkennen.

### Ergebnis Screening - Ausschlusskriterien

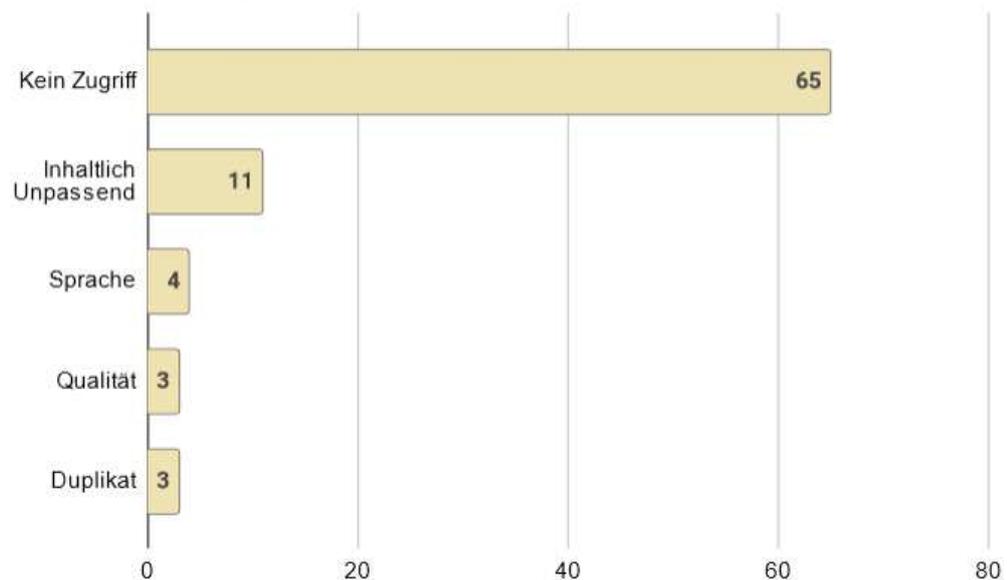
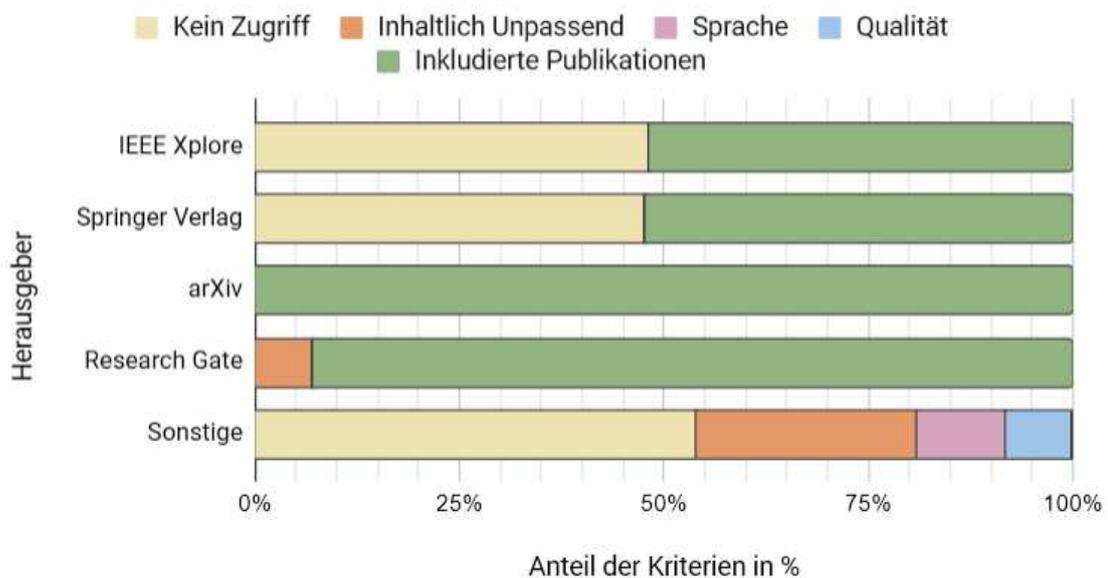


Abbildung 6.11: Diagramm Ausschlusskriterien

Der Ausschlussgrund "Kein Zugriff", also kein freier Zugriff auf die Arbeit, ist mit Abstand am häufigsten. Quellen wie IEEE Xplore haben hierbei einen sehr hohen Anteil wie in der Abbildung zur Verteilung 6.12 zu sehen ist. Grund dafür sind exklusive Inhalte oder Inhalte die nur mit Hochschul-Zugang eingesehen werden können. Des weiteren ist der zweite Punkt "Inhaltlich Unpassend", dieser alle Arbeiten ausgeschlossen, welche höchst wahrscheinlich nicht zur weiteren Analyse notwendig sind. Die inhaltliche Analyse war aber hierbei sehr oberflächlich und nur über Abstract und Titel, daher war die Inklusion einiger Arbeiten sehr großzügig um zu vermeiden, dass zu viele Arbeiten anhand des Abstracts ausgeschlossen werden. Wie anhand des Diagramms 6.12 auch zu sehen ist, wurde ein Großteil der Herausgeber "IEEE Xplore", "Springer Verlag" und "ACM DL" ausgeschlossen. Bei beiden Herausgebern wurden dabei über 90% ausgeschlossen nur weil auf diese Publikation kein freier Zugriff besteht. Bei arXiv wurden fast alle Publikation inkludiert.

## Verteilung der Ausschlusskriterien



**Abbildung 6.12:** Verteilung der Ausschlusskriterien mit Hervorhebung der Top 4 Herausgeber, relativ zur Anzahl der Gesamtpublikationen

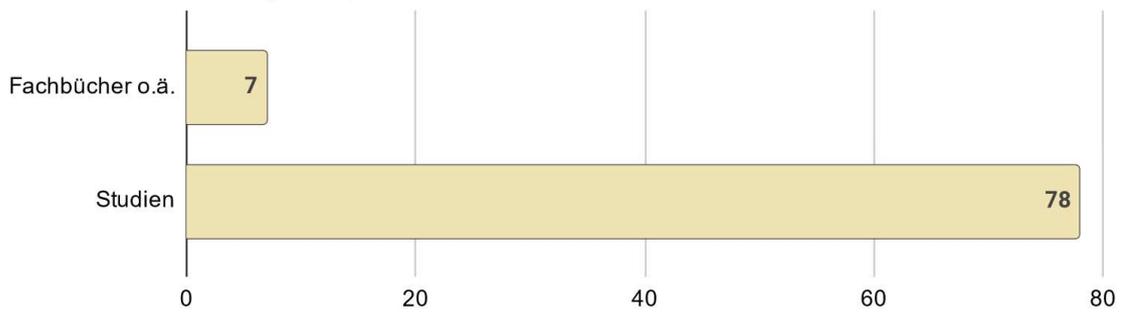
### 6.4.4.3 Eligibility Phase

Laut Page et al. (2021) wird in dieser Phase bereits der Volltext der einzelnen Studien beleuchtet. Hierbei wird ebenfalls auf Einschlusskriterien überprüft, welche in Kapitel 6.4.3.4 angeführt wurden. Da bereits durch die Screening Phase ein Großteil der Kriterien analysiert wurde, wird in der Eligibility Phase die Nachvollziehbarkeit analysiert. Hierbei müssen die folgenden Fragen mit "Ja" beantwortet werden, um die Arbeit zu inkludieren.

- Wurde systematisch nach einer definierten Methodik gearbeitet?
- Wurden qualitativ hochwertige Quellen verwendet? (Fachzeitschriften, Fachbücher, etc.)
- Kommt die Arbeit zu einem Conclusio?

Eine Ausnahme bilden Fachbücher oder ähnliches. Diese müssen diese Kriterien nicht mit "Ja" beantworten um eingeschlossen zu werden. Wie in Abbildung 6.13 ersichtlich, gibt es innerhalb der Quellen 7 Fachbücher.

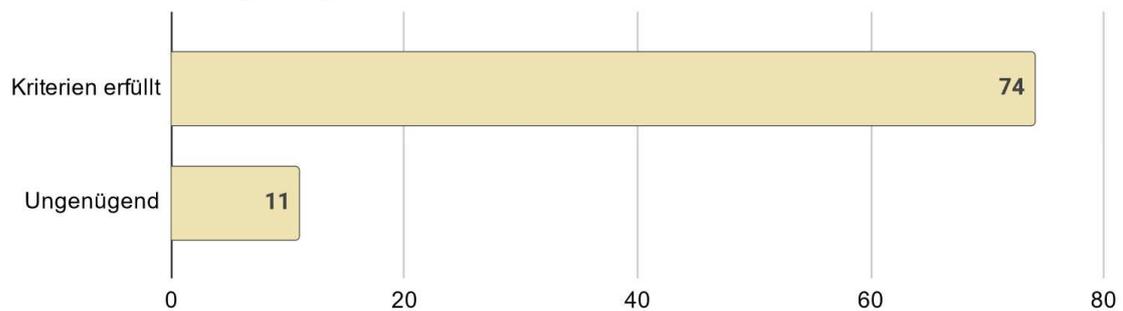
### Fachbücher Eligibility Phase



**Abbildung 6.13:** Anzahl der Fachbücher innerhalb der Eligibility Phase

Wie in Abbildung 6.14 ersichtlich werden durch die Eligibility Phase insgesamt **10** Arbeiten, anhand einer oder mehr der zuvor genannten Kriterien, ausgeschlossen. Wobei hier zu erwähnen ist, dass 4 dieser 10 Arbeiten zum Zeitpunkt der Eligibility Phase nicht mehr verfügbar waren, obwohl diese zuvor in der Screening Phase noch verfügbar waren.

### Ausschluss Eligibility Phase

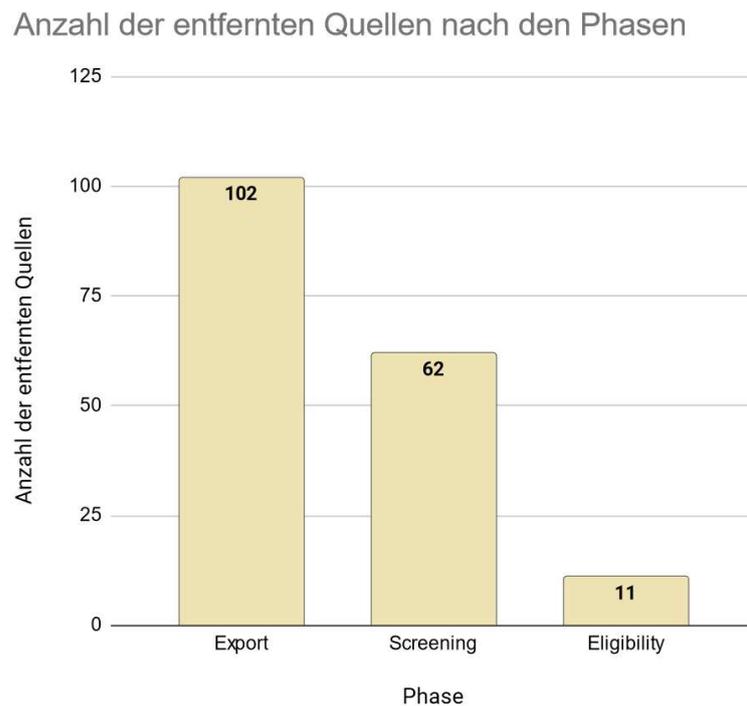


**Abbildung 6.14:** Anzahl der ausgeschlossenen Literatur innerhalb der Eligibility Phase

#### 6.4.4.4 Inclusion Phase

Laut Page et al. (2021) werden in dieser Phase die Studien ausgewählt, die die Zulassungskriterien für die Aufnahme in die Überprüfung erfüllen.

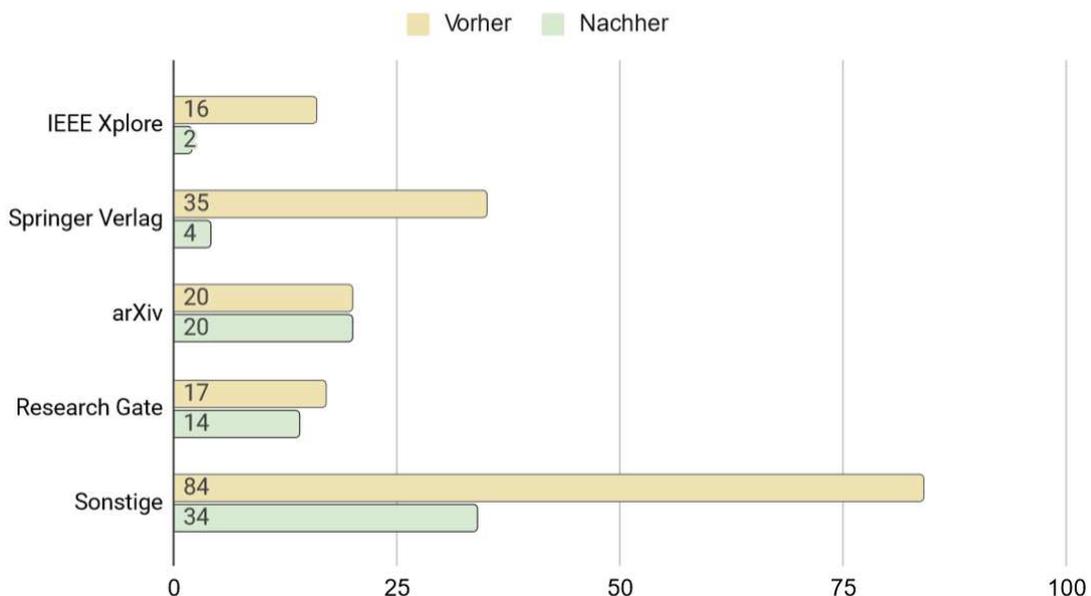
Gesamt werden, nach Abschluss der Filter-Phasen eine Anzahl von 74 Arbeiten in der Literaturanalyse im Detail analysiert. In Abbildung 6.15 sieht man die Anzahl der entfernten Arbeiten je nach Phase. Die Grafik veranschaulicht, dass je nach Stufe die Filterung filigraner wurde und immer weniger Arbeiten ausgeschlossen wurden.



**Abbildung 6.15:** Anzahl der entfernten Arbeiten, gruppiert nach der SLR Phase

In Abbildung 6.16 ergibt sich auch ein Vergleich der Publisher, die Anzahl an Arbeiten die zuvor von dem gewählten Publisher inkludiert waren und nun am Ende der Filter-Phasen noch übrig sind.

