



Matthias Prim, BSc

# **Re-Design einer Ansteuerungssoftware eines Lokomotionsgerätes für Kleinkinder**

**Masterarbeit**

zur Erlangung des akademischen Grades

Diplom-Ingenieur

eingereicht an der

**Technischen Universität Graz**

Betreuer

Univ.-Prof. Dipl.-Ing. Dr.techn. Christian Baumgartner

Assoc.Prof. Dipl.-Ing. Dr.techn. Jörg Schröttner

Institut für Health Care Engineering

Graz, März 2019

## EIDESSTÄTLICHE ERKLÄRUNG

### **AFFIDAVIT**

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit/Diplomarbeit/Dissertation identisch.

*I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis/diploma thesis/doctoral dissertation.*

---

Datum / Date

---

Unterschrift / Signature

*Die Technische Universität Graz übernimmt mit der Betreuung und Bewertung einer Masterarbeit keine Haftung für die erarbeiteten Ergebnisse: Eine positive Bewertung und Anerkennung (Approbation) einer Arbeit bescheinigt nicht notwendigerweise die vollständige Richtigkeit der Ergebnisse.*

## **Danksagung**

Ich danke meinen Eltern, die mir auf meinem Bildungsweg immer beratend zur Seite standen und mir mein Studium auch in finanzieller Hinsicht erleichtert haben.

Besonderer Dank geht auch an Wolfgang, der mir in fachspezifischen Fragen bezüglich der Umsetzung immer mit Rat und Tat zur Seite stand.

Ich danke meiner Freundin Anna für die zahlreichen Ermutigungen während meiner Studienzeit. Du hast mich durch alle Höhen und Tiefen begleitet.

Abschließend danke ich noch all meinen Freunden und Studienkollegen für die unvergessliche Zeit und die zahlreichen schönen Erinnerungen.

# **Re-Design einer Ansteuerungssoftware eines Lokomotionsgerätes für Kleinkinder**

Kinder mit infantiler Zerebralparese leiden an einer unvollständigen Lähmung, die durch eine zerebrale Läsion ausgelöst wird. In der Folge treten Störungen der Mobilität auf, die zu pathologischen Veränderungen der Gelenke, Knochen und Skelettmuskulatur führen. Das Lokomotionstraining soll die physiologische Bewegung von Kindern fördern und das Einprägen von pathologischen Gangmustern verhindern.

Die vorliegende Arbeit umfasst die Funktionserweiterung sowie insbesondere die Überarbeitung der Usability der Software für einen Lokomotionsgeräte-Prototypen für Kleinkinder. Das passive Laufband wird durch ein automatisiertes ersetzt und das Geschwindigkeitsprofil wird auf die Bewegung der Orthesen angepasst. Im Zuge dessen kommt es auch zu einer Ergänzung von Kraftsensoren zur Bestimmung der Gewichtsbelastung. Eine visuelle Alarmierung bei aktivem Not-Aus Schalter wird in der Software implementiert. Die Entwicklung und Aktualisierung der Dokumentation der Software wird auf Basis der Norm EN 62304 durchgeführt. Zusätzlich wird das vorhandene Risikomanagement aktualisiert, um mögliche neu entstehende Risiken zu detektieren und zugehörige Abhilfemaßnahmen zu erarbeiten. Alle vorhergesehenen Usability Maßnahmen konnten erfolgreich implementiert werden.

Schlüsselwörter: Lokomotionstherapie, Software, Hardware, Risikomanagement, EN 62304

## **Re-Design of control software of a locomotion therapy device for infants**

Children with infantile cerebral palsy suffer from a partial paralysis caused by cerebral lesions. This results in movement disorders which lead to pathologic changes of joints, bones and skeletal muscles. Locomotion training is an often-used therapy form which supports the physiological movement and suppresses pathologic gait patterns.

The aim of the present work has been to expand the functionality and improve the usability of the software of a locomotion therapy device for infants. The passive treadmill was substituted by an automatic one and the related velocity profile was adapted to the movement of the orthosis. As part of the replacement force sensors have been added to calculate the weight of the patient. A visual indication in case of a pressed emergency button has been implemented. The development and update of the documentation have been done according to the standard EN 62304. Further risk management has been extended, possible new arising risks have been identified and appropriate safety measures have been deduced. All designated usability measures have been successfully implemented.

