



hochschule mannheim



# Praxissemesterbericht

<b>Autor</b>	Rebekka Hahn
<b>Matrikelnummer</b>	1921861
<b>Semester</b>	11. Semester
<b>Studiengang</b>	Medizintechnik
<b>Beginn Praxissemester</b>	02.09.2024
<b>Ende Praxissemester</b>	28.02.2025
<b>Firma</b>	Löwenstein medical
<b>Betreuer</b>	Patrick von Poblitzki, Christoph Elsner

Ludwigshafen am Rhein, 27. Dezember 2024

## **Selbständigkeitserklärung**

Ich versichere, dass ich diesen PS-Bericht selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe. Die Stellen, an denen Inhalte aus den Quellen verwendet wurden, sind als solche eindeutig gekennzeichnet. Die Arbeit hat in gleicher oder ähnlicher Form bei keinem anderen Prüfungsverfahren vorgelegen.

---

Datum, Ort und Unterschrift

## BESCHEINIGUNG / *EMPLOYER'S CONFIRMATION*

der Ausbildungsstelle über die betriebliche Ausbildung  
während des Praktischen Studienseesters (PS) /  
*of the total number of days worked during an internship (practical semester)*

.....  
Nachname, Vorname des/der Studierende(n) / *student's last name, first name*

.....  
geboren am / *date of birth* in / *place of birth*

.....  
Matrikelnummer / *matriculation number*

Student/Studentin der Hochschule Mannheim, Fakultät für Informationstechnik/  
*Student of the University of Applied Sciences Mannheim*

hat in der Zeit vom / *completed from*

..... bis / *to* .....

die betriebliche Ausbildung des Praktischen Studienseesters abgeleistet im Bachelorstudiengang/  
*an internship in the*

- ☐ Medizintechnik (MTB) / *Medical Engineering*
- ☐ Nachrichtentechnik/Elektronik (NEB) / *Communication/Electronic Engineering*
- ☐ Technische Informatik (TIB) / *Computer Engineering*
- ☐ Mechatronik (MEB) / *Mechatronic Engineering*

Die regelmäßige Wochenarbeitszeit in Stunden betrug / *The standard hours worked per week was*  
..... Std. / *hours*

☐ Während der o. a. Zeit entstanden keine Fehlzeiten / *The student completed the required number of working days.*

Während der o. a. Zeit entstanden Fehlzeiten durch / *There was a shortfall during the working time because of*

..... Tage Urlaub / *days of holiday*  
..... Tage Krankheit / *days of absence owing to illness*  
..... Tage sonstige Abwesenheit / *days of absence due to other reasons*

Die Anzahl der Präsenztage beträgt / *Total number of days worked* .....

.....  
Ort / Datum Firmen-/Institutsname (Stempel), Adresse, Unterschrift  
Place / Date Employer's name (stamp), address, signature

## Abkürzungsverzeichnis

Abkürzung	Ausgeschrieben
APAP	automatic positive airway pressure
CPAP	continuous positive airway pressure
EEG	Elektroenzephalographie
ISO	(von gr. isos = dt. gleich) Internationale Organisation für Normung
IVDR	In-vitro diagnostic regulation
MDR	Medical device regulation
OSA	obstruktive Schlafapnoe
REM	rapid eye movement
RL	Reinforcement Learning
SBAS	schlafbezogenen Atmungsstörungen
SWS	slow wave sleep
PAP	positive airway pressure
TE	Tonsillektomie
UPPP	Uvulopalatopharyngoplastik

## **Abstract**

sudo make abstract

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>6</b>
<b>2</b>	<b>Löwenstein Medical</b>	<b>7</b>
2.1	Geschichte und Entwicklung . . . . .	7
2.2	Produkte . . . . .	8
2.3	Beispielgerät der Schlafatemtherapie . . . . .	9
2.3.1	Physiologie Schlaf . . . . .	9
2.3.2	Pathophysiologie Schlafapnoe . . . . .	12
2.3.3	Therapie Schlafapnoe . . . . .	13
2.3.4	prisma Smart . . . . .	15
2.4	Qualitätsmanagement . . . . .	16
2.4.1	Regulatorischer Rahmen für Medizinprodukte . . . . .	16
2.4.2	Normen und Qualitätsstandards für Medizinprodukte . . . . .	17
2.5	Meetings . . . . .	17
<b>3</b>	<b>Aufgabe: Dokumentationsautomatisierung</b>	<b>18</b>
3.1	Polarion . . . . .	20
3.2	Software Qualitätssicherung . . . . .	21
3.3	Regular Expression . . . . .	21
<b>4</b>	<b>Aufgabe: LeRobot</b>	<b>22</b>
4.1	Machine Learning . . . . .	23
4.2	Imitation learning . . . . .	24
4.3	Reinforcement Learning . . . . .	24
4.4	Das Nutzen von LeRobot . . . . .	24
<b>5</b>	<b>Ergebnisse</b>	<b>27</b>
<b>6</b>	<b>Fazit</b>	<b>28</b>

# 1 Einleitung

Dieser Bericht fasst die Erfahrungen und Tätigkeiten zusammen, die ich während meines Praxissemesters bei Löwenstein Medical am Standort Karlsruhe sammeln konnte. Als Familienunternehmen im Bereich der Medizintechnik entwickelt und vertreibt Löwenstein Medical spezialisierte Beatmungsprodukte. Der Standort Karlsruhe hat bei der Entwicklung den Schwerpunkt Schlaftherapie, digitale Therapiebegleitung und Telehealth. Während meines Semesters war ich in der Firmware-Abteilung tätig und habe an einem Projekt zur Entwicklung eines Medizingerätes mitgearbeitet.

Im Rahmen des Berichts wird ein Beispielgerät für die Schlafapnoe-Therapie vorgestellt. Dabei wird die zugrunde liegende Pathophysiologie der obstruktiven Schlafapnoe sowie die spezifische Therapie-Funktionalität des Geräts erläutert. Diese Abschnitte sollen den Zusammenhang zwischen den theoretischen Grundlagen und der praktischen Anwendung verdeutlichen. Während meines Semesters war ich in der Firmware-Abteilung tätig und konnte wertvolle Einblicke in die Entwicklung eines Medizingeräts gewinnen.

Ziel dieses Berichts ist es, Einblicke in die Arbeitsweise und die speziellen Anforderungen der Firmware-Entwicklung in der Medizintechnik zu geben und die praktischen Erfahrungen zusammenzufassen, die ich in diesem professionellen Umfeld sammeln konnte.

## 2 Löwenstein Medical

Dieses Kapitel gibt einen umfassenden Überblick über die Entwicklung des Unternehmens, präsentiert ein Beispielgerät aus dem Bereich der Heim-Schlafatemtherapie und beleuchtet den wissenschaftlichen Hintergrund der obstruktiven Schlafapnoe. Dabei wird sowohl auf technische Innovationen als auch auf therapeutische Ansätze und regulatorische Aspekte eingegangen.

### 2.1 Geschichte und Entwicklung

Nach dem Einstieg von Reinhard Löwenstein bei der Firma Heinen entstand das Unternehmen Heinen + Löwenstein im Jahr 1986, das sich zunächst auf die Neonatologie (Lehre der Pathologie und Physiologie Neugeborener) spezialisierte. Im Jahre 1992 folgte die Einstieg in den Bereich Schlafmedizin. Zwei Jahre später, 1994, wurde das Tochterunternehmen Heinen + Löwenstein Medizinelektronik gegründet, mit einem Fokus auf Schlafdiagnostiksystemen.