### Gegenüberstellung der Anzahl der Quellen nach Publisher (nach den Auswertungsphasen)



**Abbildung 6.16:** Vergleich der Anzahl der Quellen, gruppiert nach Publisher nach den Filter-Phasen

## 6.4.5 Literaturanalyse

Nach der umfassenden Literaturrecherche und dem anschließenden Ausschluss bestimmter Arbeiten auf der Grundlage vorher festgelegter Kriterien wurden insgesamt 74 Quellen für die analytische Überprüfung zusammengestellt. Diese Quellen, die sorgfältig kuratiert wurden, bilden die Grundlage für diese Untersuchung.

In der Phase der Literaturanalyse werden diese 74 Quellen einer strengen Prüfung unterzogen. Jede Quelle wird eingehend untersucht, um eine Reihe folgender festgelegter Fragen zu beantworten. Diese Fragen wurden strategisch so konzipiert, dass sie wertvolle Informationen aus jeder Quelle extrahieren und so die Analyse leiten um die Forschungsfrage so deckend wie möglich zu beantworten. Dieser methodische Ansatz der Literaturanalyse gewährleistet eine umfassende Untersuchung des Forschungsthemas und lässt keinen Aspekt unberücksichtigt.

### 6.4.5.1 Frage 1

Wurde eine KI-Technologie in dieser Arbeit im Bezug auf DevOps oder IT Operations integriert oder erwähnt?

Diese Frage dient als Leitfaden um, für die Beantwortung der Forschungsfrage, unpassende Forschungen auszugliedern. Bei Beantwortung mit "Nein" ist die Literatur unpassend zur weiteren Bearbeitung.

#### Antwortmöglichkeiten:

- Ja

- Nein

#### **6.4.5.2 Frage 2**

In welcher Phase des DevOps oder IT Operations Prozesses wird KI integriert?

Diese Frage ist essenziell für die Forschungsfrage. Sie kann auch potenzielle Lücken oder Möglichkeiten für die weitere Integration von KI in anderen Phasen des Prozesses aufzeigen. Dies könnte zu einem ganzheitlich und effektiveren Ansatz für die KI-Integration in DevOps führen.

Ebenfalls wichtig zur Beantwortung der Hypothese H3.

#### **Antwortmöglichkeiten nach Kapitel 2.1.2 und 2.6:**

- Plan
- Code
- Build
- Test
- Release
- Deploy
- Operate
- Monitor

#### **6.4.5.3 Frage 3**

Welche Herausforderungen oder Hindernisse gibt es bei der Integration von KI in DevOps/ITOps?

Dies ist ebenso eine essenzielle Frage zur Beantwortung der Forschungsfrage. Zu Herausforderungen können sowohl technische, organisatorische, kulturelle als auch finanzielle Herausforderungen gehören.

Ebenfalls wichtig zur Beantwortung der Hypothese H1.

#### **Antwortmöglichkeiten gruppiert aus Kapitel 4.3.1 und 5.2:**

- Technische Herausforderungen
  - Datenbezogen
  - Integration
  - Komplexität
- Organisatorische Herausforderungen
  - Lack of Skills
  - Change Management
- Kulturelle Herausforderungen
  - Vertrauensprobleme
  - Ethische Aspekte

- Finanzielle Herausforderungen

#### **6.4.5.4 Frage 4**

Gibt es vorgeschlagene Lösungen für diese Herausforderungen?

Dies kann helfen, Best Practices und erfolgreiche Fallstudien zu identifizieren.

Ergänzend zur Beantwortung der Hypothese H1.

#### **Antwortmöglichkeiten:**

- Ja
- Nein

#### **6.4.5.5 Frage 5**

Welche KI-Techniken werden in DevOps/ITOps verwendet?

Diese Frage kann helfen, den aktuellen Stand der KI-Integration in diesem Bereich zu verstehen.

#### **Antwortmöglichkeiten nach Kapitel 5.3:**

- Datenintegration
- Auto Triggers
- Predictive ALM
- Anomaly Detection
- Root Cause Analysis
- Sonstige (Freitext)

#### **6.4.5.6 Frage 6**

Welche KI-Methoden werden in DevOps/ITOps verwendet?

Diese Frage kann helfen, die Verbreitung der KI-Methoden in diesem Bereich zu veranschaulichen.

#### **Antwortmöglichkeiten nach Kapitel 4.1.2:**

- Maschinelles Lernen
- Neuronale Netze
- Expertensysteme
- Entscheidungsbäume
- Natürliche Sprachverarbeitung
- Sonstige (Freitext)

#### **6.4.5.7 Frage 7**

Was sind die Vorteile der Integration von KI in DevOps/ITOps?

Mithilfe der Frage können eventuell versteckte Benefits aufgezeigt werden. Ebenfalls wichtig zur Beantwortung der Hypothese H2.

#### **Antwortmöglichkeiten nach Kapitel 3.5.4 und 5.2:**

- Steigerung der geschäftlichen Agilität
- Verlängerte Hardware- und Cloud-Lebensdauer
- Kostenvorhersage
- Erleichterte Einhaltung gesetzlicher Vorschriften
- Verbesserte Kostenreduzierung
- Verbesserte Netzwerk-Effizienz
- Erleichtert IT-Transformation
- Unterstützt Cloud-Migration
- Sonstige (Freitext)

#### **6.4.5.8 Frage 8**

Welche Metriken oder KPIs werden verwendet, um den Erfolg der KI-Integration in DevOps/ITOps zu messen?

Dies kann helfen zu verstehen, wie der Erfolg in diesem Bereich definiert und gemessen wird.

Ergänzend zur Beantwortung der Hypothese H2 um Vorteile festzustellen und diese Messbar zu machen.

#### **Antwortmöglichkeiten:**

- Freitext

### **6.4.6 Forschungsagenda**

In diesem Abschnitt geht es darum auf der Grundlage der Analyse der Arbeiten detailliertere Fragen für zukünftige Forschungen zu definieren. Dies wurde allerdings nach der Diskussion der Ergebnisse in Kapitel 8.5 aufgeführt und erläutert.

## 7 ERGEBNISSE

Dieses Kapitel bietet eine Erfassung der Ergebnisse, die aus der durchgeführten SLR hervorgegangen sind. Die SLR untersuchte eine Stichprobe von insgesamt 75 Arbeiten aus einem großen Themengebiet. Diese Stichprobe wird als eine Momentaufnahme betrachtet - welche potenzielle Einflüsse, durch die verwendeten Datenbanken und spezifischen Schlüsselwörter hat. Trotz dieses begrenzten Fokus bietet die Zusammenstellung der Erkenntnisse, Daten und Trends aus diesen Studien einen strukturierten Einblick in die aktuelle Domain-Landschaft. Durch die sorgfältige Analyse dieser kuratierten Teilmenge sollen aufkommende Muster entschlüsselt, Korrelationen festgestellt und die Lücken im vorhandenen Wissen aufgezeigt werden. Anschließend werden im Kapitel 8 die Ergebnisse auf die Forschungsfrage und Hypothesen H1, H2 und H3 bezogen.

Die Erfassung der Ergebnisse erfolgte über ein Google Formular. Anschließend wurden die Daten in ein Google Sheet übertragen und nach der Matrixdarstellung von Webster und Watson (2002) ausgewertet. Zudem wurden jegliche unten angeführten Diagramme in diesem Google Sheet erstellt um in einem einzigen Dokument die gesamte Literaturanalyse nachvollziehbar zu gestalten.

### 7.1 Ausschluss

Das Diagramm 7.1 zeigt die Anzahl der wissenschaftlichen Arbeiten, die aufgrund der ersten Frage, aus dem Fragebogen von Kapitel 6.4.5, herausgefiltert wurden. Die x-Achse repräsentiert die Antwort auf die Frage, ob die Literatur für eine weitere Untersuchung geeignet ist, mit den Kategorien 'Ja' und 'Nein'. Die y-Achse zeigt die Anzahl der Arbeiten in jeder Kategorie.

Es ist deutlich zu erkennen, dass die Mehrheit der Arbeiten, insgesamt 65, als geeignet für eine weitere Untersuchung eingestuft wurden. Im Gegensatz dazu wurden nur eine geringe Anzahl von Papieren, insgesamt 15, als ungeeignet eingestuft.

Diese Ergebnisse zeigen, dass die meisten der untersuchten wissenschaftlichen Arbeiten relevante Informationen enthielten, die für die weitere Bearbeitung von Bedeutung waren.

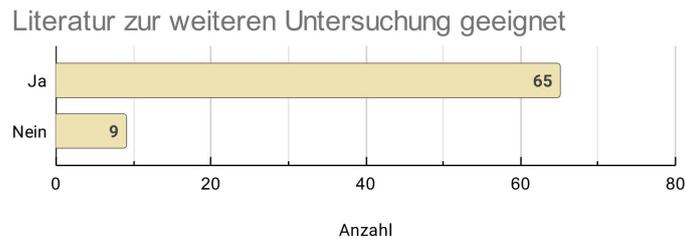


Abbildung 7.1: Anzahl der zur weiteren Behandlung geeigneten Literatur

## 7.2 Verteilung der unterstützten DevOps Phasen

Das Diagramm 7.2 zeigt die Anzahl der Literaturergebnisse für jede Phase des DevOps-Prozesses. Dieses Diagramm bezieht sich auf Frage 2 aus dem definierten Fragenkatalog. Die x-Achse repräsentiert die Anzahl der Literaturergebnisse und die y-Achse zeigt die unterstützte DevOps-Phase.

Die Tatsache, dass die Summe der Literatur für jede Phase die Gesamtzahl der analysierten Arbeiten übersteigt, lässt sich dadurch erklären, dass einige Arbeiten mehrere Phasen unterstützen. Die Abgrenzung der Phasen ist oft schwierig und die Grenzen sind häufig nicht klar zu erkennen.

Allerdings zeigt das Diagramm 7.2, dass es in der untersuchten Stichprobe, deutlich mehr Arbeiten im Bereich Operate und Monitor, als zum Beispiel im Plan, Build und Release Bereich gibt.

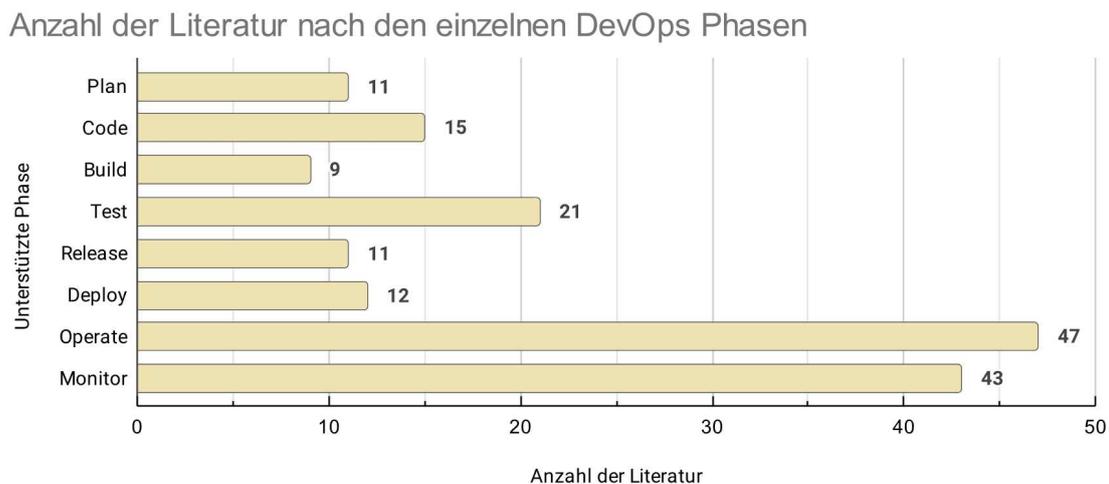


Abbildung 7.2: Unterstützte DevOps Phasen nach Anzahl der Ergebnisse

## 7.3 Herausforderungen

Das Diagramm 7.3 zeigt die Anzahl der Literaturergebnisse, welche im Zuge der Arbeit Herausforderungen in diesem Bereich erwähnen. Dieses Diagramm bezieht sich auf Frage 3 aus dem definierten Fragenkatalog. Die x-Achse repräsentiert die Anzahl der Literaturergebnisse und die y-Achse zeigt ob Herausforderungen enthalten sind.

Die Mehrzahl der analysierten Arbeiten erwähnt also mit 46 Stück keine Herausforderungen.

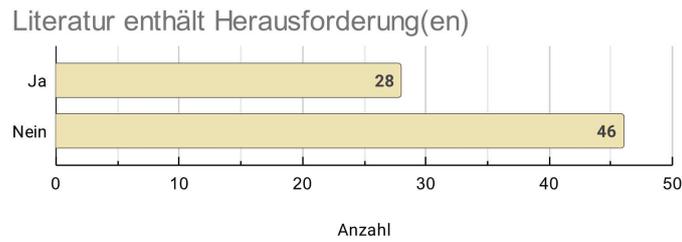


Abbildung 7.3: Anzahl der Literatur mit angeführten Herausforderungen von KI-Integration

### 7.3.1 Verteilung der Kategorien

Das Diagramm 7.4 zeigt die Anzahl der Literaturergebnisse für jede Kategorie der erwähnten Herausforderungen. Dieses Diagramm bezieht sich auf Frage 3 aus dem definierten Fragenkatalog. Die x-Achse repräsentiert die Anzahl der Literaturergebnisse und die y-Achse zeigt die Kategorie der Herausforderungen.

Dabei werden mit 27 Arbeiten am häufigsten technische Herausforderungen erwähnt, dahinter folgen organisatorische Herausforderungen. Kulturelle sowie finanzielle Herausforderungen werden kaum erwähnt.