Keywords: Locomotion therapy, software, hardware, risk management, EN 62304

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis .....</b>	<b>VIII</b>
<b>Tabellenverzeichnis .....</b>	<b>X</b>
<b>Abkürzungsverzeichnis .....</b>	<b>XI</b>
<b>Formelzeichen .....</b>	<b>XII</b>
<b>1 Einleitung.....</b>	<b>13</b>
1.1 Der menschliche Gang.....	13
1.2 Infantile Cerebralparese .....	17
1.2.1 Definition und Häufigkeit .....	17
1.2.2 Ätiologie .....	17
1.2.3 Klassifikation.....	18
1.2.4 Symptome .....	18
1.2.5 Therapie .....	19
1.3 Lokomotionstherapie .....	19
1.4 Normative Rahmenbedingungen .....	21
<b>2 Aufgabenstellung .....</b>	<b>23</b>
<b>3 Methoden .....</b>	<b>24</b>
3.1 Risikomanagement .....	24
3.1.1 Risikoanalyse .....	25
3.1.2 Risikobewertung .....	26
3.1.3 Risikobeherrschung.....	27
3.1.4 Bewertung der Akzeptanz des Gesamt-Restrisikos .....	28
3.1.5 Risikomanagementbericht .....	28
3.1.6 Informationen aus Herstellung und nachgelagerten Phasen .....	28
3.2 Software-Sicherheitsklassifizierung.....	29
3.3 Software-Entwicklungs-Prozess .....	31
3.3.1 Planung der Software-Entwicklung.....	31
3.3.2 Analyse der Software-Anforderungen .....	32
3.3.3 Design der Software-Architektur .....	33
3.3.4 Detailliertes Software-Design.....	34
3.3.5 Implementieren–Verifizieren-Integrieren.....	36
3.3.6 Prüfung des Software-Systems.....	39
3.3.7 Software-Freigabe .....	39
3.4 Problemlösungs-Prozess für Software.....	40
3.5 Software-Konfigurationsmanagement-Prozess .....	42

3.6	Ausgangssituation der Arbeit.....	46
3.7	LabView.....	49
3.8	Portierung der Lokomotionsgeräte-Software.....	50
3.8.1	USB-to-CAN Konverter .....	50
3.8.2	Verifizieren der Verbindung .....	50
3.8.3	Initialisierung der USB-CAN-Kommunikation .....	52
3.8.4	Spitzen- und Dauerstrombegrenzung .....	53
3.8.5	Korrektur der Referenzierungs-Offsets.....	54
3.9	Hardware .....	56
3.9.1	NI USB-6001.....	56
3.9.2	Gewichtsbestimmung .....	57
3.9.3	Detektion des Not-Aus-Schalters.....	63
3.10	Aufmerksamkeitsüberwachung .....	65
3.11	Usability und neue Funktionen .....	67
3.11.1	Anwenderverwaltung .....	68
3.11.2	Patientenverwaltung und Patientenakte .....	69
3.11.3	Therapiemodus.....	71
3.11.4	Fehlerbehandlung.....	74
3.12	Realisierung eines automatisierten Laufbands .....	75
3.12.1	Berechnung der Gehgeschwindigkeit .....	75
3.12.2	Auswahl des Antriebs.....	79
3.12.3	Schaltung und Steuerung mit USB-DAQ-Device .....	84
<b>4</b>	<b>Ergebnisse.....</b>	<b>90</b>
4.1	Software-Sicherheitsklassifizierung.....	90
4.1.1	Risiko des Laufbands.....	91
4.1.2	Risiken in Verbindung mit der Gewichtsmessung .....	91
4.1.3	Risikobewertung zur Software-Sicherheitsklassifizierung .....	92
4.1.4	Ergebnis der Software-Sicherheitsklassifizierung.....	92
4.2	Software-Risikomanagement.....	93
4.2.1	Gesamtbewertung .....	95
4.3	Software-Anforderungen.....	95
4.4	Software-Architektur .....	96
4.4.1	Ablauf der Lokomotionstherapie-Software Version 0.3 .....	97
4.5	Software-Detail-Design.....	100
4.6	Implementieren-Verifizieren-Integrieren .....	101