Ein bedeutender Entwicklungsschritt erfolgte im Jahr 1999, als Löwenstein Medical seine Position als führender Anbieter für respiratorische Heimversorgung in Deutschland etablierte und sich die exklusiven Vertriebsrechte für Produkte von Respironics sicherte. Die strategische Partnerschaft mit Hamilton Medical im Jahr 2002 trug zur Verstärkung des Portfolios im Beatmungsbereich bei.

In den folgenden Jahren baute das Unternehmen kontinuierlich seine Produktpalette aus. Zwischen 2005 und 2006 erfolgte die Markteinführung der Anästhesiegeräte Leon plus und Leon sowie der Beatmungsgeräte Leoni 2 und Leoni plus, speziell für Früh- und Neugeborene. Der Aufbau einer Flüssigsauerstoff - Versorgungslogistik im Jahr 2008 und die Integration von SALVIA medical waren weitere wichtige Meilensteine für die Weiterentwicklung des Unternehmens.

Die internationale Expansion begann 2009 mit der Gründung von Löwenstein Medical Austria. In den folgenden Jahren entstanden Tochtergesellschaften in zahlreichen Ländern, darunter Belgien, Frankreich, China, Australien und den USA. 2013 wurde Weinmann Homecare zu Löwenstein Medical Technologies umfirmiert, Heinen + Löwenstein wurde zu Löwenstein Medical.

2014 wurde die High-End-Intensivbeatmungsserie elisa 800 und elisa 600 auf den Markt gebracht, gefolgt von den Turbinenbeatmungsgeräten elisa 300 und elisa 500 im Jahr 2019. Der Launch des außerklinischen Beatmungsgeräts LUISA im Jahr 2020 und des Neonatologiegeräts LEONIE 4 im Jahr 2023 erweitern das Produktsortiment. [1]



## 2.2 Produkte

Löwenstein Medical bietet ein umfangreiches Produktsortiment zur Unterstützung der Patientenversorgung in den Bereichen Intensivbeatmung, außerklinische Beatmung, Schlafatemtherapie und Sauerstoffversorgung.

**Intensivbeatmungsgeräte:** Die Geräte der elisa-Serie sind für den Einsatz in der Intensivmedizin konzipiert. Diese Reihe besitzt eine agile Systemarchitektur, sie ermöglicht eine einfache Anpassung an unterschiedliche klinische Anforderungen und Hausstandards, wodurch Bedienerfehler und Schulungsaufwand reduziert werden.

**Außerklinische Beatmung:** Die LUISA-Beatmungsgeräte sind konzipiert für die häusliche Beatmung. Sie unterstützen sowohl invasive als auch nicht-invasive Beatmung.

**Schlafatemtherapie:** Die Prisma-Therapiegeräte bieten eine Anpassung an die individuellen Bedürfnisse der Patienten mit Schlafapnoe. In Kapitel 2.3 wird näher auf den prismaSMART eingegangen.

**Schlafdiagnostik und Monitoring:** Das Samoa 12-Kanal Polygraphiegerät ist für die detaillierte Diagnostik von schlafbezogenen Atmungsstörungen ausgelegt. Das Samoa lite bietet eine vereinfachte Lösung für den Heimgebrauch und ermöglicht die Erkennung von Schlafapnoe sowie die kontinuierliche Überwachung von Atmung, Körperposition und anderen wichtigen Parametern.

**Softwarelösungen:** PrismaCLOUD ist eine Telemedizinlösung, die es ermöglicht, die Therapie remote zu überwachen und zu optimieren. Sie bietet eine sichere und flexible Möglichkeit, Patienteninformationen in Echtzeit zu erhalten und die Therapie entsprechend anzupassen. Mit der automatischen Übertragung von Therapiedaten und Compliance-Statistiken von den Geräten bietet prismaCLOUD eine einfache Handhabung für das medizinische Personal und unterstützt die Patienten bei einer effektiven Therapieüberwachung. •

## 2.3 Beispielgerät der Schlafatemtherapie

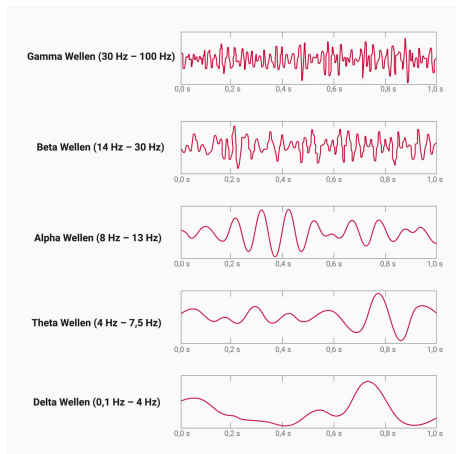
In der Behandlung der obstruktiven Schlafapnoe (OSA) sind verschiedene Therapieansätze von zentraler Bedeutung, um die Schlafqualität zu verbessern und gesundheitliche Risiken zu minimieren. Ein innovatives Gerät in der Therapie dieser schlafbezogenen Atmungsstörungen ist das prisma SMART. Es bietet die kontinuierlichen positiven Atemwegsdruck-Therapie (CPAP).

Im Folgenden werden zunächst die physiologischen Grundlagen des Schlafes sowie die pathophysiologischen Mechanismen der Schlafapnoe erläutert, um die Notwendigkeit einer gezielten Therapie und die Funktionsweise des prisma SMART in diesem Kontext zu verdeutlichen. Darüber hinaus wird die Therapie von Schlafapnoe, einschließlich moderner apparativer und konservativer Ansätze, thematisiert.

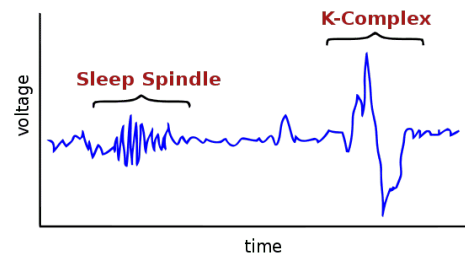
### 2.3.1 Physiologie Schlaf

Der Schlaf ist ein komplexer, dynamischer Zustand, der essenziell für die Erholung von Körper und Geist ist. Er spielt eine zentrale Rolle in zahlreichen physiologischen Prozessen, darunter die Regulation des Hormonhaushalts, die Gedächtniskonsolidierung und die zelluläre Regeneration. Darüber hinaus beeinflusst Schlaf die emotionale Stabilität, soziale Interaktion und Entscheidungsfindung, indem er die Funktion von Hirnarealen wie dem präfrontalen Cortex und der Amygdala reguliert. Ein Mangel an Schlaf hat weitreichende Auswirkungen auf das physische und psychische Wohlbefinden und kann das Risiko für chronische Krankheiten wie Herz-Kreislauf-Erkrankungen [2], Diabetes [3] und Depressionen [4] erhöhen [5]. Die Schlafphysiologie umfasst dabei ein fein abgestimmtes Zusammenspiel verschiedener Hirnregionen, Neurotransmitter und Hormone, die das Gleichgewicht zwischen Wachzustand und Schlaf sicherstellen.

In diesem Kapitel wird zunächst auf die verschiedenen Schlafphasen eingegangen und auf die allgemeine Bedeutung von Schlaf für den menschlichen Körper.