Anzahl der Literatur nach Kategorie der Herausforderungen

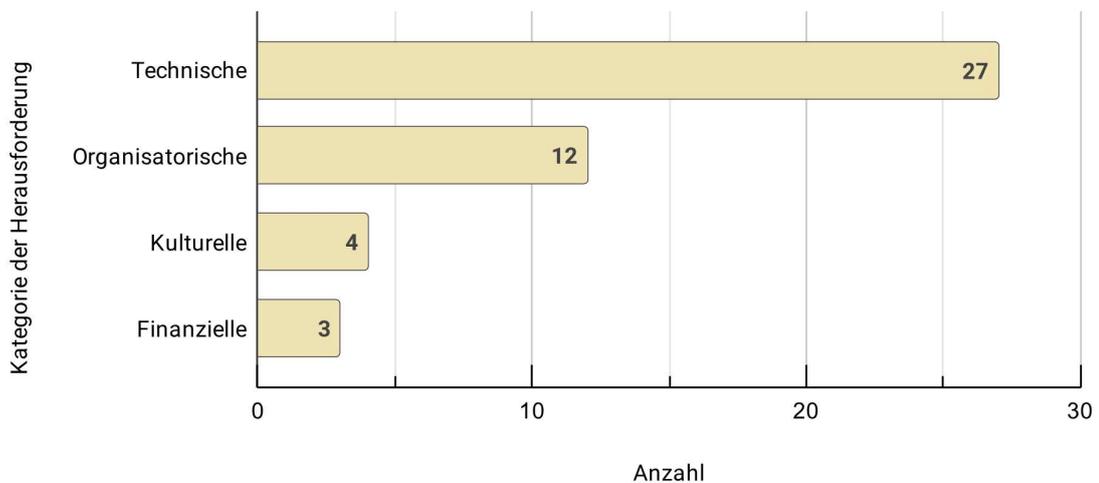


Abbildung 7.4: Unterstützte DevOps Phasen nach Anzahl der Ergebnisse

### 7.3.2 Herausforderungen grob - je Phase

Das Diagramm 7.5 zeigt den Prozentualen Anteil der Kategorien für jede Phase des DevOps Prozesses. Es zeigt welche Phase bei den jeweiligen Herausforderung gewählt wurde. Das Diagramm bezieht sich auf Frage 1 und 3 aus dem definierten Fragenkatalog. Die x-Achse repräsentiert den Prozentualen Anteil der Kategorien und die y-Achse zeigt die einzelnen DevOps Phasen.

Das Diagramm zeigt dadurch, ein Recht ausgeglichenes Verhältnis je Phase und erwähnter Herausforderung. Es ist kaum möglich klar festzulegen, dass in einer Phase eine Kategorie besonders relevant ist.

Kategorie der Herausforderungen nach Phase in %

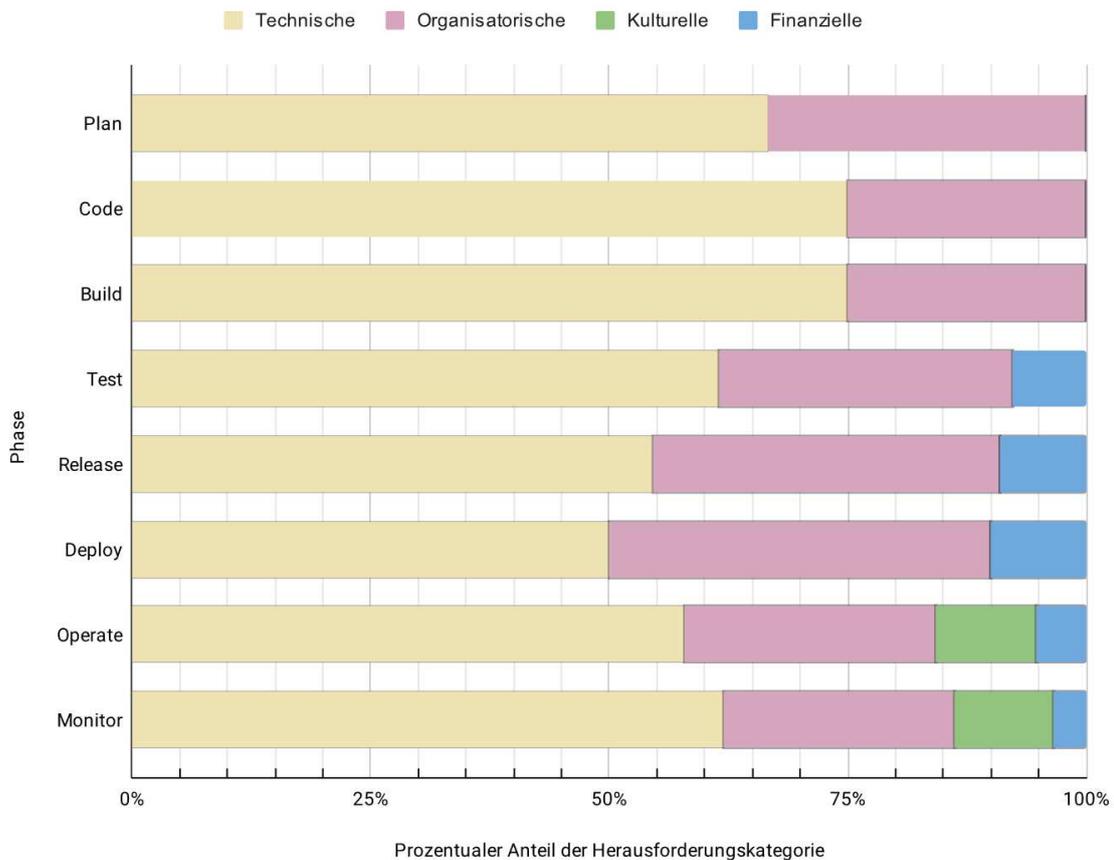


Abbildung 7.5: Kategorie der Herausforderungen nach Phase in %

### 7.3.3 Liste der Herausforderungen

Das Diagramm 7.6 zeigt die 10 häufigst erwähnten Herausforderungen im Bezug auf KI Einsatz in ITOps bzw. DevOps Prozess. Das Diagramm bezieht sich auf Frage 3 aus dem definierten Fragenkatalog. Die x-Achse repräsentiert die Anzahl der Literaturergebnisse und die y-Achse zeigt die einzelnen Herausforderungen.

Datenbezogene Herausforderungen werden dabei am häufigsten erwähnt. Darunter fallen unter anderem Herausforderungen bezogen auf Datenqualität, mögliche Datenquellen, Datenquantität und Menge der Daten. Desweiteren werden Herausforderungen

### Mögliche Herausforderungen - Top 10

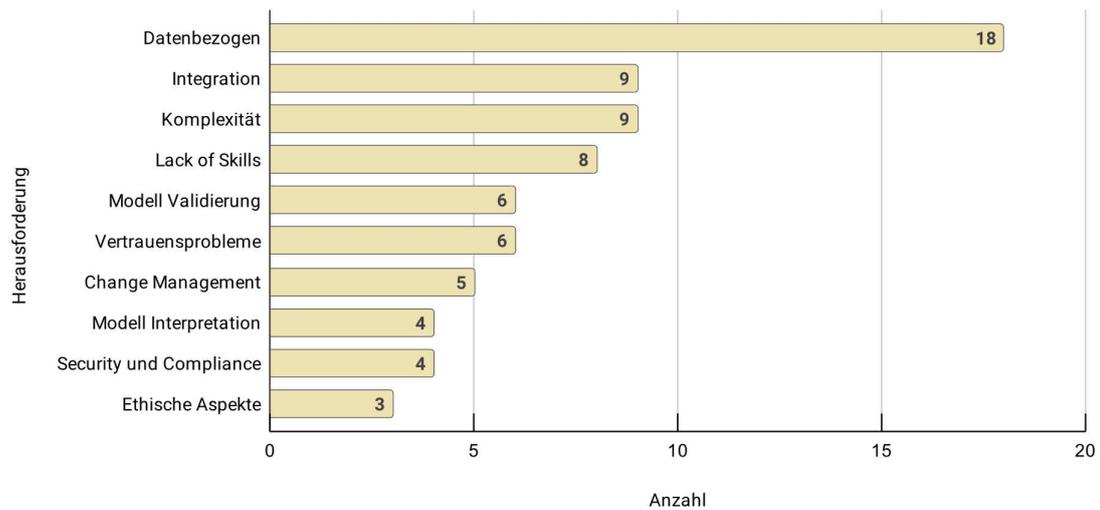


Abbildung 7.6: Mögliche Herausforderungen - Top 10

### 7.3.4 Lösungen

Das Diagramm 7.7 zeigt die Anzahl der Literaturergebnisse, welche im Zuge der Arbeit Lösungen zu erwähnten Herausforderungen erwähnen. Dieses Diagramm bezieht sich auf Frage 4 aus dem definierten Fragenkatalog. Die x-Achse repräsentiert die Anzahl der Literaturergebnisse und die y-Achse zeigt ob Lösungen enthalten sind.

Die Mehrzahl der analysierten Arbeiten erwähnt also mit 18 Stück keine Lösungen zu den erwähnten Herausforderungen. 10 Stück jedoch bieten eine Lösung oder einen Lösungsansatz zu einer oder mehr der genannten Herausforderung.

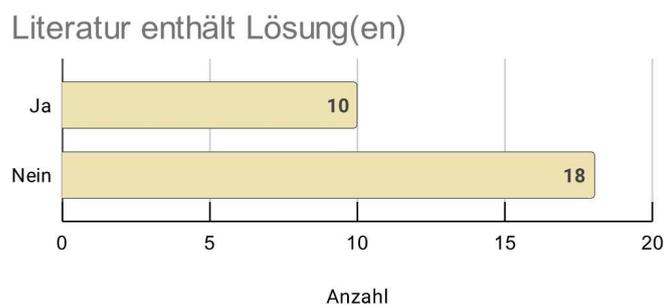


Abbildung 7.7: Anzahl der Literatur welche zu den angeführten Herausforderungen auch Lösungsansätze enthält

### 7.3.5 Beispiele

Da die Liste der Lösungen sehr individuell und umfangreich ist, folgen in Tabelle 7.3.5 erwähnte Lösungsansätze zu verbreiteten Herausforderungen.

Herausforderung	Lösung aus der Literatur
-----------------	--------------------------

Continued on next page

(Continued)

Datenbezogen	<ul style="list-style-type: none"> <li>• Pflege und gemeinsame Nutzung einer Taxonomie von Instrumenten und Methoden</li> </ul>
Integration	<ul style="list-style-type: none"> <li>• Schrittweise Integration</li> <li>• Beginnen mit einfachen statistischen Analysen zur Problemerkennung</li> </ul>
Komplexität	<ul style="list-style-type: none"> <li>• Konzentration auf Datenqualität und -quantität</li> <li>• Schrittweise Integration</li> <li>• Beginnen mit einfachen statistischen Analysen zur Problemerkennung</li> </ul>
Lack of Skills	<ul style="list-style-type: none"> <li>• Förderung der Zusammenarbeit und des Wissensaustauschs zwischen IT-Teams und Datenwissenschaftlern</li> <li>• Erwerb von Wissen und Verständnis für KI und maschinelles Lernen - durch Weiterbildung</li> </ul>

## 7.4 Verteilung KI-Techniken

Das Diagramm 7.8 zeigt die Top 5 erwähnten KI-Techniken. Es zeigt welche KI-Techniken in den Arbeiten eingesetzt wurde im Bezug auf eine Aktivität im DevOps oder ITOps Prozess. Das Diagramm bezieht sich somit auf Frage 5 aus dem definierten Fragenkatalog. Die x-Achse repräsentiert die Anzahl der Arbeiten und die y-Achse zeigt die einzelnen KI-Techniken. Das Diagramm wurde auf 5 KI-Techniken beschränkt, da diese mehr als 1 mal erwähnt wurden. Andere Erwähnung waren zum Beispiel:

1. Random Forest
2. LSTM
3. Reinforcement Learning
4. ...

Das Diagramm zeigt deutlich, dass ein Großteil der Arbeiten mit einer Anomaly Detection Technik arbeitet. Diese war vor allem in Arbeiten, welche die Monitor oder Operate Phase unterstützen, beliebt. Dahinter folgt die Root Cause Analysis und Predictive ALM.

## Top 5 verwendete KI-Techniken

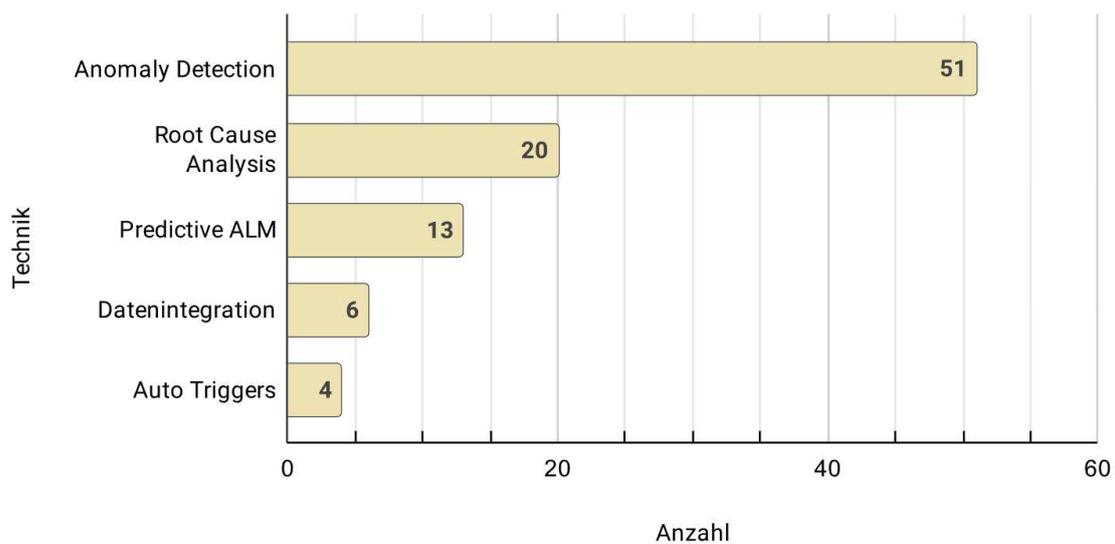


Abbildung 7.8: Top 5 verwendete KI-Techniken

## 7.5 Verteilung KI-Methoden

Das Diagramm 7.9 zeigt die Top 6 erwähnten KI-Methoden. Es zeigt welche KI-Techniken in den Arbeiten eingesetzt wurde im Bezug auf eine Aktivität im DevOps oder ITOps Prozess. Das Diagramm bezieht sich somit auf Frage 6 aus dem definierten Fragenkatalog. Die x-Achse repräsentiert die Anzahl der Arbeiten und die y-Achse zeigt die einzelnen KI-Methoden. Das Diagramm wurde auf 6 KI-Methoden beschränkt, da diese mehr als 1 mal erwähnt wurden. Andere Erwähnung waren unter anderem:

1. Graph Neural Network
2. Swarm Intelligence
3. Bayesian Networks
4. ...

Das Diagramm zeigt deutlich, dass ein Großteil der Arbeiten mit Machine Learning Methoden arbeitet. Mit großem Abstand folgt dahinter die Verwendung von Neuronalen Netzen. Knapp dahinter findet sich NLP wieder.