4.7	Verifizierung des Software-Systems.....	102
4.7.1	Verifizierung der Bewegung der Orthesen.....	102
4.7.2	Verifizierung der Geschwindigkeit des Laufbands.....	106
4.8	Problembereichte und Änderungsanträge .....	107
4.9	Software-Freigabe .....	108
<b>5</b>	<b>Diskussion .....</b>	<b>109</b>
5.1	Portierung und Debugging.....	109
5.2	Dokumentation und Verifizierung.....	109
5.3	Versionsverwaltung und Backup.....	110
5.4	Gewichtsmessung.....	110
5.5	Automatisiertes Laufband.....	110
5.6	Risikomanagement .....	111
5.7	Verbesserungspotential.....	111
<b>6</b>	<b>Schlussfolgerung .....</b>	<b>112</b>
<b>7</b>	<b>Literatur .....</b>	<b>113</b>
<b>8</b>	<b>Anhang.....</b>	<b>117</b>
A.	Weitere Abbildungen .....	117
B.	Berechnung der Rollerdrive Fördergeschwindigkeit.....	120
B.1	Minimale Fördergeschwindigkeit .....	120
B.2	Maximale Fördergeschwindigkeit .....	121
B.3	Fördergeschwindigkeit in Abhängigkeit der Spannung .....	122
C.	Usability Änderungen .....	124
D.	Durchgeführte Änderungen an der Lokomotionssoftware .....	126
E.	Vorschläge für Lokomotionstherapie V-0.4.....	139

*Aufgrund der besseren Lesbarkeit wurde beim Verfassen dieser Diplomarbeit auf eine geschlechterkonforme Schreibweise verzichtet. Sämtliche Personenbezeichnungen sind als geschlechtsneutral anzusehen.*

# Abbildungsverzeichnis

Abbildung 1: Schrittlänge (step length) und Doppelschrittlänge (stride length) [3].....	14
Abbildung 2: Der menschliche Gangzyklus vgl. [1] .....	15
Abbildung 3: Winkelverläufe in Abhängigkeit des Gangzyklus vgl. [4].....	16
Abbildung 4: Verschiedene Systeme von Lokomotionstherapie-Geräten.....	20
Abbildung 5: Ablaufdiagramm Risikomanagement-Prozess vgl. [14] .....	24
Abbildung 6: Fehlerbaumanalyse (FBA / FTA) und Fehlermöglichkeits- und Einflussanalyse (FMEA) [15].....	25
Abbildung 7: Risikomatrix mit Risikostufen I bis IV .....	26
Abbildung 8: Software-Sicherheitsklassifizierungs-Prozess vgl. [10].....	29
Abbildung 9: Ableitung der Software-Anforderungen .....	32
Abbildung 10: Ableitung der Software-Architektur.....	33
Abbildung 11: Exemplarisches Detail-Design-Dokument (0.4 Select Printer-V-0).....	35
Abbildung 12: Implementieren-Verifizieren-Integrieren Aktivität.....	36
Abbildung 13: Exemplarisches Testprotokoll Seite 1 (1.3 Zugangsbeschränkung-V-0.2).....	37
Abbildung 14: Exemplarisches Testprotokoll Seite 2 (1.3 Zugangsbeschränkung-V-0.2).....	38
Abbildung 15: Problemlösungs-Prozess für Software .....	40
Abbildung 16: Exemplarischer Problembereicht (2 Patientenverwaltung-V-0.3.vi) .....	41
Abbildung 17: Änderungskontroll-Prozess.....	43
Abbildung 18: Exemplarischer Änderungsantrag Seite 1 (2 Patientenverwaltung-V-0.3.vi)...	44
Abbildung 19: Exemplarischer Änderungsantrag Seite 2 (2 Patientenverwaltung-V-0.3.vi)...	45
Abbildung 20: Re-Design des Lokomotionstherapiegeräts (3D-Entwurf) [18].....	47
Abbildung 21: Steuerungskonzept der Orthesen vgl. [12].....	48
Abbildung 22: Lokomotionstherapie-Software Ausschnitt: Frontpanel rechts, Blockdiagramm links.....	49
Abbildung 23: IXXAT USB-to-CAN V2 compact.....	50
Abbildung 24: Faulhaber Motion Manager (Dr. Fritz Faulhaber GmbH & Co. KG, Deutschland) .....	51
Abbildung 25: Faulhaber Motion Controller MCDC 3006 S CF .....	53
Abbildung 26: Motion Controller Strombegrenzung .....	54
Abbildung 27: NI USB-6001.....	56
Abbildung 28: Wägezelle.....	58
Abbildung 29: Instrumentenverstärker INA 125P mit Vollbrücke [33].....	59
Abbildung 30: 3D Modell Instrumentenverstärker-Board.....	60
Abbildung 31: Verstärkte Brückenspannung in Abhängigkeit des Gewichts.....	62