(a) Übersicht EEG-Signaltypen [6]



(b) Schlafspindeln und K-Komplex-Gehirnströme [7]

Abbildung 1: EEG Signale

Die Phasen des Schlafs unterteilen sich in zwei Hauptphasen, dem Non-REM-Schlaf (NREM) und dem REM-Schlaf (Rapid Eye Movement). Während des Schlafens ändern sich die Hirnströme, was mittels Elektroenzephalographie (EEG) zu erfassen ist. Das EEG misst die Potentialveränderung des Gehirns, welche die elektrische Aktivität des Gehirns darstellt und grafisch darstellbar ist, wie in Abbildung (1a) erkennbar. Die Abbildung zeigt beispielhaft verschiedene Frequenzbänder (EEG-Bänder). Ganz grob lassen sich die Bänder-Ausschläge beurteilen, wenn es nach unten ausschlägt, eine Positivierung, sind die tiefen Schichten erregt (erregendes postsynaptisches Potential) oder die oberflächlichen Schichten inhibiert (inhibitorisches postsynaptisches Potential), entsprechend beim ein Ausschlag nach oben, einer Negativierung, vice versa. [8]

Der NREM lässt sich in drei Stadien unterteilen (N1-N3), dazu noch die REM-Phase und der Wachzustand:

- **Wachphase**

Im EEG sind zunächst  $\beta$ -Wellen zu erkennen Abb. 1a, je schläfriger die Person wird, desto dominanter werden die  $\alpha$ -Wellen.

- **Stage 1: N1 - leichter Schlaf**

Macht 5 % der Schlafzeit aus und dauert ca. 1 - 5 Minuten. Es ist die Übergangsphase von wach zu schlafend. Das EEG zeigt  $\theta$ -Wellen Abb. 1a (niedrige Spannung), diese Phase beginnt, wenn mehr als 50% der  $\alpha$ -Wellen durch niederfrequente Wellen ersetzt wurden. Diese Phase ist die Übergangsphase und soll den Körper auf tiefere Schlafphasen vorbereiten.

- **Stage 2: N2 - tieferer Schlaf**

Die Dauer dieser Phase beginnt im ersten Schlafzyklus mit 25min und wird im Verlauf der Nacht länger, sie macht 45% der Schlafzeit aus. Das EEG zeigt K-Komplexe und Schlafspindeln, siehe Abbildung (1b). K-Komplexe sind biphasische Wellen hoher Amplitude ( $125-150\mu V$ ) und niedriger Frequenz(1-2Hz). K-Komplexe werden durch akustische Reize provoziert, sie spiegeln die Wahrnehmung von Umweltreizen wieder. Schlafspindeln haben eine hohe Frequenz (11-15Hz) und eine niedrige Amplitude ( $<50\mu V$ ), man nimmt einen schlafstabilisierenden Effekt an, da sie aus hemmenden Rückkopplungen der Nervenfasern entstehen, die vom Thalamus zur Großhirnrinde gehen. Die Schlafspindeln sind kennzeichnend für Schlaf. [8] [9]

In dieser Phase beginnt der Körper sich zu entspannen, Herzfrequenz und Körpertemperatur sinken.

- **Stage 3: N3 - tiefster, NREM Schlaf**

Wird auch slow-wave-sleep (SWS) genannt und beansprucht etwa 25% der Schlafzeit. Im EEG sind  $\delta$ -Wellen erkennbar Abb.1a, diese haben die niedrigste Frequenz und die höchste Amplitude. N3 ist eine sehr tiefe Schlafphase mit einer hohen Weckschwelle und ist essenziell für die systemische Regeneration des Körpers. Das beinhaltet die Reparatur und das Wachstum von Geweben und Stärkung des Immunsystems.

- **REM Schlaf**

Die Dauer der REM-Phase umfasst 25% der Schlafzeit, beginnt etwa 90min nach dem Einschlafen mit 10min Dauer und wird mit jedem Schlafzyklus länger, sie kann bis zu 1h dauern. Kennzeichnend für diese Phase ist die schnelle, unregelmäßige Bewegung der Augen, auch rapid eye movement. Die Skelettmuskulatur ist fast völlig erschlafft, während die Gehirnaktivität, der Blutdruck und die Herz- und Atemfrequenz erhöht sind. Das EEG zeigt vor allem  $\beta$ -Wellen Abb.1a, diese liegen bei 14-30Hz.  $\beta$ -Wellen treten neben der REM Phase bei wachen Personen auf, deren Kortex (Hirnrinde) gerade am Arbeiten ist. Im REM-Schlaf sollen sich Gedächtnisinhalte festigen. [8]

Ein Schlafzyklus läuft in der Reihenfolge N1 - N2 - N3 - REM ab und dauert 90-110min. In dem ersten Zyklus ist die REM-Phase noch kurz, aber wird mit jedem Zyklus zum Morgen hin länger. [10]

[5]

### 2.3.2 Pathophysiologie Schlafapnoe

Die obstruktive Schlafapnoe (OSA) gehört zu den schlafbezogenen Atmungsstörungen (SBAS) und ist eine Erkrankung die durch wiederholte Atempausen während des Schlafens gekennzeichnet ist. Es gibt vier verschiedene Phänotypen die eine OSA verursachen können:

5.1 Obstruktive Schlafapnoe Entsprechend der ICSD-3 [10] wird eine obstruktive Schlafapnoe (OSA) dann diagnostiziert, wenn die Atmungsstörung durch keine andere Schlafstörung oder medizinische Erkrankung oder durch Medikamente oder andere Substanzen erklärbar ist und entweder ein AHI > 15/h (Ereignis jeweils größergleich 10 s) Schlafzeit oder ein AHI größergleich 5/h Schlafzeit in Kombination mit einer typischen klinischen Symptomatik oder relevanten Komorbidität vorliegt. Tagesschläfrigkeit bis hin zum unfreiwilligen Einschlafen ist das führende klinische Symptom der obstruktiven Schlafapnoe –Hauptbefund Nächtliches Aufschrecken mit kurzzeitiger Atemnot, Schnarchen (bei 95prozent der Betroffenen), Isoliert betrachtet, weisen die Symptome jedoch nur eine geringe Spezifität auf –Nebenbefund → Definition OSA

- **Anatomische Einschränkungen der oberen Atemwege:**

Hierbei handelt es sich um strukturelle Faktoren, wie eine Verengung der Atemwege durch vergrößerte Mandeln, Fettansammlungen oder andere anatomische Besonderheiten. Diese Phänotypen sind häufig bei übergewichtigen oder älteren Menschen zu beobachten.

- **niedrige respiratorische Erregungsschwelle (Arousals):**

Eine niedrige Erregungsschwelle bedeutet, dass Personen während des Schlafs leichter durch Atemprobleme geweckt werden. Dies führt zu fragmentiertem Schlaf und verhindert eine kontinuierliche Atmung. Umgekehrt kann eine zu hohe Schwelle die Sauerstoffsättigung gefährlich abfallen lassen.

- **Instabilität des Atemantriebs („Loop Gain“):**

Diese Phänotypen beschreiben Menschen, deren Atmungssystem zu Überreaktionen neigt, was zu wechselnden Phasen von Hyperventilation und Hypoventilation führt. Dies verstärkt das Auftreten von Atempausen und Sauerstoffmangel.

- **schlechte Funktion der oberen Atemwegsmuskulatur:**

Hier liegt das Problem in einer unzureichenden Aktivierung oder Kontrolle der Muskeln, die die Atemwege während des Schlafs offenhalten sollten. Besonders während des REM-Schlafs, in dem der Muskeltonus generell abnimmt, kann dies zu Atemwegsblockaden führen.