## Top 6 verwendete KI-Methoden

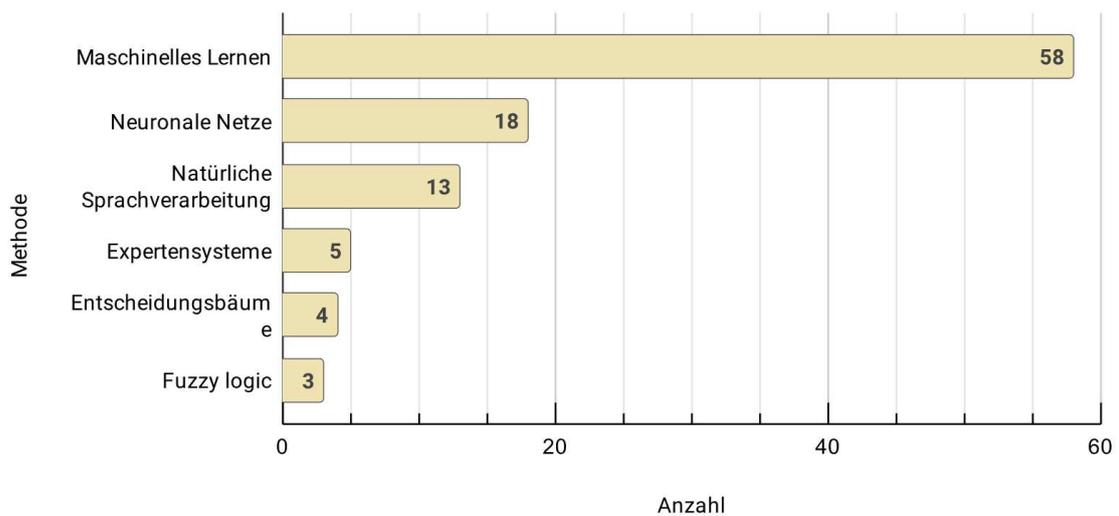


Abbildung 7.9: Top 6 verwendete KI-Methoden

## 7.6 KI-Methoden - je Phase

Das Diagramm 7.10 zeigt den Prozentualen Anteil der verwendeten KI-Methode für jede Phase des DevOps Prozesses. Es zeigt welche Phase bei den jeweiligen KI-Methode unterstützt wurde. Das Diagramm bezieht sich auf Frage 1 und 6 aus dem definierten Fragenkatalog. Die x-Achse repräsentiert den Prozentualen Anteil der KI-Methode und die y-Achse zeigt die einzelnen DevOps Phasen.

Das Diagramm zeigt dadurch, ein Recht ausgeglichenes Verhältnis je Phase und erwähnter KI-Methode. Allerdings deutet das Diagramm darauf hin, dass der Einsatz von NLP bei der "Plan", "Build" und "Test" Phase höher ist, als bei den Betrieb Phasen "Release", "Deploy", "Operate" und "Monitor". Der Einsatz von Machine Learning verläuft recht gleichmäßig bis auf einen leichten Rückgang im Bereich "Test" und einer leichten Erhöhung im Bereich "Operate" und "Monitor".

KI-Methoden nach Phase in %

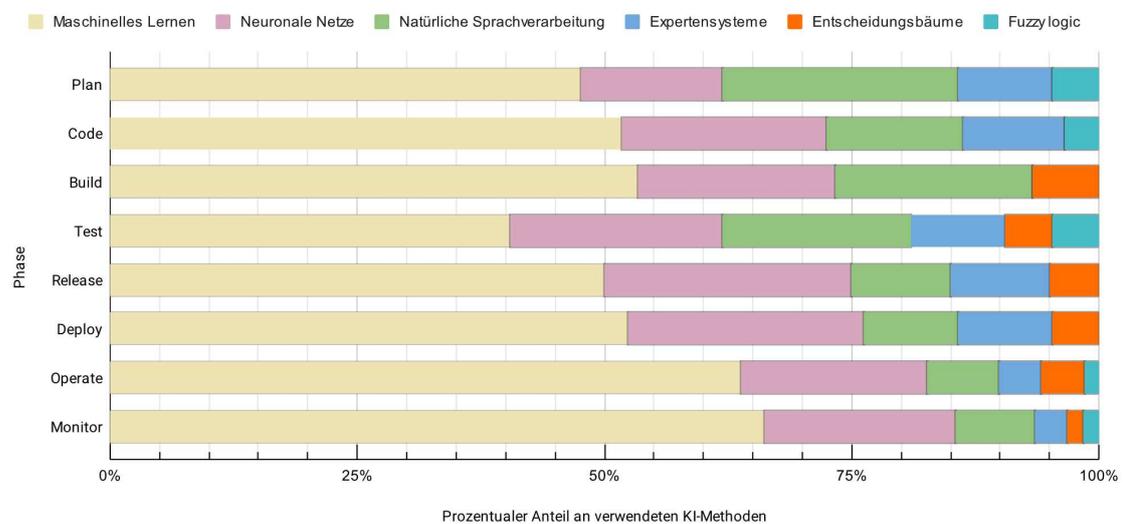
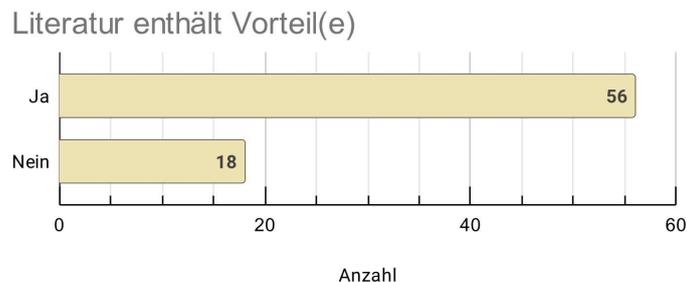


Abbildung 7.10: KI-Methoden nach Phase in %

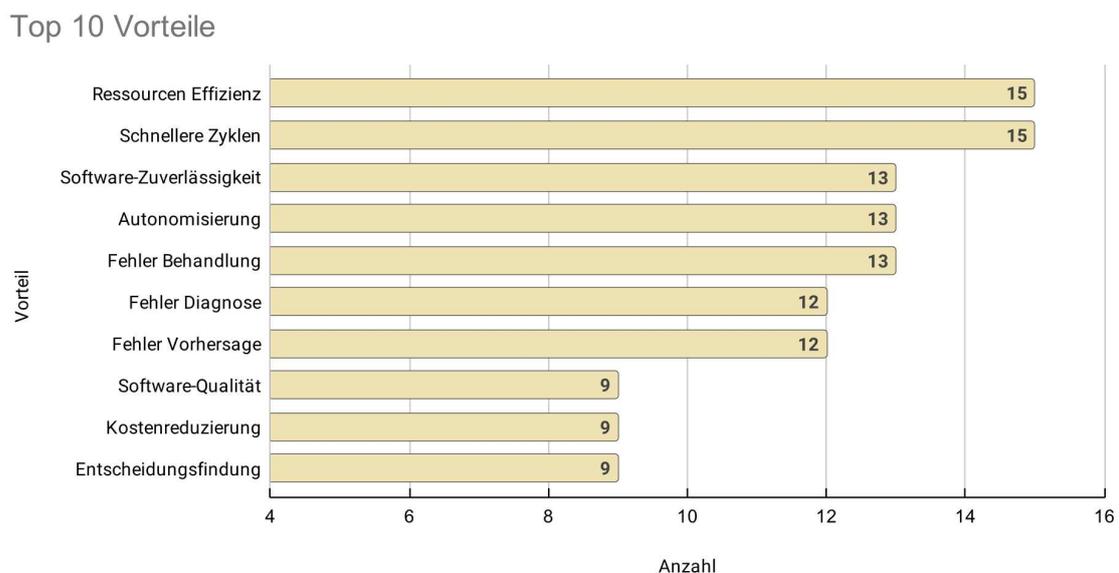
## 7.7 Vorteile

Das Diagramm 7.11 gibt an wie viele Arbeiten zudem auch Vorteile angeben. Hier ist klar zu erkennen, dass ein Großteil der Arbeiten Vorteile enthält. Insgesamt enthalten grob 75% der analysierten Arbeiten Vorteile. Das sind um einiges mehr als der Anteil an Arbeiten welche Herausforderungen enthalten, dieser Anteil beläuft sich grob auf 38%.



**Abbildung 7.11:** Anzahl der Literatur mit angeführten Vorteilen von KI-Integration

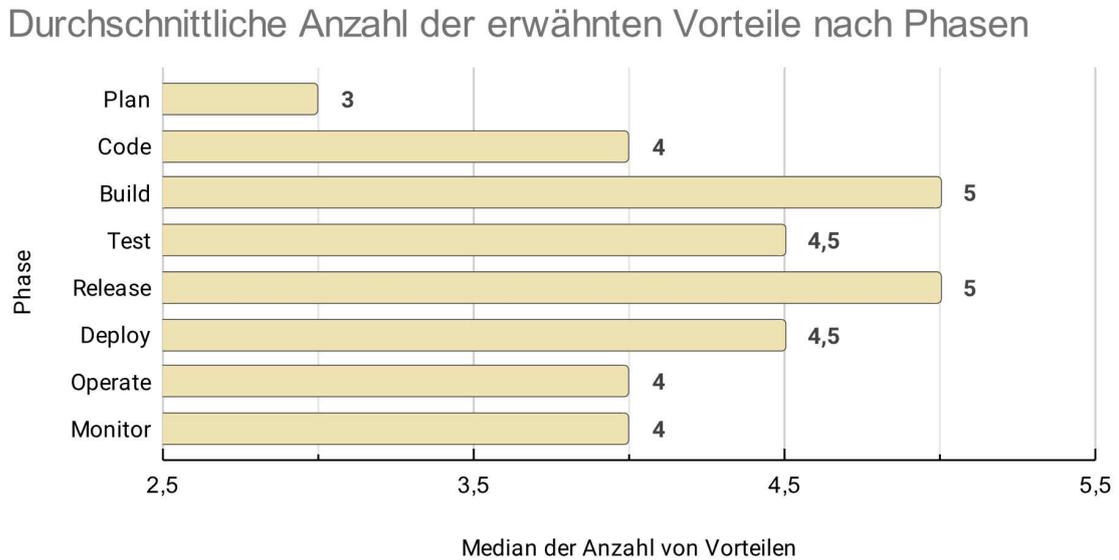
Insgesamt ergaben sich dabei eine enorme Liste von Vorteilen, welche Versucht wurde möglichst zu gruppieren. Die Top 10 meist erwähnten Bereiche, im Bezug auf Vorteile, sind in Diagramm 7.12 dargestellt. Dabei ist der Punkt "Ressourcen Effizienz" mit 15 Stück und der Punkt "Schnellere Zyklen" ebenfalls mit 15 Stück an der Spitze. "Ressourcen Effizienz" beinhaltet Vorteile wie eine bessere Verteilung der Last im System durch KI, als auch das automatisieren von repetitiven Aufgaben, was zu Zeit und Ressourcen Einsparung führen kann. "Schnellere Zyklen" beschreibt den Vorteil der Möglichkeit von schnelleren Phasen im DevOps Prozess, darunter fällt eine schnellere Entwicklung, Lösungsfindung, schnelleres Testing und einige mehr.



**Abbildung 7.12:** Top 10 in der Literatur erwähnten Vorteile

### 7.7.1 Durchschnitt Anzahl Vorteile - je Phase

Das Diagramm 7.13 zeigt den Median der erwähnten Vorteile je Phase. Um zu vermeiden, dass wenige Antworten in Phasen das Ergebnis beeinflussen wurde hier statt dem dem Durchschnitt, der Median genommen. Das Diagramm zeigt, dass die Anzahl der erwähnten Vorteile je Phase nicht stark differieren und



**Abbildung 7.13:** Median der Anzahl der erwähnten Vorteile nach Phasen

## 7.8 KPIs

Insgesamt wurden wie in folgender Tabelle ersichtlich, in 23 Arbeiten KPIs oder sonstige Metriken erwähnt.

Die in den Arbeiten erwähnten KPIs sind in folgender Aufzählung aufgelistet. Dabei wurden allerdings oftmals auch Metriken zur Validierung von Modellen angeführt, da dies auch eine große Herausforderung darstellt.

Tabelle 7.1: In der Literatur verwendete KPIs und Metriken

Metrik	Anzahl
Accuracy	2
Alert-Effektivität	1
Anzahl Incidents	1
AUC Score	4
Ausgelöste Warnungen	1
Code-Abweichung	1
Confusion Matrix	1
CORR (correlation coefficient)	1

Continued on next page

Tabelle 7.1: (Continued)

CPU-Auslastung	2
Designfehler-Erkennungszeit	1
Diagnosezeit / F-0.5 Score	1
Alarmverteilung	1
F1-score	5
Falsch-Positiv-Rate (FPR)	1
Fehler Vorhersage	1
Fehlerquote	1
Gefundene Ursachen	1
Incident-Problem-Verhältnis	1
Inferenzdauer	1
Intelligenter Filter vs. Unüberwachtes ML	1
Konsistenz-Messungen	1
Latenz	1
MAE	2
MAPE	1
ME	1
Mean Reciprocal Rank (MRR)	1
MPE	1
MSE	1
MTTD	1
MTTR	4
MTTT	1
Network Usage	1
Netzwerkdurchsatz	1
nRMSE	1
page view counts	1
Performancebewertung: Filter und Teamentlastung	1
Precision	5
Predicted daily alert notifications	1
Problembhebungszeit	1

Continued on next page

Tabelle 7.1: (Continued)

Recall	4
RMSE	1
Service Verfügbarkeit	1
Service-Reaktionszeit	1
Service-spezifische Warnverteilung	1
Speicherverbrauch	2
Success-to-Fail Ratio	1
Support-Ticket Anzahl	1
TTD	1
TTE	1
TTM	1
Ursachenidentifikationszeit	1
Vorhersage vs. tatsächliche Email Alerts	1
Warnmeldungen: Regeln vs. Intelligenter Filter	1
zyklomatische Komplexität	1

## 8 CONCLUSIO

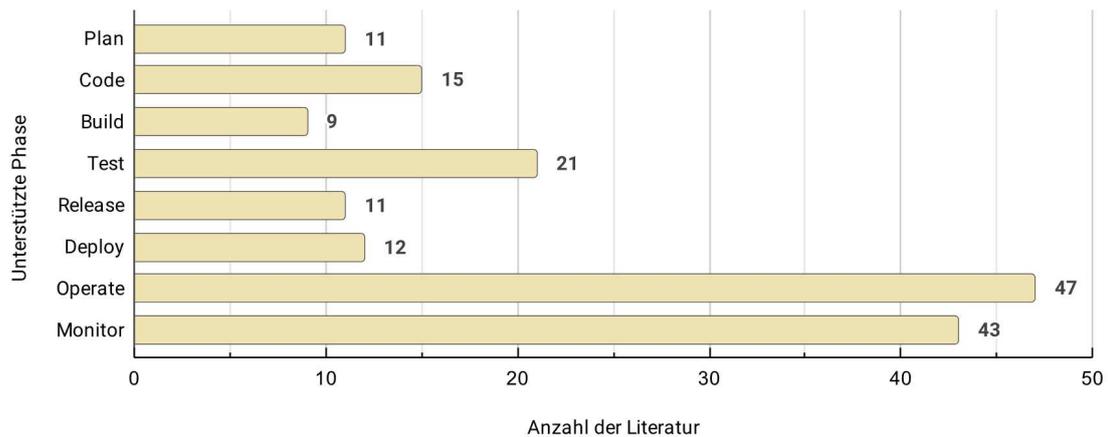
Der vorliegende Abschnitt zielt darauf ab, die Ergebnisse, die im vorangegangenen Kapitel 7 präsentiert wurden, zusammenzufassen und umfassend zu interpretieren. Ein weiterer Fokus liegt auf der eingehenden Beantwortung der Forschungsfrage sowie sämtlicher aufgestellter Hypothesen im Verlauf dieses Abschnitts.