Abbildung 32: Schütz Schrack 25A, 3S+1Ö, 230VAC und Schaltbild [35] .....	63
Abbildung 33: USB-Fußschalter, f-pro .....	65
Abbildung 34: Neues und altes Design des Hauptmenüs .....	67
Abbildung 35: Anwenderverwaltung .....	68
Abbildung 36: Patientenverwaltung .....	69
Abbildung 37: Patientendaten ändern.....	70
Abbildung 38: Patientenakte.....	71
Abbildung 39: Re-Design des Therapiemodus.....	72
Abbildung 40: Mathscript RT .....	73
Abbildung 41: Fehlermeldung der Lokomotionsgerätesoftware .....	74
Abbildung 42: Maximaler Hüftwinkel Orthesenhalterungen.....	76
Abbildung 43: Berechnung der Step Length (Schrittlänge) .....	76
Abbildung 44: Interroll Rollerdrive EC310 .....	79
Abbildung 45: Rollerdrive EC310 Pinbelegung [38] .....	80
Abbildung 46: Rollerdrive EC310 Drehzahl in Abhängigkeit der Spannung .....	83
Abbildung 47: Rollerdrive EC310 Geschwindigkeit in Abhängigkeit der Spannung.....	83
Abbildung 48: Rollerdrive Steuerung mit USB-DAQ-Device.....	84
Abbildung 49: Kadenzminimum beim Anfahren in Abhängigkeit der Schrittlänge .....	86
Abbildung 50: Rel. Abweichung der Laufbandgeschwindigkeit in Abhängigkeit der Kadenz .....	87
Abbildung 51: Finales Re-Design des Lokomotionsgeräts.....	89
Abbildung 52: Neue relevante Risiken für die Software-Sicherheitsklassifizierung durch das Re-Design .....	91
Abbildung 53: Vereinfachtes Hierarchiediagramm der Lokomotionstherapie- Software [12] .....	96
Abbildung 54: Ablauf der Lokomotionstherapie-Software .....	99
Abbildung 55: Soll- Ist-Vergleich Kniewinkel 20 Schritte/min .....	103
Abbildung 56: Soll- Ist-Vergleich Hüftwinkel 20 Schritte/min .....	103
Abbildung 57: Soll- Ist-Vergleich Kniewinkel 60 Schritte/min .....	104
Abbildung 58: Soll- Ist-Vergleich Hüftwinkel 60 Schritte/min .....	104
Abbildung 59: Soll- Ist-Vergleich Kniewinkel 120 Schritte/min .....	105
Abbildung 60: Soll- Ist-Vergleich Hüftwinkel 120 Schritte/min .....	105
Abbildung 61: Verifizierung der Laufbandgeschwindigkeit .....	106
Abbildung 62: V-Modell des Software-Entwicklungs-Prozesses [12].....	117
Abbildung 63: Konstruktionsplan Instrument Amplifier Board.....	118
Abbildung 64: Hierarchiediagramm der Lokomotionssoftware V 0.3 .....	119