Diese Phänotypen sind nicht immer isoliert, sondern treten oft in Kombination auf. Ein besseres Verständnis der individuellen Merkmale ermöglicht eine gezieltere Diagnostik und Therapie der OSA. So können gezielte Therapielösungen für die spezifischen Pathomechanismen entwickelt werden. [11] [12]

### 2.3.3 Therapie Schlafapnoe

Die Behandlung der obstruktiven Schlafapnoe (OSA) richtet sich nach dem individuellen Beschwerdebild, den Begleiterkrankungen sowie den persönlichen Bedürfnissen und dem Therapiewillen des Patienten. Ziel ist es, die schlafbezogenen Atmungsstörungen zu beseitigen, die Schlafqualität zu verbessern und das Risiko für kardiovaskuläre und andere Komplikationen zu senken. Abhängig von der Schwere der Erkrankung stehen verschiedene Therapieansätze zur Verfügung, die von konservativen Maßnahmen über apparative Unterstützung bis hin zu chirurgischen Eingriffen reichen. [13] [8]

- **apparative Therapie**

Die Standardtherapie der obstruktiven Schlafapnoe ist die nächtliche Überdruckbeatmung („positive airway pressure“, PAP), als die konkrete Referenzmethode im kontinuierlichen PAP-Modus (CPAP, „continuous positive airway pressure“). Die Indikationsstellung zur CPAP-Therapie erfolgt anhand einer Kombination aus klinischer Anamnese, polysomnographischem Befund und Begleiterkrankungen. Besonders wenn ohne Therapie eine Verschlechterung dieser Erkrankungen zu erwarten ist, wird eine CPAP-Therapie empfohlen. Der Therapiewille des Patienten sowie dessen individuelle Situation spielen ebenfalls eine entscheidende Rolle.

Neben CPAP kommt auch der APAP-Modus (automatisch titrierendes PAP) zum Einsatz, der den Atemwegsdruck flexibel an die Bedürfnisse des Patienten anpasst. Beide Ansätze zielen darauf ab, den Kollaps der oberen Atemwege zu verhindern und die Atmung während des Schlafes zu stabilisieren. Kontraindikation der APAP sind zentrale Atmungsstörungen, kardio- pulmonale Erkrankungen und nächtliche Hypoventilationen. Die APAP kommt vorallem zum Einsatz bei Patienten, die den kontinuierlichen Druck der CPAP nicht mehr ertragen, bei komplexen Apnoen oder mangelnder Compliance.

- **konservative Therapie**

Konservative Maßnahmen umfassen Lebensstiländerungen wie Gewichtsreduktion, die Vermeidung von Alkohol und Sedativa sowie das Einhalten einer guten Schlafhygiene. Bei lageabhängiger OSA kann Lagetraining, das eine Rückenlage vermeidet, ebenfalls hilfreich sein.

- **medikamentöse Therapie**

Medikamentöse Ansätze spielen bei der Behandlung der OSA nur eine untergeordnete Rolle, da bisher keine Substanzen eine ausreichende Wirksamkeit gezeigt haben.

- **chirurgische Therapie**

Operative Eingriffe werden nur bei spezifischen anatomischen Ursachen wie Tonsillenhypertrophie oder kraniofazialen Fehlbildungen in Betracht gezogen. Zu den Methoden zählen beispielsweise die Uvulopalatopharyngoplastik oder Kieferrekonstruktionen.

Die Uvulopalatopharyngoplastik mit Tonsillektomie nach Fujita, die TE-UPPP, ist ein chirurgisches Verfahren bei welchem die Atemwege durch Entfernung von überschüssigem Gewebe erweitert werden. Das Ziel des Eingriffs ist eine Verringerung der Kollapsneigung der oberen Atemwege während des Schlafens. Im Rahmen eines Case Reports war eine Indikation der OSA einen verengten oropharyngealen Raum mit vergleichsweise großer Uvula (Gaumenzäpfchen) und überschüssiger Mucosa (Schleimhaut) des umgebenden Gewebes. Die Tonsillektomie ist die vollständige chirurgische Entfernung der Tonsilla Palatina (Gaumenmandel). Anschließend wird redundante Mucosa entfernt und das Uvula korrigiert. [14]

### 2.3.4 prisma Smart

Das prisma SMART ist ein druckkontrolliertes, nicht-invasives Therapiegerät, das zur Behandlung schlafbezogener Atmungsstörungen (SBAS) wie der obstruktiven Schlafapnoe (OSA) mit Maske eingesetzt wird (Kapitel 2.3.2). Es nutzt den Auto-CPAP-Modus, der den Therapiedruck kontinuierlich anpasst, um die oberen Atemwege des Patienten offen zu halten und so Atemaussetzer während des Schlafes zu verhindern. Dieser Modus ist besonders effektiv bei der Behandlung der OSA, da er den Druck anpasst, um die Atemwege während des gesamten Schlafzyklus offen zu halten. Im Vergleich zu anderen Geräten wie dem CPAP-Modus, der einen konstanten Druck liefert, bietet der Auto-CPAP-Modus eine individuellere Anpassung (Kapitel 2.3.3).

Die physiologische Grundlage dieser Therapie beruht auf der Pathophysiologie der OSA (Kapitel 2.3.2), bei der es zu wiederholten Atempausen aufgrund eines Kollapses der oberen Atemwege kommt. Während des Schlafs, insbesondere in der REM-Phase, verringert sich der Muskeltonus, was das Risiko von Atemwegsverengungen erhöht. Der prisma SMART wirkt dem entgegen, indem er durch den erzeugten Überdruck die Atemwege stabilisiert, was die nächtlichen Atemaussetzer verhindert und eine ausreichende Sauerstoffversorgung sicherstellt. Dies fördert nicht nur die Atmung, sondern verbessert auch die Schlafqualität. [15]



Abbildung 2: prisma SMART [16]



## 2.4 Qualitätsmanagement

Die Medizinprodukteverordnung (Medical Device Regulation, MDR) regelt Produkte, die eine medizinische Zweckbestimmung aufweisen und vom Hersteller für den Einsatz beim Menschen vorgesehen sind. Dazu gehören auch In-vitro-Diagnostika (IVDR), die für die Untersuchung von Proben aus dem menschlichen Körper außerhalb des Körpers bestimmt sind. Weiterhin umfasst die Verordnung Produkte, die Stoffe oder Zubereitungen aus Stoffen enthalten oder damit beschichtet sind, welche bei gesonderter Anwendung als Arzneimittel oder Bestandteil eines Arzneimittels (einschließlich Plasmaderivate) klassifiziert werden und die eine pharmakologische, immunologische oder metabolische Wirkung auf den menschlichen Körper ausüben können. [17]

### 2.4.1 Regulatorischer Rahmen für Medizinprodukte

**Medizinprodukteverordnung** Die Medical Device Regulation (MDR), seit 26. Mai 2021 rechtskräftig, ersetzt die bisherige Medizinprodukterichtlinie (93/42/EWG) und die Richtlinie über aktive implantierbare medizinische Geräte (90/385/EWG). Sie modernisiert und verschärft die Anforderungen an Medizinprodukte, um Sicherheit und Transparenz zu gewährleisten. Mit der MDR erweitert sich der Geltungsbereich auf Produkte ohne medizinische Zweckbestimmung, wie farbige Kontaktlinsen und Implantate für ästhetische Zwecke. Außerdem legt sie erhöhte Anforderungen an klinische Bewertungen, Dokumentation und Sicherheitsüberwachung fest.