### 8.1 Interpretation

In diesem Kapitel geht es darum die gewonnenen empirischen Daten einem tieferen Verständnisprozess zu unterziehen. Es zielt darauf ab, über die bloße Darstellung von beobachteten Mustern und Zusammenhängen hinauszugehen und die zugrunde liegenden Mechanismen und Implikationen zu enthüllen. Diese Phase betont die Notwendigkeit einer umfassenden Analyse, um die Ergebnisse in einen theoretischen Rahmen zu setzen und somit neue Perspektiven und Erkenntnisse für das spezifische Forschungsgebiet zu generieren.

Die Abbildung 8.1 zeigt eine stark erhöhte Anzahl an Literaturen im Bereich Operate und Monitor. Gründe dafür könnte sein, dass dieser Bereich stärker erforscht ist als andere. Allerdings kann es auch sein, dass die gewählte Literatur durch ein gewähltes Stichwort mehr Arbeiten in diesem Gebiet hervorgebracht hat. Ebenfalls kann ein Grund sein, dass die "Operate" und "Monitor" Phase viele Einsatzmöglichkeiten für maschinelle Analysen bietet, da in diesen Phasen typischerweise eine enorme Menge an Daten bereitsteht. Ursprung dieser großen Datenmengen sind in der Regel, sinnvolles Monitoring zu betreiben und im laufenden Betrieb Kennzahlen wie Performance, Down-Time, Auslastung und andere auszulesen. Andererseits entsteht durch diverse Infrastruktur und eingesetzte Software auch eine große Menge an LOG Daten, welche typischerweise enorme Größen annehmen.

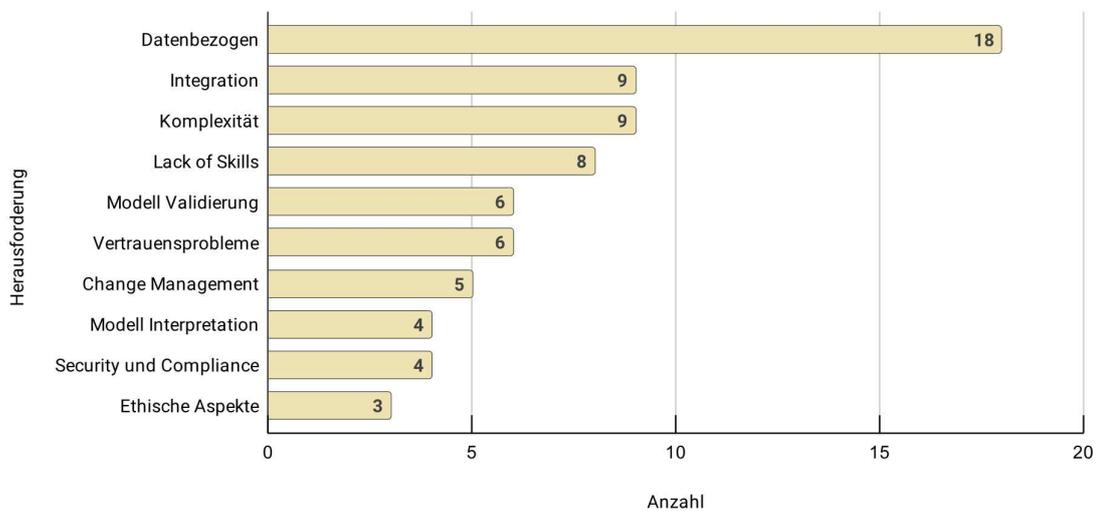
### Anzahl der Literatur nach den einzelnen DevOps Phasen



**Abbildung 8.1:** Unterstützte DevOps Phasen nach Anzahl der Ergebnisse

Gründe für die nach Abbildung 8.2 ersichtliche extreme Erwähnung der Daten-bezogenen Herausforderungen könnte folgende sein. In Daten-reichen Phasen wie "Monitor" ist es oftmals schwierig, dass alle Daten eine vorgeschriebene Qualität und Konsistenz erreichen, was für eine automatisierte Verarbeitung besonders wichtig ist. Andererseits gibt es in Phasen wie "Plan" bei weitem weniger Daten, da diese Phase unter anderem sehr Menschen-bezogen ist.

### Mögliche Herausforderungen - Top 10



**Abbildung 8.2:** Mögliche Herausforderungen - Top 10

Dies könnte auch ein Grund für den in Abbildung 8.3 ersichtlichen verstärkten Einsatz von "Natürliche Sprachverarbeitung" (NLP) sein. NLP selbst beschäftigt sich damit die menschliche Sprache für den Computer verständlich zu machen. Da "Plan" unter anderem viel mit Menschen-formulierten Anforderungen Zutun hat, könnte daher der Einsatz von NLP sehr hilfreich sein. Einige der analysierten Studie, welche die "Plan" Phase unterstützten erwähnten Anforderungen und den Versuch mit Anforderungen mit KI verständlicher zu formulieren.

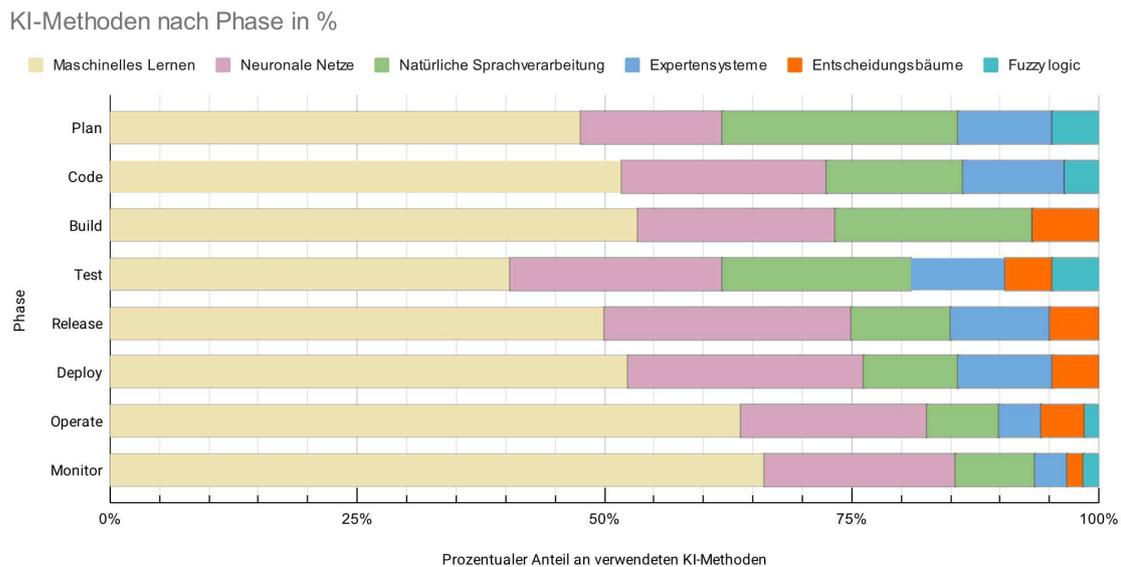


Abbildung 8.3: KI-Methoden nach Phase in %

## 8.2 Hypothesenprüfung

Dieser Abschnitt konzentriert sich darauf, wie die gesammelten Daten und die durchgeführte Analyse dazu beigetragen haben, die anfänglichen Hypothesen zu testen und zu bewerten. Dabei wird die systematische Überprüfung jedes Hypothesenansatzes im Kontext der gesammelten empirischen Evidenz präsentiert. Dieser Prozess ermöglicht nicht nur die Beurteilung der Hypothesen in Bezug auf ihre Validität, sondern bietet auch Einblicke in die Komplexität und Tiefe des untersuchten Themas.

### 8.2.1 H1

Die aufgestellte Hypothese H1 inklusive Gegenhypothese lautet wie folgt:

**H1<sub>0</sub>:** Durch die Integration von KI im DevOps Prozess entstehen keine neuen Herausforderungen bzw. Probleme in Bezug auf die Bereiche Sicherheit, Ethik und Verantwortlichkeit der KI Anwendungen.

**H1<sub>1</sub>:** Wenn KI im DevOps Prozess integriert wird, entstehen neue Herausforderungen für die Sicherheit, Ethik und Verantwortlichkeit der KI Anwendungen.

#### 8.2.1.1 Beantwortung

Wie in der Grafik 8.4 deutlich dargestellt wird, nehmen Sicherheit und Ethik einen Platz unter den Top 10 ein, was ihre Relevanz in diesem Kontext unterstreicht. Auffällig ist jedoch, dass die explizite Erwähnung der 'Verantwortlichkeit der KI-Anwendung' fehlt, obwohl Sicherheit und Ethik als bedeutsam identifiziert werden.

Die Feststellung, dass Sicherheit und Ethik prominent sind, während die explizite Erwähnung der 'Verantwortlichkeit der KI-Anwendung' fehlt, könnte darauf hinweisen, dass diese möglicherweise nicht als eigenständige Aspekte betrachtet, sondern als integraler Bestandteil von Sicherheits- und Ethikfragen angesehen werden. Das könnte den Hypothesen **H1<sub>0</sub>** und **H1<sub>1</sub>** eine komplexere Dimension verleihen, die näherer Untersuchung bedarf, um mögliche Zusammenhänge und Implikationen besser zu verstehen.

### Mögliche Herausforderungen - Top 10

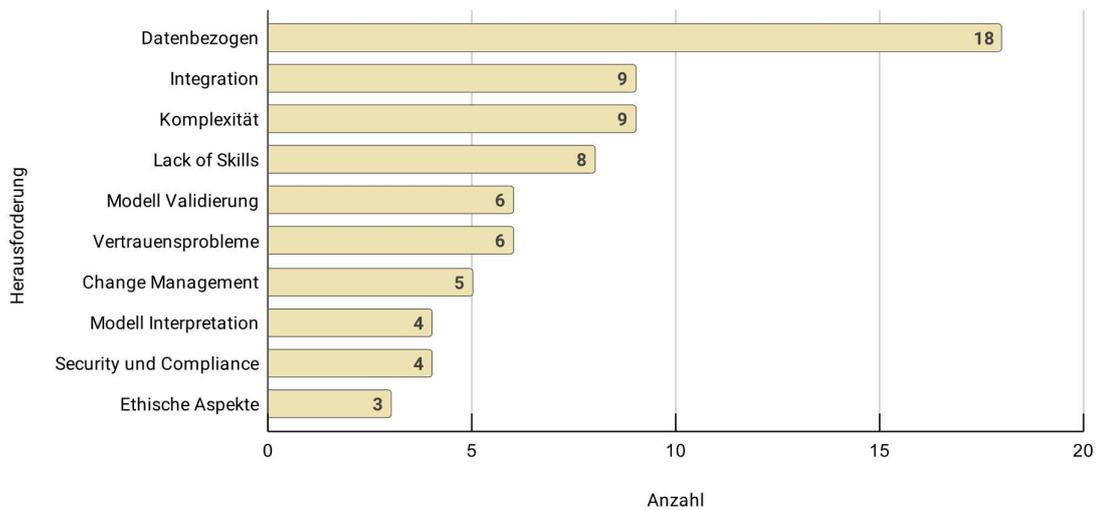


Abbildung 8.4: Mögliche Herausforderungen - Top 10

## 8.2.2 H2

Die aufgestellte Hypothese H2 inklusive Gegenhypothese lautet wie folgt:

**H2<sub>0</sub>:** Wenn KI für Monitoring im DevOps Prozess eingesetzt wird, hat das keinen signifikanten Einfluss auf die Erkennung und Behebung von Anomalien in Softwareprodukten.

**H2<sub>1</sub>:** Wenn KI für Monitoring im DevOps Prozess wird, hat das einen signifikanten positiven Einfluss auf die Erkennung und Behebung von Anomalien in Softwareprodukten.

### 8.2.2.1 Beantwortung

Die Hypothese **H2<sub>1</sub>** scheint gestützt zu werden, da das Diagramm 8.5 auf einen Zusammenhang zwischen dem Einsatz von KI im Monitoring im DevOps-Prozess und der verbesserten Fehlererkennung und -behebung hinweist. Konkret zeigen die Daten, dass in 79,3 % der Fälle, in denen verbesserte Fehlererkennung oder -behebung als Vorteil erwähnt wird, die Monitoring-Phase als unterstützende Phase identifiziert wird. Dies deutet darauf hin, dass der Einsatz von KI im Monitoring tatsächlich einen positiven Einfluss auf die Erkennung und Behebung von Anomalien in Softwareprodukten hat, was der Hypothese **H2<sub>1</sub>** entspricht.