# Tabellenverzeichnis

Tabelle 1: Phasen des Gangzyklus [1] .....	15
Tabelle 2: Beschreibung der qualitativen Schadensfolgen zur Risikobewertung .....	26
Tabelle 3: Risikostufen (I bis IV) und ihre Bedeutung .....	27
Tabelle 4: Kritikalitäts-Stufen des Problemlösungs-Prozesses .....	40
Tabelle 5: Vorgesehener Anwendungsbereich des Lokomotionsgeräts (Re-Design) .....	46
Tabelle 6: Komponenten des Antriebssystems .....	48
Tabelle 7: Referenzierungs-Winkel der Hüfte und Knie .....	55
Tabelle 8: Not-Aus-Schalter Logiktable .....	64
Tabelle 9: Zulässige Beinlängen des Lokomotionsgeräts (Re-Design) .....	75
Tabelle 10: Berechnete Ergebnisse der Gehgeschwindigkeit .....	78
Tabelle 11: Technische Daten Rollerdrive EC310 [38] .....	80
Tabelle 12: Rollerdrive EC310 Motorstecker Pinbeschreibung .....	81
Tabelle 13: Rollerdrive EC310 Drehzahl und Geschwindigkeit .....	82
Tabelle 14: Ursprüngliche Risikomatrix für die Software-Sicherheitsklassifizierung [12] .....	90
Tabelle 15: Aktuelle Risikomatrix für die Software-Sicherheitsklassifizierung .....	90
Tabelle 16: Risikomatrix des Software-Risikomanagements nach Durchführung der Abhilfemaßnahmen .....	94
Tabelle 17: Risiko-Ursachen und angewandte Kontrollmaßnahmen des Re-Designs .....	94
Tabelle 18: Zusätzliche Risikokontrollmaßnahmen des Re-Designs .....	94
Tabelle 19: Übersicht aller Software-Komponenten .....	100
Tabelle 20: Standardabweichung der Knie- und Hüftwinkel in Abhängigkeit der Ganggeschwindigkeit .....	102
Tabelle 21: Restanomalien der Lokomotionstherapie-Software V 0.3 .....	108
Tabelle 22: Usability Defizite und Abhilfemaßnahmen der Lokomotionstherapie- Software Version 0.2 .....	124

## Abkürzungsverzeichnis

AC.....	Wechselstrom (alternating current)
AEA.....	Änderungsantrag
ALARA .....	As low as reasonable achievable
ALARP .....	As low as reasonable practicable
CAN .....	Controller Area Network
DC.....	Gleichstrom (direct current)
DLL .....	dynamische Programmbibliothek (Dynamic Link Library)
FMEA .....	Fehler-Möglichkeiten- und -einflussanalyse (Failure Mode and Effect Analysis)
FTA .....	Fehlerbaumanalyse (Fault Tree Analysis)
GMFCS .....	Gross Motor Function Classification System
GUI.....	Benutzeroberfläche (Graphical User Interface)
ICP .....	Infantile Zerebralparese (infantile cerebral palsy/paresis)
IVI.....	Implementieren-Verifizieren-Integrieren
NC.....	Normalbedingung (normal condition)
PB .....	Problembereich
PCB.....	Leiterplatte, Platine (Printed Circuit Board)
PHA.....	Vorläufige Gefährdungsanalyse (Preliminary Hazard Analysis)
RA .....	Risikoanalyse
SFC.....	Erster Fehlerfall (single fault condition)
SOUP .....	Software unbekannter Herkunft (Software of unknown provenance)
SWA.....	Software-Anforderungen
SysA.....	System-Anforderungen
VI.....	Virtuelles Instrument (virtual instrument)

## Formelzeichen

$A$	Querschnittsfläche des Leiters
$\gamma$	Maximaler Hüftwinkel der Orthesenhalterung
$\varepsilon$	Dehnung
$G$	Verstärkung
$GK$	Gegenkathete
$HYP$	Hypotenuse
$i$	Untersetzungsverhältnis
$k$	Empfindlichkeit
$K$	Kadenz (Schritte pro Zeiteinheit)
$l$	Länge des Leiters
$\Delta l$	Längenänderung des Leiters
$l_{B,max}$	Größte zulässige Beinlänge
$l_{O,max}$	Größte zulässige Oberschenkellänge
$l_{U,max}$	Größte zulässige Unterschenkellänge
$l_{S,max}$	Maximale Schrittlänge (max. step length)
$l_{B,min}$	Kleinste zulässige Beinlänge
$l_{O,min}$	Kleinste zulässige Oberschenkellänge
$l_{U,min}$	Kleinste zulässige Unterschenkellänge
$l_{S,min}$	Minimale Schrittlänge (min. step length)
$n_1$	Innere Drehzahl
$n_2$	Äußere Drehzahl
$r$	Radius
$R$	Ohmscher Widerstand
$\Delta R$	Widerstandsänderung
$R_G$	Widerstandswert für die Verstärkung $G$
$\rho$	spezifischer Widerstand
$\sigma$	Standardabweichung
$s$	Weg / Strecke
$\sin$	Sinus
$t$	Zeit
$U$	Spannung
$U$	Umfang
$\Delta U$	Spannungsdifferenz
$v$	Geschwindigkeit
$v_{ist}$	Ist-Geschwindigkeit
$v_{max}$	Maximale Gehgeschwindigkeit
$v_{min}$	Minimale Gehgeschwindigkeit
$v_{soll}$	Soll-Geschwindigkeit
$\Delta v_{abs}$	Absolute Geschwindigkeits-Abweichung
$\Delta v_{rel}$	Relative Geschwindigkeits-Abweichung