**CE-Kennzeichnung** Die CE-Kennzeichnung bleibt ein zentraler Aspekt der MDR und signalisiert die Konformität eines Produkts mit den europäischen Anforderungen. Hersteller müssen dabei erweiterte klinische Nachweise erbringen, insbesondere für Medizinprodukte der Klassen III und implantierbare Produkte. Die MDR führt zudem das System der Unique Device Identification (UDI) ein, um die Rückverfolgbarkeit und Produktsicherheit zu verbessern. Die zugehörige europäische Datenbank (EUDAMED) unterstützt die Transparenz, ist jedoch noch nicht vollständig implementiert.

**Zulassungsverfahren** Im MDR-Zulassungsprozess spielen benannte Stellen eine zentrale Rolle, indem sie die Konformitätsbewertung und Überprüfung klinischer Daten durchführen. Unangekündigte Audits und Produktprüfungen nach dem Inverkehrbringen erhöhen die Sicherheitsstandards. Medizinprodukte nach bisheriger Regulierung (MDD/AIMDD) genießen keinen Bestandsschutz und müssen bis zu den jeweiligen Übergangsfristen erneut zertifiziert werden. Neu ist auch die Pflicht, eine qualifizierte Person in der Organisation zu benennen, die die Einhaltung regulatorischer Anforderungen sicherstellt.

[18]

## 2.4.2 Normen und Qualitätsstandards für Medizinprodukte

**ISO 13485** Die ISO-Norm (Internationale Organisation für Normung) spezifiziert die Anforderungen an ein Qualitätsmanagementsystem für die Entwicklung und Produktion von Medizinprodukten. Die aktuelle Version ist seit 2016 in Kraft.

**FDA 21 CFR Part 820** Qualitätsanforderungen der US-amerikanischen FDA

**IEC 62304** Norm definiert den Lebenszyklus von Software in Medizinprodukten und beschreibt die Anforderungen an das Qualitätsmanagement in Bezug auf Softwareentwicklung und -sicherheit.

## 2.5 Meetings

SCRUM - Erklären

### SCRUM

Scrum ist ein systematischer Ansatz um Projekte strukturiert durchzuführen. Es soll die Teams bei der Lösung komplexer Probleme unterstützen indem Rollen, Regeln und Ereignisse definiert werden. Die zugrundeliegenden Prinzipien sind Empirie und Lean Thinking. „Empirie, die Erfahrung selbst und die auf Erfahrung beruhende Erkenntnis. [19]" something something [20]

## Aufgaben

In den folgenden Abschnitten werden die Tätigkeiten während des Praxissemesters im Überblick dargestellt. Aufgrund der Vertraulichkeit und der wirtschaftlichen Sensibilität der zugrunde liegenden Daten ist eine detaillierte Darstellung spezifischer Inhalte nur eingeschränkt möglich.

## 3 Aufgabe: Dokumentationsautomatisierung

In der Softwareentwicklung ist das Testen essenziell, um die Funktionalität und Korrektheit des Codes sicherzustellen. Ein wesentlicher Bestandteil dieses Prozesses ist die Dokumentation der durchgeführten Tests, einschließlich verwendeter Parameter und spezifischer Bedingungen. Diese Dokumentation ist jedoch oft zeitaufwändig.

Um diesen Aufwand in Zukunft zu minimieren, wird ein Tool entwickelt, das während der Testlaufzeit automatisch die benötigte Dokumentation erstellt und diese direkt in das entsprechende Workitem auf Polarion hochlädt. Das Ziel des Tools ist es, den Aufwand für die Testenden erheblich zu reduzieren. Lediglich einige Strings und Befehle müssen eingefügt werden, damit die Dokumentation generiert und hochgeladen werden kann.

Da Polarion die Einträge im HTML-Format speichert, wird die Dokumentation ebenfalls als HTML hochgeladen. Der fertige HTML-Code wird mithilfe eines Jinja-Templates generiert. Jinja ist eine Template-Engine, die durch eine spezielle Platzhaltersyntax ermöglicht, an definierten Stellen Daten einzufügen.

Im folgenden wird das Beispiel aus Abbildung 3 näher erläutert. In Abbildung 3a ist ein einfaches Beispiel für ein template gezeigt. Die Jinja Platzhalter sind gut zu erkennen durch die doppelten geschweiften Klammern um die Variablen, in dem Beispiel unter anderem in Zeile 8 zu sehen, wo hinter dem String "Hallo" ein beliebiger Name eingefügt werden kann. In der selben Abbildung Zeile 13 bis 15 zeigt das man eine Auflistung mit verschiedenen Inhalten füllen kann. Jinja kann if-Statements und for-Schleifen verwenden um zum Beispiel in das Templater übergebene Listen anzuzeigen. Die Syntax dafür sind `{% for.... %}` und `{% if.... %}` in Kombination mit klassischer Python for/if-Syntax.

Abbildung 3b zeigt den Python Code der die Daten in die Template-Datei gibt und rendert. Zeile 6 wird das Environment mit dem aktuellen Verzeichnis geladen, in Zeile 8 wird daraus dann die Template-Datei geladen, Zeile 11 bis 15 sind ein dictionary wo die Keys den Variablen im Template entsprechen. In Zeile 18 werden die Daten in das Template übergeben und gerendert, anschließend wird die gerenderte Datei erst in der Konsole ausgegeben und dann in einer separaten HTML-Datei gespeichert. Abbildung 3c zeigt dann diese fertige Datei. [21]

```

1 <!DOCTYPE html>
2 <html lang="de">
3 <head>
4   <meta charset="UTF-8">
5   <title>{{ title }}</title>
6 </head>
7 <body>
8   <h1>Hallo, {{ name }}!</h1>
9   <p>Willkommen auf meiner Webseite.</p>
10
11   <h2>Deine Lieblingsdinge:</h2>
12   <ul>
13     {% for item in favorite_things %}
14     <li>{{ item }}</li>
15     {% endfor %}
16   </ul>
17 </body>
18 </html>

```

**Hallo, {{ name }}!**

Willkommen auf meiner Webseite.

**Deine Lieblingsdinge:**

```

{% for item in favorite_things %}
• {{ item }}
{% endfor %}

```

```

1 from jinja2 import Environment, FileSystemLoader
2 import os
3
4 # Jinja-Umgebung einrichten
5 current_dir = os.getcwd() + "/jinja_stuff"
6 env = Environment(loader=FileSystemLoader(current_dir))
7 # Template laden
8 template = env.get_template("beispielbeispiel.html")
9
10 # Daten für das Template
11 data = {
12     "title": "Meine Webseite",
13     "name": "Max",
14     "favorite_things": ["Pizza", "Programmieren", "Reisen"]
15 }
16
17 # Template rendern
18 rendered_html = template.render(data)
19 print(rendered_html)
20
21 # Rendered HTML in eine Datei schreiben
22 with open("template_output.html", "w", encoding="utf-8") as file:
23     file.write(rendered_html)
24

```

(a) Die Jinja Template File

(b) Handling des Templates in Python

```

1 <!DOCTYPE html>
2 <html lang="de">
3 <head>
4   <meta charset="UTF-8">
5   <title>Meine Webseite</title>
6 </head>
7 <body>
8   <h1>Hallo, Max!</h1>
9   <p>Willkommen auf meiner Webseite.</p>
10
11   <h2>Deine Lieblingsdinge:</h2>
12   <ul>
13     <li>Pizza</li>
14     <li>Programmieren</li>
15     <li>Reisen</li>
16   </ul>
17 </body>
18 </html>

```

**Hallo, Max!**

Willkommen auf meiner Webseite.