### Erwähnt Vorteile im Bereich Fehlererkennung und -behebung

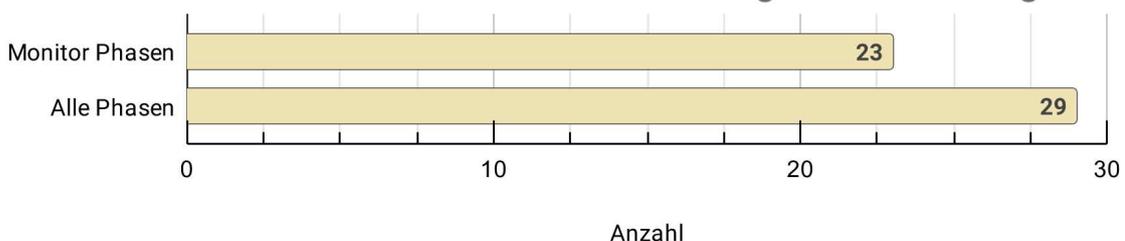


Abbildung 8.5: Vorteile im Bereich Fehlererkennung und -behebung, Monitoring Phase im Vergleich

### 8.2.3 H3

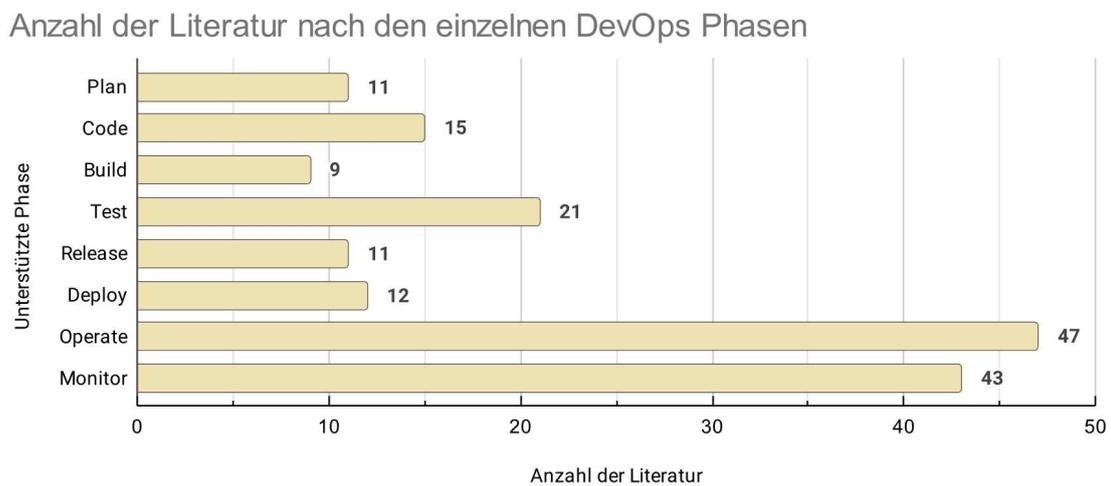
Die aufgestellte Hypothese H3 inklusive Gegenhypothese lautet wie folgt:

**H3<sub>0</sub>**: Es kann nicht jeder Teil des DevOps Prozesses, wie Planung, Entwicklung, Testen, Bereitstellung und Überwachung, von KI unterstützt werden.

**H3<sub>1</sub>**: Jeder Teil des DevOps Prozesses, wie Planung, Entwicklung, Testen, Bereitstellung und Überwachung, kann von KI unterstützt werden.

#### 8.2.3.1 Beantwortung

Die Hypothese **H3<sub>1</sub>** wird definitiv von dem Diagramm 8.6 gestützt. Dies wie bereits mehrmals erwähnt zeigt die Verteilung der Integration entlang der Phasen. Die Verteilung ist wie in Kapitel 8.1 beschrieben. Auch wenn die Betriebs-Phasen mehr Literatur enthalten gibt es für jede Phase einige Arbeiten die KI-Unterstützungen erwähnen.



**Abbildung 8.6:** Unterstützte DevOps Phasen nach Anzahl der Ergebnisse

### 8.3 Schlussfolgerung auf die Forschungsfrage

Durch die Zusammenführung der gesammelten Daten, Analysen und Erkenntnisse zielt diese Schlussfolgerung darauf ab, eine klare und fundierte Antwort auf die ursprüngliche Forschungsfrage zu präsentieren. Hier werden die Ergebnisse in Bezug auf die ursprüngliche Zielsetzung der Studie kontextualisiert und die Bedeutung dieser Ergebnisse für das breitere Forschungsfeld herausgearbeitet. Dieser Abschnitt ermöglicht nicht nur eine Bestätigung oder Ablehnung der anfänglichen Fragestellung, sondern auch die Diskussion über potenzielle Auswirkungen und zukünftige Entwicklungen aufgrund der erzielten Erkenntnisse.

In der angegebenen Auflistung 8.7 werden alle Phasen mit den Top 3 in der Literatur erwähnten KI-Methode, KI-Technik und Herausforderung.

Plan		
Methoden	Techniken	Herausforderungen
Maschinelles Lernen	Anomaly Detection	Integration
Natürliche Sprachverarbeitung	Root Cause Analysis	Lack of Skills
Neuronale Netze	Datenintegration	Datenbezogen

Code		
Methoden	Techniken	Herausforderungen
Maschinelles Lernen	Anomaly Detection	Datenbezogen
Neuronale Netze	Root Cause Analysis	Integration
Natürliche Sprachverarbeitung	Predictive ALM	Komplexität

Build		
Methoden	Techniken	Herausforderungen
Maschinelles Lernen	Anomaly Detection	Datenbezogen
Neuronale Netze	Root Cause Analysis	Komplexität
Natürliche Sprachverarbeitung	Datenintegration	Integration

Test		
Methoden	Techniken	Herausforderungen
Maschinelles Lernen	Anomaly Detection	Datenbezogen
Neuronale Netze	Root Cause Analysis	Integration
Natürliche Sprachverarbeitung	Predictive ALM	Komplexität

Release		
Methoden	Techniken	Herausforderungen
Maschinelles Lernen	Anomaly Detection	Datenbezogen
Neuronale Netze	Root Cause Analysis	Lack of Skills
Natürliche Sprachverarbeitung	Predictive ALM	Integration

Deploy		
Methoden	Techniken	Herausforderungen
Maschinelles Lernen	Anomaly Detection	Datenbezogen
Neuronale Netze	Root Cause Analysis	Lack of Skills
Natürliche Sprachverarbeitung	Predictive ALM	Integration

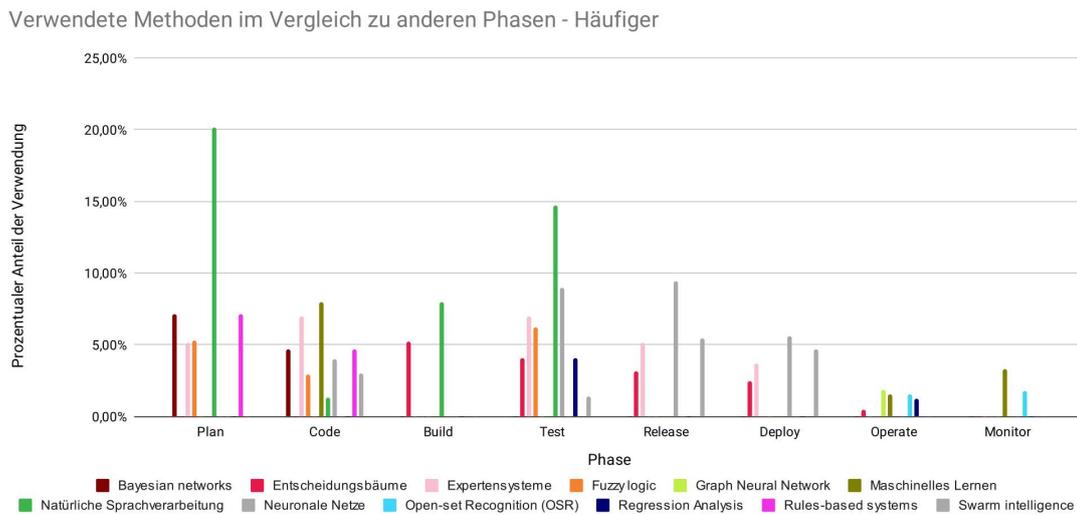
Operate		
Methoden	Techniken	Herausforderungen
Maschinelles Lernen	Anomaly Detection	Datenbezogen
Neuronale Netze	Root Cause Analysis	Integration
Natürliche Sprachverarbeitung	Predictive ALM	Lack of Skills

Monitor		
Methoden	Techniken	Herausforderungen
Maschinelles Lernen	Anomaly Detection	Datenbezogen
Neuronale Netze	Root Cause Analysis	Integration
Natürliche Sprachverarbeitung	Predictive ALM	Lack of Skills

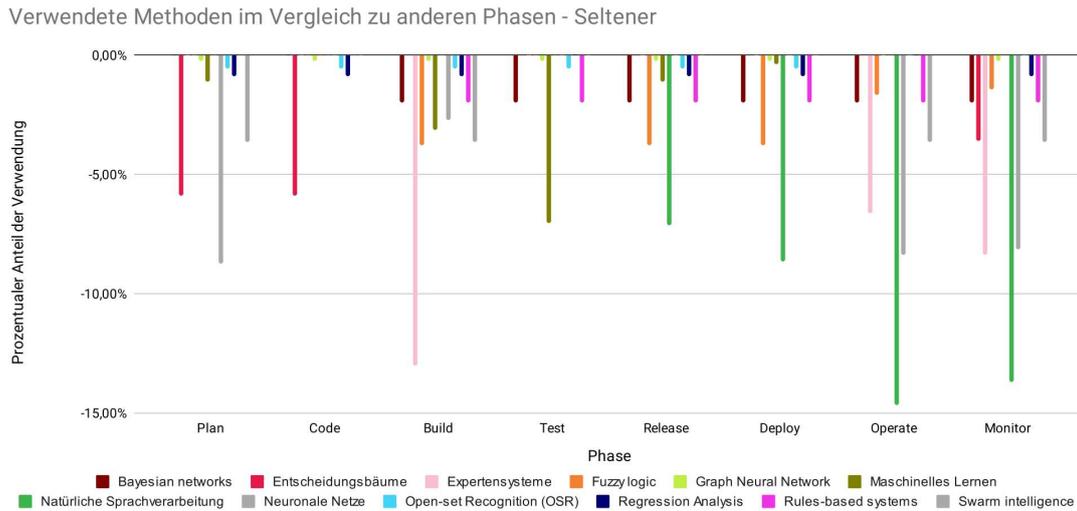
Abbildung 8.7: Beantwortung der Forschungsfrage

Da allerdings die Auswertung der meist verwendeten Technologie wenig aufschlussreich ist und keine der Methoden besonders heraus sticht wurde anhand des Diagrams 8.8 die Verwendung der KI-Methoden gegeneinander verglichen. Dafür wurde der prozentuale Anteil der Methoden pro Phase errechnet und dieser im Anschluss mit dem Durchschnitt über alle Phasen verglichen. Das Diagramm zeigt also die Häufigkeit der Verwendung pro Phase im Vergleich zum Rest der Phasen. Hierbei ist gut ersichtlich, dass in der Phase "Plan" und "Test" NLP deutlich öfter zum Einsatz kommt als im Durchschnitt. Ebenfalls erkennt man, dass die Verwendung von neuronalen Netzen vorallem in der "Test" und "Release" Phase öfter verwendet wird als in den restlichen Phasen. Dies könnte ebenfalls auf eine passende Methode für diese Phasen hinweisen.



**Abbildung 8.8:** Verwendete KI-Methoden der einzelnen Phasen im Vergleich zum Durchschnitt (Darstellung Positiv-Werte)

Das Diagramm 8.9 zeigt hier auch noch das Gegenteil. Es zeigt welche der KI-Methoden im Vergleich zum Durchschnitt seltener in den Phasen verwendet wird. Hierbei ist gut ersichtlich, dass in den ITOps Phasen NLP deutlich seltener zum Einsatz kommt als im Durchschnitt. Dies könnte darauf hinweisen, dass NLP für diese Phasen nicht optimal ist. Auch sieht man sehr stark, dass Expertensystem in der "Build" Phase nicht sehr häufig verwendet wird. Das könnte ebenfalls auf eine unpassende Methode hinweisen.



**Abbildung 8.9:** Verwendete KI-Methoden der einzelnen Phasen im Vergleich zum Durchschnitt (Darstellung Negativ-Werte)

### 8.3.1 Handlungsempfehlungen basierend auf dem Ergebnis

Basierend auf der umfassenden Analyse der am häufigsten verwendeten KI-Methoden, darunter Maschinelles Lernen, Natürliche Sprachverarbeitung und Neuronale Netze, in sämtlichen Phasen des Prozesses, zeigt sich die zentrale Bedeutung der Integration als wiederkehrende Herausforderung. Die Implementierung von Techniken wie Anomaly Detection, Root Cause Analysis, Datenintegration und Predictive ALM unterstreicht ebenfalls deren kritische Rolle in allen Phasen des Prozesses. Um diesem Muster zu begegnen, ist es entscheidend, integrative Ansätze zu entwickeln, die sich spezifisch auf die Herausforderungen der Integration konzentrieren, um die Effizienz und Wirksamkeit der KI-Integration zu steigern.

Darüber hinaus deutet die häufige Nennung von Herausforderungen wie dem Mangel an Fähigkeiten, datenbezogenen Problemen und der Komplexität darauf hin, dass eine gezielte Entwicklung von Kompetenzen und Fertigkeiten in Bezug auf Datenanalyse und KI von entscheidender Bedeutung ist. Die Bewältigung dieser Herausforderungen erfordert strategische Initiativen zur Verbesserung der Kompetenzen und Schulungen, um ein tieferes Verständnis für die Datenverarbeitung und KI-Methoden zu entwickeln. Dies würde die Organisationen dabei unterstützen, besser auf die spezifischen Anforderungen in jeder Phase des Prozesses einzugehen.

Um diese Herausforderungen anzugehen und die Vorteile der KI-Integration zu maximieren, sollte ein verstärkter Fokus auf die Implementierung von Maschinellem Lernen, Natürlicher Sprachverarbeitung und Neuronalen Netzen entlang des gesamten Prozesses liegen. Diese KI-Methoden könnten als Instrumente dienen, um datenbezogene Probleme zu lösen und gleichzeitig eine effizientere Verarbeitung und Analyse während der verschiedenen Phasen zu ermöglichen. Diese zielgerichtete Implementierung könnte helfen, den Nutzen der KI-Integration zu steigern und gleichzeitig die identifizierten Herausforderungen zu bewältigen.

## **8.4 Kritische Betrachtung**

Dieser Abschnitt zielt darauf ab, über die Oberfläche der präsentierten Erkenntnisse hinauszugehen und die methodischen Herangehensweisen, potenzielle Limitationen sowie alternative Perspektiven eingehend zu beleuchten. Durch eine kritische Reflexion der methodischen Entscheidungen und des Erkenntnisgewinns wird eine umfassende Einsicht in die Robustheit und Validität der Ergebnisse angestrebt. Diese Phase eröffnet die Möglichkeit, die Tiefe und den Kontext der präsentierten Erkenntnisse zu untersuchen und liefert einen Beitrag zur wissenschaftlichen Diskussion und Weiterentwicklung des Forschungsfeldes.