# 1 Einleitung

Am Institut für Health Care Engineering der Technischen Universität Graz wurde in Kooperation mit der Forschungsabteilung der Klinik Judendorf-Straßengel „Auch ich will gehen“, ein Prototyp zur Lokomotionstherapie für Kleinkinder mit Infantiler Cerebralparese entwickelt. Ausgangspunkt für das Re-Design der Software stellte ein überarbeitetes Konzept der ursprünglichen Lokomotionsgeräte-Hardware dar, welche zeitgleich entstand. Nachfolgend werden der physiologische Gang des Menschen, das Krankheitsbild und die Lokomotionstherapie erläutert.

## 1.1 Der menschliche Gang

Kinder erlernen mit 9–12 Monaten die Standstabilität, welche als Voraussetzung der Gehfunktion gegeben sein muss. Fortan beginnen sie mit geführtem und schließlich dann auch mit freiem Gehen. Regelmäßiges Hinfallen und Aufstehen stellt einen wichtigen Trainingsreiz zur stetigen Verbesserung der Haltungsverstärkung und damit zum selbstständigen Gehen dar. Obwohl sich die freie Gehfunktion um das 1. Lebensjahr entwickelt, ist diese erst viel später vollständig ausgereift. Mit 7–8 Jahren stellt sich sodann ein gleichbleibender stabiler Gang ein. [1]

Der menschliche Gang wird wie folgt definiert:

„Walking uses a repetitious sequence of limb motion to move the body forward while simultaneously maintaining stance stability.“

Zitiert aus [2]

Wichtige Parameter zur Klassifikation des Gangs bilden die Schrittlänge (Einzel- und Doppelschritt), die Gehgeschwindigkeit und die Kadenz (Schrittfrequenz). Die Schrittlänge (step length) ist definiert als die Distanz zwischen dem Fersenkontakt des einen und dem Vorfußkontakt des kontralateralen Fußes. Unter einer Doppelschrittlänge (stride length) versteht man den Abstand zwischen zwei Initialkontakten desselben Fußes. Das Zeitintervall eines Doppelschrittes wird als Gangzyklus bezeichnet, da sich der vollendete Bewegungsverlauf kontinuierlich periodisch wiederholt. [1] [2]

Abbildung 1 stellt die Schrittlänge des jeweiligen Fußes und den Gangzyklus dar.

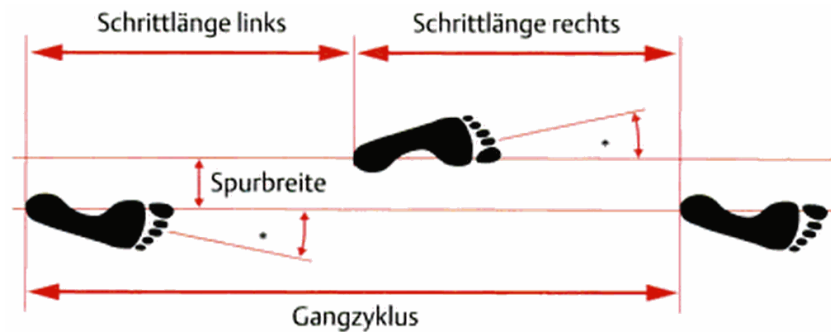


Abbildung 1: Schrittlänge (step length) und Doppelschrittlänge (stride length) [3]