**Deine Lieblingsdinge:**

- Pizza
- Programmieren
- Reisen

(c) Fertige HTML File nach dem rendern

Abbildung 3: Simple Beispiel für die Anwendung von Jinja templates

### 3.1 Polarion

Polarion (Siemens Company) ist eine webbasierte Anforderungsmanagement-Software, die Unternehmen dabei unterstützt, den gesamten Lebenszyklus von Projekten und Produkten effizient zu verwalten. Als integrierte Lösung bietet Polarion eine Plattform, auf der Anforderungen, Tests, Fehler und Änderungsanfragen nahtlos miteinander verknüpft werden können.

Die Software basiert auf drei Kernprinzipien: Zusammenarbeit (Collaboration), Rückverfolgbarkeit (Traceability) und Workflow-Management. Diese Prinzipien ermöglichen es Teams, in Echtzeit zu kommunizieren und Projekte durchgängig transparent und nachvollziehbar zu gestalten.

Polarion zeichnet sich durch folgende Funktionen aus:

**Webbasierter Zugang:** Polarion ist vollständig browserbasiert, was den Zugriff auf Projektdaten von überall ermöglicht.

**Rückverfolgbarkeit und Auditierung:** Jede Änderung an Anforderungen oder anderen Work Items wird automatisch versioniert und dokumentiert, was die Einhaltung von Compliance- und Auditvorgaben erleichtert.

**Workflow-Steuerung:** Workflows können individuell definiert werden, um sicherzustellen, dass Anforderungen und andere Objekte einen strukturierten Lebenszyklus durchlaufen.

**Kollaborationswerkzeuge:** Funktionen wie Wikis, Benachrichtigungen und Diskussionsforen fördern die Zusammenarbeit zwischen verschiedenen Stakeholdern, von Analysten über Entwickler bis hin zu QA-Teams.

Ein besonderes Merkmal von Polarion ist die Unterstützung verschiedener Entwicklungsansätze wie agiler Methoden, Wasserfall-Modellen oder hybriden Ansätzen. Teams können ihre bevorzugte Methodik nutzen und gleichzeitig von automatisierter Rückverfolgbarkeit, Versionierung und einem flexiblen Berichtswesen profitieren.

Darüber hinaus bietet Polarion die Möglichkeit, Anforderungen und Testfälle parallel zu verwalten. Mit der LiveDocs-Funktion können Teams gemeinsam an Spezifikationen arbeiten, wobei jede Änderung sofort nachverfolgbar ist. Dies erhöht die Transparenz und ermöglicht es, Anforderungen und Tests kontinuierlich abzustimmen.

Zusammenfassend unterstützt Polarion Unternehmen dabei, Projekte effizient zu planen, durchzuführen und zu dokumentieren, während gleichzeitig regulatorische Vorgaben eingehalten werden. [22]

## 3.2 Software Qualitätssicherung

Die Qualitätssicherung von Software ist ein wesentlicher Bestandteil moderner Softwareentwicklung. Ein zentrales Element dabei ist das Testen, das sowohl die Funktionalität aus Nutzersicht als auch die technische Korrektheit des Codes absichert. Code-Tests, wie Unit-, Integrations- und End-to-End-Tests, prüfen einzelne Programmkomponenten bis hin zum Gesamtsystemverhalten. Methoden wie Test-Driven Development (TDD) fördern strukturiertes Arbeiten, indem Testfälle vor der Codeerstellung definiert werden. Diese Vorgehensweise steigert nicht nur die Fehlererkennung, sondern sorgt auch für robusten, wartbaren Code und trägt nachhaltig zur Qualitätssicherung bei. [23]

### Verifizierung

Code == Lastenheft / Funktionsfähig

### Validierung

Code == Nutzungsziele erfüllt?

### 3.3 Regular Expression

Die re-Bibliothek in Python ermöglicht die Anwendung regulärer Ausdrücke (regular expression - regex) zur flexiblen und effizienten Textverarbeitung. Regex sind Muster, die gezielt nach Zeichenfolgen in Textdaten suchen und so vielfältige Datenoperationen ermöglichen. Mit der re-library können Funktionen wie `search`, `match`, `findall` und `sub` genutzt werden, um beispielsweise Texte zu durchsuchen, Muster zu ersetzen und Daten zu validieren. [24]

Die wichtigsten Funktionen der Bibliothek sind:

- `search()` – sucht das erste Vorkommen eines Musters im Text. Wenn das Muster existiert gibt die Funktion ein Match Objekt zurück
- `findall()` – gibt alle Vorkommen des Musters als Liste zurück.
- `sub()` – ersetzt alle Vorkommen des Musters durch einen angegebenen String.
- `match()` – prüft, ob der Text mit dem Muster beginnt.
- `Match.group(idx)` - gibt eine oder mehrere Subgruppen des Matches zurück

Die re-Syntax bietet eine Vielzahl von Operatoren zum definieren der Muster:

.	steht für ein beliebiges Zeichen (außer Zeilenumbrüche)
\	danach kommt ein Charakter mit vordefinierter Funktion zB \d matched alle Unicode decimal Ziffern
* +	geben null oder mehr (*) bzw. mindestens eine (+) Wiederholungen des vorherigen Elements an.
[ ] ( )	definiert Zeichenmengen ( <i>sets</i> ), zB [abc] für die Zeichen a, b oder c.

Listing 1: Beispiel für regex Syntax:

```

1 import re
2
3 txt = "Das ist ein string mit 123 Zahlen"
4 pattern = r"\d+" # alle Ziffern
5
6 # in txt wird das pattern mit "00" ersetzt
7 new_txt = re.sub(pattern, "00", txt)
8 # new_txt: Das ist ein string mit 00 Zahlen
9
10 my_match = re.search(pattern, txt)
11 if my_match:
12     my_group = my_match.group(0)
13     # my_group: 123

```

## 4 Aufgabe: LeRobot

LeRobot ist eine Plattform für maschinelles Lernen im Bereich der realen Robotik, die Modelle, Datensätze und Tools in PyTorch bereitstellt. Ziel der Plattform ist es, den Zugang zur Robotik zu erleichtern, indem Datensätze und vortrainierte Modelle geteilt werden. Der Fokus liegt auf modernen Lernansätzen wie Imitations- und Reinforcement Learning, die speziell für die Anwendung in der realen Welt entwickelt wurden. Die Plattform bietet eine Sammlung von vortrainierten Modellen, Demonstrationsdatensätzen sowie simulierten Umgebungen und hat das Ziel, die Unterstützung für eine Vielzahl von erschwinglichen und leistungsfähigen Robotern kontinuierlich auszubauen.

LeRobot unterscheidet sich durch seine Integration fortschrittlicher Techniken des maschinellen Lernens, die speziell auf die Herausforderungen der Robotik ausgerichtet sind.

Dies umfasst sowohl Imitationslernen (Kapitel 4.2) als auch Reinforcement Learning (Kapitel 4.3), die auf der Plattform verfügbar sind und für eine Vielzahl von Aufgaben im Bereich der Robotik eingesetzt werden können. Die Nutzung dieser Technologien ermöglicht es, Roboter in komplexen, dynamischen Umgebungen zu trainieren, ohne dass eine manuelle Programmierung jeder Aufgabe erforderlich ist.

Durch die Verwendung von Hydra für die Konfiguration von Trainingsszenarien wird die Plattform flexibel und anpassbar. Nutzer können verschiedene Umgebungen und Lernalgorithmen auf einfache Weise kombinieren und testen. Weitere Informationen über den Umgang mit der Plattform sind in dem Kapitel Using LeRobot (4.4) zu finden.