### **8.4.1 Methode**

Die gewählte Methodik der Systematischen Literaturübersicht (SLR) erwies sich grundsätzlich als solide Auswahl für die Erfassung und Analyse relevanter Studien. Allerdings stellte sich bei der Übertragung der gewonnenen Erkenntnisse in die Matrixdarstellung nach Webster und Watson (2002) eine gewisse Herausforderung dar, insbesondere aufgrund der Beschaffenheit der Antworten, die häufig in Form von Freitexten vorlagen. Die Matrixdarstellung nach Webster und Watson (2002) ist ursprünglich auf eine Struktur mit wenigen Single- oder Multi-Choice-Antwortmöglichkeiten ausgelegt, was die direkte Übertragung von Freitextantworten in diese Form erschwerte. Dies führte zu einer gewissen Unklarheit oder Einschränkung bei der adäquaten Einordnung und Darstellung der gesammelten Informationen innerhalb dieser spezifischen Matrixstruktur. Es wurde deutlich, dass die Effektivität dieser Darstellungsform bei einer Vielzahl von Freitextantworten begrenzt war und möglicherweise alternative Ansätze oder Anpassungen erforderte, um die Breite und Tiefe der gesammelten Erkenntnisse adäquat widerzuspiegeln.

### **8.4.2 Fragebogen**

Der entwickelte Fragebogen bot bereits eine solide Grundlage für die Datenerfassung, doch ein tiefergehendes Eingehen auf die spezifischen Aspekte der verwendeten Künstlichen Intelligenz (KI) hätte die Forschungsfrage in ihrem Umfang erweitern können. Die Integration von umfassenderen Informationen bezüglich der angewandten KI-Methoden und -Algorithmen hätte dazu beigetragen, einen detaillierteren Einblick in die Funktionsweise und die Anwendbarkeit der KI auf das spezifische Forschungsgebiet zu gewinnen. Eine erweiterte Beschreibung dieser Methoden hätte möglicherweise ein tieferes Verständnis der Zusammenhänge ermöglicht und die Möglichkeiten zur Ableitung von wertvolleren Erkenntnissen erweitert. Somit hätte eine ausführlichere Betrachtung der angewandten KI-Methodik einen erweiterten Mehrwert für die Beantwortung der Forschungsfrage und für potenzielle Anwendungen in diesem Bereich geschaffen.

### **8.4.3 Limitation**

Die Systematische Literaturübersicht (SLR) ist effektiv, kann jedoch Studien aufgrund stark beeinflussender Keywords übersehen. Die Abhängigkeit von bestimmten Schlüsselbegriffen könnte relevante Arbeiten ausschließen, die nicht exakt mit den verwendeten Keywords übereinstimmen.

Des Weiteren kann die Auswahl bestimmter Datenbanken oder Fachzeitschriften das Ergebnis beeinflussen und unbeabsichtigte Verzerrungen verursachen. Es besteht die Möglichkeit, dass die Begrenzung auf spezifische Datenbanken oder Journale zu einer ungewollten Voreingenommenheit führt, da relevante Arbeiten, die außerhalb dieser ausgewählten Quellen liegen, möglicherweise nicht berücksichtigt werden.

## 8.5 Weitere Forschungsmöglichkeiten

Mögliche Vertiefungen gäbe es im Bereich der praktischen Implementierung, Ethik und Governance, Performance-Metriken, Adaptivität und Skalierbarkeit, Human-KI-Kollaboration.

**Praktische Implementierung:** Erforschung der tatsächlichen Umsetzung von KI-Integrationen in DevOps/IT Operations, um Einblicke in Herausforderungen und Erfolge bei der praktischen Anwendung zu gewinnen.

**Ethik und Governance:** Untersuchung der ethischen Implikationen, Governance-Strukturen und regulatorischen Anforderungen, die mit der Integration von KI in DevOps/IT Operations verbunden sind.

**Performance-Metriken:** Analyse spezifischer Leistungsindikatoren und Metriken, um die tatsächliche Wirksamkeit von KI-Integrationen zu bewerten und ihren Einfluss auf Effizienz, Qualität oder Kosten zu quantifizieren.

**Adaptivität und Skalierbarkeit:** Untersuchung, wie sich KI-Integrationen in DevOps/IT Operations an veränderliche Anforderungen und Skalierbarkeit anpassen, insbesondere in sich verändernden oder schnell wachsenden Umgebungen.

**Human-KI-Kollaboration:** Analyse der optimalen Zusammenarbeit zwischen menschlicher Expertise und KI-Technologien, um Synergien zu maximieren und effiziente Arbeitsabläufe zu gestalten.

## LITERATUR

- Aebi, D. (2004). 1 Grundlagen und Begriffe. In *Praxishandbuch Sicherer it-Betrieb Risiken Erkennen, Schwachstellen beseitigen, it-Infrastrukturen schützen* (S. 1–27). Betriebswirtschaftlicher Verlag Dr. Th. Gabler. (Siehe S. 20, 21, 81).
- Alégroth, E. (2015). Visual GUI Testing: Automating High-level Software Testing in Industrial Practice (siehe S. 12, 13).
- Altunel, H., & Say, B. (2021). Software Product System Model: A Customer-Value Oriented, Adaptable, DevOps-Based Product Model. *SN Computer Science*, 3(1), 38 (siehe S. 8).
- Anne-Wil Harzing. (o. D.). *Publish or Perish* (Version 8). (Siehe S. 41, 42, 81).
- Azad, N., & Hyrynsalmi, S. (2023). DevOps critical success factors — A systematic literature review. *Information and Software Technology*, 157, 107150 (siehe S. 5, 6, 81).
- Baltes, S., Knack, J., Anastasiou, D., Tymann, R., & Diehl, S. (2018). (No) Influence of Continuous Integration on the Commit Activity in GitHub Projects (siehe S. 18).
- Bharadiya, J. (2023). Machine Learning and AI in Business Intelligence: Trends and Opportunities. *International Journal of Computer (IJC)*, 123–134 (siehe S. 29).
- Brocke, J. v., Simons, A., Niehaves, B., Riemer, K., Plattfaut, R., & Cleven, A. (2009). Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature Search Process. <http://www.alexandria.unisg.ch/Publikationen/67910> (siehe S. 35–39, 81).
- Brown, A., & Hellerstein, J. (2005). Reducing the Cost of IT Operations - Is Automation Always the Answer? (Siehe S. 23, 24).
- Bryan, F. (2021). *Repeatability, Reliability, and Scalability Through GitOps : Continuous Delivery and Deployment Codified*. Packt Publishing. (Siehe S. 18).
- Charak, P. (2023, Juli). *Role of Continuous Monitoring in DevOps Pipeline*. <https://www.pcloudy.com/blogs/role-of-continuous-monitoring-in-devops-pipeline/> (accessed: 23.10.2023). (Siehe S. 13).
- Cook, D. A. (2019). Systematic and Nonsystematic Reviews: Choosing an Approach. In D. Nestel, J. Hui, K. Kunkler, M. W. Scerbo & A. W. Calhoun (Hrsg.), *Healthcare Simulation Research: A Practical Guide* (S. 55–60). Springer International Publishing. (Siehe S. 34, 39).
- Cooper, H. M. (1988). Organizing knowledge syntheses: A taxonomy of literature reviews. *Knowledge in Society*, 1(1), 104 (siehe S. 40).
- Dang, Y., Lin, Q., & Huang, P. (2019). AIOps: Real-World Challenges and Research Innovations. *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 4–5 (siehe S. 32).

- Dhakad, K. (2023, Februar). *Adopting Continuous Integration Practices to Achieve Quality in DevOps*. (Siehe S. 9).
- Faber, F. (2020). Testing in devops. *The Future of Software Quality Assurance*, 27–38 (siehe S. 8).
- Farley, D., & Humble, J. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley Professional. (Siehe S. 11).
- Fowler, M., & Foemmel, M. (2006). Continuous integration. (Siehe S. 11).
- Ghaleb, T. A., da Costa, D. A., & Zou, Y. (2019). An empirical study of the long duration of continuous integration builds. *Empirical Software Engineering*, 24(4), 2102–2139 (siehe S. 18).
- Halstenberg, J., Pfitzinger, B., & Jestädt, T. (2020). *DevOps Ein Überblick*. Springer Vieweg. (Siehe S. 19).
- Hanson-Abromeit, D. (2014). The systematic review as a research process in music therapy. *Journal of music therapy*, 51 1, 4–38 (siehe S. 34).
- Hermansyah, M., Najib, A., Farida, A., Sapiro, R., & Rintyarna, B. (2023). Artificial Intelligence and Ethics: Building an Artificial Intelligence System that Ensures Privacy and Social Justice. *International Journal of Science and Society*, 5, 154–168 (siehe S. 27, 28).
- Heß, J. (2021). Strukturierte Literaturrecherche – ein Rahmenkonzept. In *Regionale Erfolgsfaktoren entlang des Gründungsprozesses: Arten und Einfluss regionaler Erfolgsfaktoren im Start-up-Prozess* (S. 125–139). Springer Fachmedien Wiesbaden.
- Hollier, M., & Wagner, A. (2017). Chapter 1: Defining Continuous Testing. In M. Hollier & A. Wagner (Hrsg.), *Continuous Testing for dummies* (S. 3–13). IBM Press. (Siehe S. 13).
- Karamitsos, I., Albarhami, S., & Apostolopoulos, C. (2020). Applying DevOps Practices of Continuous Automation for Machine Learning. *Information*, 11(7) (siehe S. 9).
- Kerravala, Z. (2021). AIOps Provides a Path To Fully Autonomous Networks. *eWeek*, N.PAG (siehe S. 32).
- Knight, L. (2020, Oktober). *The Role of DevOps Feedback Loops in Your Value Stream*. <https://devops.com/the-role-of-devops-feedback-loops-in-your-value-stream/> (accessed: 20.10.2023). (Siehe S. 14).
- Krancher, O., Luther, P., & Jost, M. (2018). Key Affordances of Platform-as-a-Service: Self-Organization and Continuous Feedback. *Journal of Management Information Systems*, 35(3), 776–812 (siehe S. 19).
- Kumar, N. A., & Gondkar, R. R. (2021). Role of ITOps in DevOps. *Social Science Research Network*.
- Len Bass, L. Z., Ingo Weber. (2015). *DevOps: A software architect's perspective*. Addison-Wesley Professional. (Siehe S. 4, 5, 81).
- Li, Y., Jiang, Z., Li, H., Hassan, A. E., He, C., Huang, R., Zeng, Z., Wang, M., & Chen, P. (2020). Predicting Node Failures in an Ultra-Large-Scale Cloud Computing Platform: An AIOps Solution. *ACM Transactions on Software Engineering and Methodology*, 29 (siehe S. 31, 32).
- Linthicum, D. (2021). What's the Real Value of AIOps?. *eWeek*, N.PAG (siehe S. 31).
- López-Fernández, D., Díaz, J., García, J., Pérez, J., & González-Prieto, Á. (2022). DevOps Team Structures: Characterization and Implications. *IEEE Transactions on Software Engineering*, 48(10), 3716–3736 (siehe S. 8, 9).
- Marrone, M., & Kolbe, L. M. (2011). Impact of IT Service Management Frameworks on the IT Organization. *Business & Information Systems Engineering*, 3, 5–18 (siehe S. 22).

- McCarthy, J., Minsky, M., Rochester, N., & Shannon, C. (1955). A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955. *AI Magazine*, 27, 12–14 (siehe S. 25).
- Mohammad, S. M. (2019). DevOps Automation Advances I.T. Sectors with the Strategy of Release Management. *SSRN Electronic Journal*, 67, 82–88 (siehe S. 24).
- Mohan, S. (2022). *What is CICD Pipeline? Explanation of CICD Pipeline along with Examples*. <https://razorops.com/blog/what-is-cicd-pipeline-explanation-of-cicd-pipeline-along-with-examples/> (accessed: 09.10.2023).
- Moore, S. (2019, März). How to Get Started With AIOps. (Siehe S. 31).
- Nielsen, P., Winkler, T., & Nørbjerg, J. (2017). Closing the IT Development-Operations Gap: The DevOps Knowledge Sharing Framework (siehe S. 20–22).
- Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., ... Moher, D. (2021). The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, 372, n71 (siehe S. 43–46, 48, 50, 81).
- Pannu, A. (2015). Artificial intelligence and its application in different areas. *Artificial Intelligence*, 4(10), 79–84 (siehe S. 26).
- Pati, D. (2018). How to Write a Systematic Review of the Literature. *HERD: Health Environments Research & Design Journal*, 11, 15–30 (siehe S. 34).
- Pennington, J. (2019). *The Eight Phases of a DevOps Pipeline*. <https://medium.com/taptuit/the-eight-phases-of-a-devops-pipeline-fda53ec9bba> (accessed: 09.10.2023). (Siehe S. 16, 17).
- Perri, L. (2023, August). What's New in Artificial Intelligence from the 2023 Gartner Hype Cycle. (Siehe S. 28).
- Pittet, S. (2023). *Continuous integration vs. delivery vs. deployment*. <https://www.atlassian.com/continuous-delivery/principles/continuous-integration-vs-delivery-vs-deployment> (accessed: 09.10.2023). (Siehe S. 17).
- PlanetScale. (2023a, März). *The Deploy phase of the DevOps cycle*. <https://planetscale.com/docs/devops/the-deploy-phase-of-devops> (accessed: 10.10.2023). (Siehe S. 19).
- PlanetScale. (2023b, April). *The release phase of the DevOps cycle*. <https://planetscale.com/docs/devops/the-release-phase-of-devops> (accessed: 10.10.2023). (Siehe S. 18).
- Rathore, B. (2023). Future of AI and Generation Alpha: ChatGPT beyond Boundaries. *Eduzone : international peer reviewed/refereed academic multidisciplinary journal*, 12, 63–68 (siehe S. 30).
- RedHat. (2022, Mai). Was ist eine CI/CD-pipeline? (Siehe S. 10, 81).
- Senapathi, M., Buchan, J., & Osman, H. (2018). DevOps Capabilities, Practices, and Challenges. *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018* (siehe S. 6).
- Serrano, J., Faustino, J., Adriano, D., Pereira, R., & da Silva, M. M. (2021). An IT Service Management Literature Review: Challenges, Benefits, Opportunities and Implementation Practices. *Information*, 12(3) (siehe S. 22, 23, 81).
- Shahin, M., Ali Babar, M., & Zhu, L. (2017). Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices. *IEEE Access*, PP (siehe S. 7).