Die Plattform stellt eine benutzerfreundliche API zur Verfügung, mit der Entwickler benutzerdefinierte Trainingsumgebungen erstellen und konfigurieren können. Diese Flexibilität ist besonders wertvoll für Anwendungen in Bereichen wie Manipulation und Greifen, die häufig in der Robotik vorkommen.

LeRobot fördert die Zusammenarbeit und Transparenz, indem es vortrainierte Modelle und Datensätze öffentlich zugänglich macht, was die schnelle Entwicklung und Implementierung von Lösungen ermöglicht. Durch die Bereitstellung dieser Ressourcen wird die Forschung in der Robotikgemeinschaft unterstützt und der Austausch von Best Practices erleichtert.

Zusätzlich ermöglicht LeRobot die Wiederaufnahme von Trainingsläufen, was für langwierige Trainingsprozesse, wie sie im Reinforcement Learning häufig vorkommen, von besonderem Vorteil ist. Dies erlaubt es den Nutzern, Trainingsunterbrechungen zu überwinden und das Training ohne Datenverlust fortzusetzen.

Die kontinuierliche Erweiterung der unterstützten Roboterplattformen und die Integration leistungsfähiger Lernmethoden machen LeRobot zu einer wertvollen Ressource für die Robotikforschung und -entwicklung. [25].

## 4.1 Machine Learning

Machine Learning (ML) bezeichnet den Prozess, bei dem Maschinen eigenständig Muster in Daten erkennen und daraus lernen, um zukünftige Entscheidungen zu optimieren. ML-Modelle sind iterative Prozesse, die Eingaben analysieren, durch Rückmeldungen optimieren und schließlich eigenständig Lösungen für komplexe Probleme entwickeln. Dieser Ansatz ist die Grundlage für fortschrittliche Technologien wie Imitation- und Reinforcement Learning. [23]



## 4.2 Imitation learning

Imitation Learning stammt aus dem Bereich des maschinellen Lernens und ermöglicht es Robotern, neue Fähigkeiten durch das Beobachten und Nachahmen von Verhaltensweisen zu erlernen. Dieser Ansatz basiert auf Konzepten der Psychologie und Biologie, die es Robotern erlauben, nicht nur eigenständig durch *trial and error* oder durch detaillierte Programmierung zu lernen, sondern auch von den Erfahrungen anderer, sei es von Menschen oder anderen Robotern, zu profitieren. Das Nachahmen von Bewegungsabläufen ermöglicht es Robotern, gewünschte Verhaltensweisen schneller und effizienter zu erlernen. Dies bietet eine Reihe von Vorteilen: Roboter können sich schneller an neue Aufgaben oder Umgebungen anpassen, indem sie bestehende Verhaltensweisen direkt übernehmen. Gleichzeitig entfallen aufwändiges Programmieren und langwierige Trainingsphasen. Darüber hinaus ermöglicht Imitation Learning den Wissenstransfer zwischen Robotern, unabhängig von ihrer Hardware oder eingesetzten Programmiersprachen. [26]

## 4.3 Reinforcement Learning

Reinforcement Learning (RL) ist ein maschinelles Lernverfahren, bei dem ein Agent durch Interaktion mit einer Umgebung lernt, optimale Entscheidungen zu treffen. Der Agent befindet sich in einem Zustand und wählt basierend auf einer Policy (Strategie) eine Aktion, die den nächsten Zustand beeinflusst und eine Belohnung generiert. Im Gegensatz zu herkömmlichen Methoden erfordert RL keine explizite Programmierung der Verhaltensweisen. Es wird häufig eingesetzt, um motorische Fähigkeiten zu erlernen. Ein wesentlicher Vorteil von RL besteht darin, dass ein Roboter nicht nur spezifische Aufgaben bewältigen kann, sondern auch in der Lage ist, sich durch die Kombination mehrerer Bewegungsmuster an neue Situationen anzupassen. Dabei erfolgt das Lernen weitgehend autonom, ohne ständiges menschliches Eingreifen. [27]

## 4.4 Das Nutzen von LeRobot

LeRobot bietet eine benutzerfreundliche Schnittstelle zur Nutzung von vortrainierten Modellen, Datensätzen und Simulationsumgebungen für Robotikanwendungen. Die Nutzung der Plattform erfolgt hauptsächlich über die API, die in Python implementiert ist und auf der PyTorch-Bibliothek basiert. Benutzer können LeRobot sowohl für die Entwicklung von maschinellen Lernmodellen als auch für das Training von Robotern in simulierten Umgebungen einsetzen. Im Folgenden wird ein Überblick über die grundlegende Nutzung der Plattform und ihre Hauptfunktionen gegeben.

### Installation und Einrichtung

Um LeRobot zu nutzen, müssen zunächst die erforderlichen Bibliotheken und Abhängigkeiten installiert werden. LeRobot basiert auf PyTorch und erfordert eine Umgebung, die diese Bibliothek unterstützt. Eine detaillierte Anleitung zur Installation findet sich in der [offiziellen Dokumentation](#). Nach der Installation können die verschiedenen Module und Werkzeuge geladen und konfiguriert werden.

### **Roboterarm SO-ARM100 von TheRobotStudio**

Der SO-ARM100 ist ein kostengünstiges Roboterarm-Modell entwickelt von TheRobotStudio, entwickelt von TheRobotStudio, einem Unternehmen, das Roboter inspiriert von biologischen Bewegungsapparaten entwickelt und herstellt [28]. Das zugehörige [GitHub-Repository](#) enthält Designs und Anleitungen für den Bau von zwei Roboterarmen, wobei für viele Anwendungen bereits ein einzelner Arm ausreichend ist. Ziel des SO-ARM100 ist es, einen preiswerten und dennoch funktionalen Roboterarm zu ermöglichen. Die Gesamtkosten für die Bauteile belaufen sich auf etwa 150 € pro Arm, einschließlich einer Ein-Kilogramm-Rolle PLA-Filament (ca. 20 €) für den 3D-Druck der Komponenten. [29]

### **Daten und Modelle**

LeRobot stellt eine Vielzahl von [Datensätzen und vortrainierten Modellen](#) zur Verfügung, die in unterschiedlichen Szenarien der Robotik eingesetzt werden können. Die Datensätze umfassen verschiedene Arten von Simulations- und Echtwelt-Daten, die für das Training und die Evaluierung von Modellen in Aufgaben wie Manipulation, Greifen und Navigation verwendet werden können. Vortrainierte Modelle sind in der Regel auf häufige Robotikanwendungen optimiert und ermöglichen eine schnelle Implementierung und Evaluierung von Ansätzen für Imitations- und Reinforcement Learning.

### **Trainingsumgebungen und Konfigurationen**

LeRobot bietet eine Vielzahl von konfigurierbaren Trainingsumgebungen, die über die Hydra-Konfigurationsbibliothek verwaltet werden. Mit Hydra können Benutzer ihre Trainingsszenarien durch einfache Änderungen an den Konfigurationsdateien anpassen, ohne den Code neu schreiben zu müssen. Es ist möglich, sowohl simulierte als auch reale Umgebungen zu definieren, indem die entsprechenden Konfigurationsparameter angepasst werden. Weitere Informationen zur Konfiguration von Trainingsumgebungen und zur Nutzung von Hydra finden sich in [Repository Beispiel 4](#).