- Srivastava, S. (2023). Optimizing Automation and Specialized Testing Techniques in DevOps (siehe S. 6, 7, 9, 12).
- Srivastava, S., & Bhosale, S. (2023). DevSecOps in Cloud-Native and Serverless Architectures (siehe S. 15, 16).
- Teich, I. (2020). Meilensteine der Entwicklung Künstlicher Intelligenz. *Informatik Spektrum* (siehe S. 25).
- Thantharate, P. (2023). IntelligentMonitor: Empowering DevOps Environments with Advanced Monitoring and Observability, 800–805 (siehe S. 13, 14).
- Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind, New Series*, 59, 433–460 (siehe S. 25).
- Volkmar, G., Reinecke, S., & Fischer, P. M. (2021). Künstliche Intelligenz im Marketing: Möglichkeiten und Herausforderungen. *Die Unternehmung / Swiss Journal of Business Research and Practice*, 75(3), 359–375 (siehe S. 28).
- Webster, J., & Watson, R. T. (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MIS Quarterly*, 26(2), xiii–xxiii (siehe S. 37, 38, 55, 75, 81).
- Wiles, J. (2022, September). Neues aus dem Gartner Hype Cycle 2022 zum Thema KI. (Siehe S. 26–29).
- Yaraghi, A. S., Bagherzadeh, M., Kahani, N., & Briand, L. C. (2023). Scalable and Accurate Test Case Prioritization in Continuous Integration Contexts. *IEEE Transactions on Software Engineering*, 49(4), 1615–1639 (siehe S. 18).
- Zampetti, F., Geremia, S., Bavota, G., & Di Penta, M. (2021). CI/CD Pipelines Evolution and Restructuring: A Qualitative and Quantitative Study, 471–482 (siehe S. 10).

## ABBILDUNGSVERZEICHNIS

2.1	DevOps Phasen adaptiert aus Len Bass (2015)	5
2.2	DevOps Phasen adaptiert aus Azad und Hyrynsalmi (2023)	6
2.3	Vereinfachte CI/CD Pipeline adaptiert von RedHat (2022)	10
2.4	Acht CI/CD Phasen (Pennington, 2019)	17
3.1	Komponenten der Infrastruktur nach Aebi (2004)	21
3.2	Zusammenhänge der ITSM Herausforderungen übersetzt aus Serrano et al. (2021)	23
4.1	HypeCycle für KI (Wiles, 2022)	29
6.1	5 Phasen systematische Literaturrecherche (Brocke et al., 2009)	35
6.2	Charakteristik - Auswahl (Brocke et al., 2009)	36
6.3	Prozess der Literatursuche (Brocke et al., 2009)	37
6.4	Matrixdarstellung Literaturanalyse (Webster & Watson, 2002)	38
6.5	SLR Ausschnitt aus der Konzept-Mindmap	41
6.6	Ausschnitt einer Google Scholar Suchanfrage über "Publish or Perish" von Anne-Wil Harzing (o. D.)	42
6.7	"Publish or Perish" Suchdefinition AIOps	43
6.8	Aktualisierter PRISMA Prozess von Page et al. (2021)	44
6.9	Auf diese Masterarbeit angepasster PRISMA Prozess von Page et al. (2021)	45
6.10	Prozess Screening Phase	47
6.11	Diagramm Ausschlusskriterien	47
6.12	Verteilung der Ausschlusskriterien mit Hervorhebung der Top 4 Herausgeber, relativ zur Anzahl der Gesamtpublikationen	48
6.13	Anzahl der Fachbücher innerhalb der Eligibility Phase	49
6.14	Anzahl der ausgeschlossenen Literatur innerhalb der Eligibility Phase	49
6.15	Anzahl der entfernten Arbeiten, gruppiert nach der SLR Phase	50
6.16	Vergleich der Anzahl der Quellen, gruppiert nach Publisher nach den Filter-Phasen	51
7.1	Anzahl der zur weiteren Behandlung geeigneten Literatur	56
7.2	Unterstützte DevOps Phasen nach Anzahl der Ergebnisse	56
7.3	Anzahl der Literatur mit angeführten Herausforderungen von KI-Integration	57
7.4	Unterstützte DevOps Phasen nach Anzahl der Ergebnisse	57
7.5	Kategorie der Herausforderungen nach Phase in %	58

7.6	Mögliche Herausforderungen - Top 10 . . . . .	59
7.7	Anzahl der Literatur welche zu den angeführten Herausforderungen auch Lösungsansätze enthält . . . . .	59
7.8	Top 5 verwendete KI-Techniken . . . . .	61
7.9	Top 6 verwendete KI-Methoden . . . . .	62
7.10	KI-Methoden nach Phase in % . . . . .	62
7.11	Anzahl der Literatur mit angeführten Vorteilen von KI-Integration . . . . .	63
7.12	Top 10 in der Literautr erwähnten Vorteile . . . . .	63
7.13	Median der Anzahl der erwähnten Vorteile nach Phasen . . . . .	64
8.1	Unterstützte DevOps Phasen nach Anzahl der Ergebnisse . . . . .	68
8.2	Mögliche Herausforderungen - Top 10 . . . . .	68
8.3	KI-Methoden nach Phase in % . . . . .	69
8.4	Mögliche Herausforderungen - Top 10 . . . . .	70
8.5	Vorteile im Bereich Fehlererkennung und -behebung, Monitoring Phase im Vergleich . . . . .	70
8.6	Unterstützte DevOps Phasen nach Anzahl der Ergebnisse . . . . .	71
8.7	Beantwortung der Forschungsfrage . . . . .	72
8.8	Verwendete KI-Methoden der einzelnen Phasen im Vergleich zum Durchschnitt (Darstellung Positiv-Werte) . . . . .	73
8.9	Verwendete KI-Methoden der einzelnen Phasen im Vergleich zum Durchschnitt (Darstellung Negativ-Werte) . . . . .	74

**TABELLENVERZEICHNIS**

7.1 KPIs und Metriken . . . . . 64

## LITERATURSUCHE-QUELLEN

- Aghili, R., Li, H., & Khomh, F. (2022). Studying the Characteristics of AIOps Projects on GitHub.
- Ahmed, S., Singh, M., & Bucholc, M. (2023). An evaluation of BERT applied for AIOps.
- Alhakeem, M. (2023). Role of AIOps in Petroleum industry; A Letter to editor.
- Alnafessah, A., Gias, A., Wang, R., Zhu, L., Casale, G., & ... (2021). Quality-aware devops research: Where do we stand?
- An, L., Tu, A., Liu, X., & Akkiraju, R. (2022). Realtime Statistical Log Anomaly Detection with Continuous AIOps Learning.
- Andenmatten, M. (2019). AIOpsArtificial Intelligence for IT Operations: Todays Challenges of new Technologies and new Methodologies in IT Operations.
- Arya, V., Shanmugam, K., Aggarwal, P., & Wang, Q. (2021). Evaluation of causal inference techniques for AIOps.
- Barabash, K., Carrozzo, G., Lorenz, D., & Meth, K. (2021). Zerotouch aiops in multi-operator 5g networks.
- Barenkamp, M. (2020). Applications of AI in classical software engineering.
- Batarseh, F. (2015). Predicting failures in agile software development through data analytics.
- Batarseh, F. A. (2021). The Application of Artificial Intelligence in Software Engineering – A Review Challenging Conventional Wisdom.
- Battina, D. (2016). AI-Augmented Automation for DevOps, a Model-Based Framework for Continuous Development in Cyber-Physical Systems.
- Battina, D. (2017). Best practices for ensuring security in devops: A case study approach.
- Battina, D. (2019). An intelligent devops platform research and design based on machine learning.
- Battina, D. (2020). Devops A New Approach To Cloud Development Testing.
- Battina, D. (2021). Ai and devops in information technology and its future in the united states.
- Becker, S., Schmidt, F., & Gulenko, A. (2020). Towards aiops in edge computing environments.
- Bhateja, N. (2017). ACHIEVING QUALITY IN AUTOMATION of SOFTWARE TESTING USING AI BASED TECHNIQUES.
- Bheri, S., & Vummenthala, S. (2019). An Introduction to the DevOps Tool Related Challenges.
- Calefato, F. (2023). Assessing the Use of AutoML for Data-Driven Software Engineering.
- Chagas, J. (2020). On the Reuse of Knowledge to Develop Intelligent Software Engineering Solutions.
- Cheng, Q., Sahoo, D., Saha, A., Yang, W., & Liu, C. (2023). AI for IT Operations (AIOps) on Cloud Platforms: Reviews, Opportunities and Challenges.
- Colantoni, A. (2020). DevOpsML towards modeling DevOps processes and platforms.

- Davis, J. (2016). A taxonomy of artificial intelligence approaches for adaptive distributed real-time embedded systems.
- Diamantopoulos, T. (2015). Towards Interpretable Defect-Prone Component Analysis using Genetic Fuzzy Systems.
- Dias, H. P. (2017). Automated use case diagram generator using NLP and ML.
- Dias, T. (2023). TestLab An Intelligent Automated Software Testing Framework.
- Dong, W. (2022). AIOps Architecture in Data Center Site Infrastructure Monitoring.
- Fawzy, A., Wassif, K., & Moussa, H. (2023). Framework for automatic detection of anomalies in DevOps.
- Feldt, R. (2018). Ways of Applying Artificial Intelligence in Software Engineering.
- Gaikwad, R., Deshpande, S., & Vaidya, R. (2021). A framework design for algorithmic it operations (aiops).
- GHEORGHIOĂ, A., & Petre, I. (2020). Securely driving IoT by integrating AIOps and blockchain.
- Harman, M. (2012). The Role of Artificial Intelligence in Software Engineering.
- Hrusto, A., Runeson, P., & Engström, E. (2021). Closing the feedback loop in devops through autonomous monitors in operations.
- Hua, Y. (2021). A Systems Approach to Effective AIOps Implementation.
- Imam, A. (2015). The Definition of Intelligent Computer Aided Software Engineering (I-CASE) Tools.
- Khaliq, Z. (2022). Artificial Intelligence in Software Testing : Impact, Problems, Challenges and Prospect.
- Kulkarni, N. (2012). Automated Analysis of Textual Use-Cases: Does NLP Components and Pipelines Matter?
- Kumar, D., & Sampath, A. (2022). Orchestration of ML/AI models using MLOps/AIOps frameworks.
- Levin, A., Garion, S., & Kolodner, E. (2019). AIOps for a cloud object storage service.
- Li, J., Tan, F., He, C., Wang, Z., Song, H., & Wu, L. (2022). HigeNet A Highly Efficient Modeling for Long Sequence Time Series Prediction in AIOps.
- Li, Y., Jiang, Z., Li, H., Hassan, A., & He, C. (2020). Predicting node failures in an ultra-large-scale cloud computing platform: an aiops solution.
- Li, Z., Zhao, N., Zhang, S., Sun, Y., Chen, P., & Wen, X. (2022). Constructing large-scale real-world benchmark datasets for AIOps.
- Lyu, Y., Li, H., Sayagh, M., & Jiang, Z. (2021). An empirical study of the impact of data splitting decisions on the performance of AIOps solutions.
- Lyu, Y., Rajbahadur, G., Lin, D., & Chen, B. (2021). Towards a consistent interpretation of aiops models.
- Maddila, C. (2022). Recommender Systems for DevOps.
- Masri, R. A. (2021). Applications of Dynamic Heterogeneous Information Networks for AIOps.
- Mboweni, T., Masombuka, T., & ... (2022). A Systematic Review of Machine Learning DevOps.
- Moens, P., Andriessen, B., Sebrechts, M., & Volckaert, B. (2023). Edge Anomaly Detection Framework for AIOps in Cloud and IoT.
- Monperrus, M. (2017). Automatic Software Repair: a Bibliography.
- Naik, M., & Mallya, R. (2018). DevOps IN INTEGRATION WITH AI.
- Notaro, P., Cardoso, J., & Gerndt, M. (2021). A survey of aiops methods for failure management.
- Paradkar, S. (2020). APM to AIOps-Core Transformation.
- Pavlek, A. (2022). Validierung Vergleich und Bewertung verschiedener AIOps-Modelle und-Algorithmen zur Steigerung der Problem-und Anomalieerkennung basierend auf Daten aus ...

- Pendy, B. (2022). Role of AIOps in Health Care Units.
- Reiter, L., & Wedel, F. (2021). AIOpsA Systematic Literature Review.
- Remil, Y. (2021). How can subgroup discovery help AIOps?
- Saha, A., & Hoi, S. (2022). Mining root cause knowledge from cloud service incident investigations for AIOps.
- Said, B. (2022). Towards AIDoArT Objectives via Joint Model-based Architectural Effort.
- Shankari, K. (2014). A Survey on Using Artificial Intelligence Techniques in the Software Development Process.
- Singh, J., & Shyamsundar, S. (2019). AIOps FRAMEWORK FOR ALERTING PERFORMANCE ISSUES IN MICROSERVICES USING TIME SERIES FORECASTING.
- Sorte, B. W. (2015). Use of Artificial Intelligence in Software Development Life Cycle: A state of the Art Review.
- Strandberg, P. (2021). Ethical AI-Powered Regression Test Selection.
- Teggi, P., Malakreddy, B., & Teggi, P. (2021). AIOps Prediction for Server Stability Based on ARIMA Model.
- Todd, A., Purkayastha, A., Egan, H., Sickinger, D., & Eash, M. (2021). Artificial Intelligence for Data Center Operations (AIOps).
- Wang, H., & Zhang, H. (2020). AIOps prediction for hard drive failures based on stacking ensemble model.
- Wang, J. (2023). Software Testing with Large Language Model: Survey, Landscape, and Vision.
- Wang, K., Tan, Y., Zhang, L., Chen, Z., & Lei, J. (2022). A Network Traffic Prediction Method for AIOps Based on TDA and Attention GRU.
- Wu, Z., Xu, H., Pang, G., Yu, F., Wang, Y., & Jian, S. (2021). Dram failure prediction in aiops: Empirical evaluation, challenges and opportunities.
- Yang, L., & Rossi, D. (2021). Quality monitoring and assessment of deployed deep learning models for network AIOps.
- Yeruva, A. (2023a). Monitoring Data Center Site Infrastructure Using AIOps Architecture.
- Yeruva, A. (2023b). Providing A Personalized Healthcare Service To The Patients Using AIOps Monitoring.
- Zhong, Z., Fan, Q., Zhang, J., Ma, M., Zhang, S., & Sun, Y. (2023). A Survey of Time Series Anomaly Detection Methods in the AIOps Domain.