### **Training und Modell-Resumption**

Das Training von Modellen in LeRobot erfolgt in einer standardisierten Schleife, die Vorwärts- und Rückwärtsdurchläufe sowie Optimierungsschritte umfasst. Benutzer können die Trainingsprozesse über die Kommandozeile starten und steuern, wobei die Konfigurationen dynamisch angepasst werden können. Sollte das Training unterbrochen werden, bietet LeRobot eine Funktion zum Fortsetzen des Trainings, sodass der Trainingsprozess nahtlos fortgeführt werden kann. Weitere Details zur Wiederaufnahme von Trai-

ningsläufen sind in [Repository Beispiel 5](#) beschrieben.

### **Evaluierung und Ergebnisse**

LeRobot unterstützt auch die Evaluierung von trainierten Modellen in simulierten Umgebungen. Die Evaluierung erfolgt über festgelegte Metriken, die die Leistung eines Modells bei der Ausführung einer bestimmten Aufgabe messen. Zu den wichtigsten Metriken gehören der kumulierte Reward, die Erfolgsrate und die Reaktionszeit des Modells. Benutzer können die Evaluierungsparameter über die Konfigurationsdateien anpassen, um detaillierte Leistungsanalysen zu erhalten.

### **Erweiterbarkeit und Anpassung**

Ein herausragendes Merkmal von LeRobot ist seine Flexibilität und Erweiterbarkeit. Benutzer können benutzerdefinierte Datensätze und Modelle integrieren, eigene Trainingsumgebungen erstellen oder die vorgefertigten Umgebungen und Modelle für neue Aufgaben anpassen. Dies ermöglicht es Entwicklern und Forschern, die Plattform an spezifische Anforderungen anzupassen und schnell neue Ideen zu testen.

### **Zusammenarbeit und Community**

LeRobot fördert die Zusammenarbeit durch die Bereitstellung einer offenen Plattform, die es Entwicklern und Forschern ermöglicht, Modelle, Datensätze und Trainingsszenarien zu teilen. Diese Offenheit trägt dazu bei, die Entwicklung von Lösungen in der Robotik zu beschleunigen und Best Practices auszutauschen. Weitere Informationen zur Community und den verfügbaren Ressourcen sind auf der Website von LeRobot zugänglich.

Zusammenfassend lässt sich sagen, dass LeRobot eine leistungsstarke und flexible Plattform für maschinelles Lernen in der Robotik ist, die eine breite Palette von Tools und Ressourcen bietet, um die Entwicklung und das Training von Robotern zu erleichtern. Die umfangreiche Dokumentation und die benutzerfreundliche API machen den Einstieg in die Robotik und das maschinelle Lernen für neue Benutzer einfach und effizient.

## 5 Ergebnisse

Präsentiere und diskutiere hier die Ergebnisse deines Berichts.

## 6 Fazit

Im Fazit fasst du alles zusammen und gibst einen Ausblick.

## Literatur

- [1] Löwenstein Medical. Zugriff am: 05.11.2024.
- [2] David Gozal and Leila Kheirandish-Gozal. Cardiovascular morbidity in obstructive sleep apnea: oxidative stress, inflammation, and much more. *American Journal of Respiratory and Critical Care Medicine*, 177(4):369–75, 2007.
- [3] Karsten Müssig. Gesunder Schlaf - gesunder Stoffwechsel. *Info Diabetologie*, 18:12–13, Dezember 2024.
- [4] Dong-Fang Zhao, Yi-Zhu Zhang, Xue Sun and Chun-Yan Su, and Li-Qiang Zhang. Association between obstructive sleep apnea severity and depression risk: a systematic review and dose-response meta-analysis. *Sleep and Breathing*, 28:2175–2185, June 2024.
- [5] Zlatan Križan. *Schlaf, Persönlichkeit und soziales Verhalten*. Springer Cham, 1 edition, 2024.
- [6] Fabienne Reh. EEG Brain Wave Frequencies. <https://www.doccheck.com/de/detail/photos/42653-eeg-baender>. Zugriff am: 03.12.2024.
- [7] Schlafspindeln und K-Komplex-Gehirnströme. <https://www.schlaf.org/schlafspindeln>. Zugriff am: 03.12.2024.
- [8] DocCheck Flexikon. Zugriff am: 27.11.2024.
- [9] Willibald Pschyrembel und Co. *Pschyrembel Klinisches Wörterbuch*. De Gruyter, 267 edition, 2017.
- [10] Aakash K. Patel, Vamsi Reddy, Karlie R. Shumway, and John F. Araujo. *Physiology, Sleep Stages*. StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing, Januar 2024. PMID: 30252388, Bookshelf ID: NBK526132.
- [11] C.Heiser and D. Eckert. Pathophysiologie der obstruktiven Schlafapnoe. *HNO*, 67:654–662, August 2019.
- [12] Deutsche Gesellschaft für Schlafforschung und Schlafmedizin. *S3-Leitlinie Nicht erholsamer Schlaf/Schlafstörung: Kapitel „Schlafbezogene Atmungsstörungen bei Erwachsenen“*. AWMF, 2.0 edition, August 2017. Zugriff am: 25.11.2024.
- [13] Thomas Verse and Karl Hörmann. Operative Therapie der Obstruktion bei schlafbezogenen Atmungsstörungen. *Deutsches Ärzteblatt*, 108(13):216–221, 2011.

- [14] Dr Shiro Fujita MD, Dr William Conway MD, Dr Frank Zorick MD, and Dr Thomas Roth PhD. Surgical Correction of Anatomic Abnormalities in Obstructive Sleep Apnea Syndrome: Uvulopalatopharyngoplasty. *Otolaryngology–Head and Neck Surgery*, 89:889–1054, November 1981.
- [15] Löwenstein Medical Technology. *Gebrauchsanweisung prisma SMART*.
- [16] Löwenstein Medical. prisma SMART. [https://hul.de/at/produkt/prisma-smart-3/?ref=at.hul.de&ref\\_lang=at](https://hul.de/at/produkt/prisma-smart-3/?ref=at.hul.de&ref_lang=at). Zugriff am: 03.12.2024.
- [17] Bundesministerium für Gesundheit. Website Bundesministerium für Gesundheit. Zugriff am: 27.11.2024.
- [18] TÜV Süd. EU-Medizinprodukteverordnung (MDR). Zugriff am 23.12.2024.
- [19] Hogrefe. Dorsch - Lexikon der Psychologie, 2024. Zugriff am: 30.10.2024.
- [20] Ken Schwaber and Jeff Sutherland. The Scrum Guide, 2020. Zugriff am: 29. Oktober 2024.
- [21] Pallets Projects. Jinja Documentation. Zugriff am: 19.12.2024.
- [22] Siemens Industry Software GmbH. Zugriff am: 03.12.2024.
- [23] Philipp Winniewski. *Grundlagenwissen der Software-Entwicklung - IT-Konzepte und Fachbegriffe für das Projektmanagement*. Springer Vieweg Wiesbaden, 1 edition, Januar 2024.
- [24] Python Software Foundation. Zugriff am: 03.12.2024.
- [25] Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, and Thomas Wolf. LeRobot: State-of-the-art Machine Learning for Real-World Robotics in Pytorch. <https://github.com/huggingface/lerobot>, 2024.
- [26] Paul Bakker and Yasuo Kuniyoshi. Robot See, Robot Do : An Overview of Robot Imitation. *AISB96 Workshop on Learning in Robots and Animals*, 05 1996.
- [27] Jens Kober and Jan Peters. *Learning Motor Skills - From Algorithms to Robot Experiments*. Springer Cham, 1 edition, November 2013.
- [28] TheRobotStudio Website. Zugriff am 23.12.2024.
- [29] TheRobotStudio. SO-ARM100. <https://github.com/TheRobotStudio/SO-ARM100?tab=readme-ov-file>.