Die Gehgeschwindigkeit beschreibt die zurückgelegte Strecke pro Zeiteinheit und wird in Meter pro Sekunde oder häufig auch in Meter pro Minute, für eine bessere Vergleichbarkeit mit der Kadenz, angegeben. Die Kadenz ist die Schrittfrequenz und wird in der Maßeinheit Schritte pro Minute angegeben. Als Referenzwerte für Erwachsene gelten eine Doppelschrittlänge von 1,4 m, eine Gehgeschwindigkeit von 1,4 m/s (84 m/min) und eine Kadenz von ca. 120 Schritten/min. Bedingt durch die kürzeren Beine bei Kindern, weisen diese im Alter von einem Jahr eine Kadenz von etwa 170 Schritte/min auf. Die stride length ist in dieser Altersklasse nahezu ident mit der Körpergröße. Der zurückgelegte Weg pro Sekunde entspricht ebenfalls derselben Größenordnung. Demnach hat ein Kind mit 0,6 m Körpergröße eine stride length von 0,6 m und bewegt sich mit 0,6 m/s fort. Mit sieben Jahren ist die Kadenz bei Kindern bereits auf 140 Schritte/min abgesunken. [3]

Der menschliche Gang wird charakterisiert durch einen singulären Gangzyklus, der sich konstant fortlaufend wiederholt. Um den Vorgang zu beschreiben, wird ein Referenzbein, in der Regel das dem Betrachter zugewandte Bein, ausgewählt. Der Zyklus beginnt mit dem initialen Bodenkontakt (Fersenkontakt) des Referenzbeines. Berührt dieses erneut den Boden ist ein Gangzyklus vollständig und somit abgeschlossen. Das kontralaterale Bein vollzieht während des Gangs analoge Bewegungsabläufe, jedoch um einen halben Gangzyklus verschoben. Unterteilt wird der Gangzyklus in die Stand- und die Schwungphase. Die Standphase umfasst bei einer üblichen Gehgeschwindigkeit etwa 60 %, die Schwungphase 40 % des Zyklus. Die Standphase ist der Zeitraum, in welchem der Fuß mit dem Boden verbunden ist. Die Schwungphase hingegen beschreibt die Dauer des zusammengesetzten Bewegungsablaufes, der vom Boden abgehoben vor sich geht. Es werden acht Phasen, fünf Stand- und drei Schwungphasen unterschieden. [1] [2] [3]

Folgende funktionelle Aufgaben gewährleistet der Gangzyklus [3]:

- Gewichtsübernahme
- Einbeinstand
- Vorwärtsbewegung des Schwungbeins

In der Standphase vollzieht sich die Gewichtsübernahme durch den Initial Contact und die Loading Response. Die zweite Aufgabe der Standphase, den Einbeinstand, führen die Phasen Mid-Stance, Terminal Stance und Preswing aus. Die Pre-swing Phase beendet den Einbeinstand und führt in den Schwungphasenbeginn über. In dieser Schwungphase wird die Vorwärtsbewegung durch den Initial-, Mid- und Terminal Swing generiert. [3]

Tabelle 1 gibt Aufschluss über die einzelnen Teilphasen des Gangzyklus, grafisch dargestellt ist der komplette Vorgang in Abbildung 2.

Tabelle 1: Phasen des Gangzyklus [1]

<b>Standphase</b>	IC (Initial Contact)	Erstkontakt über die Ferse	Doppelunterstützung
	LR (Loading Response)	Gewichtsübernahme über die Fußsohle (Doppelunterstützung)	Doppelunterstützung
	MSt (Mid Stance)	Standphasenmitte über die Fußsohle	Einzelunterstützung
	TSt (Terminal Stance)	Standphasenende (Einzelunterstützung)	Einzelunterstützung
	PSw (Preswing)	Schwungphasenvorbereitung	Einzel-, dann Doppelunterstützung
<b>Schwungphase</b>	ISw (Initial Swing)	Schwungphasenbeginn	Einzelunterstützung Gegenseite
	MSw (Mid Swing)	Schwungphasenmitte	Einzelunterstützung Gegenseite
	TSw (Terminal Swing)	Schwungphasenende	Einzel-, dann Doppelunterstützung

Die Bezeichnungen Einzel- und Doppelunterstützung stehen für die Anzahl der sich mit den in Bodenkontakt befindlichen Füßen.

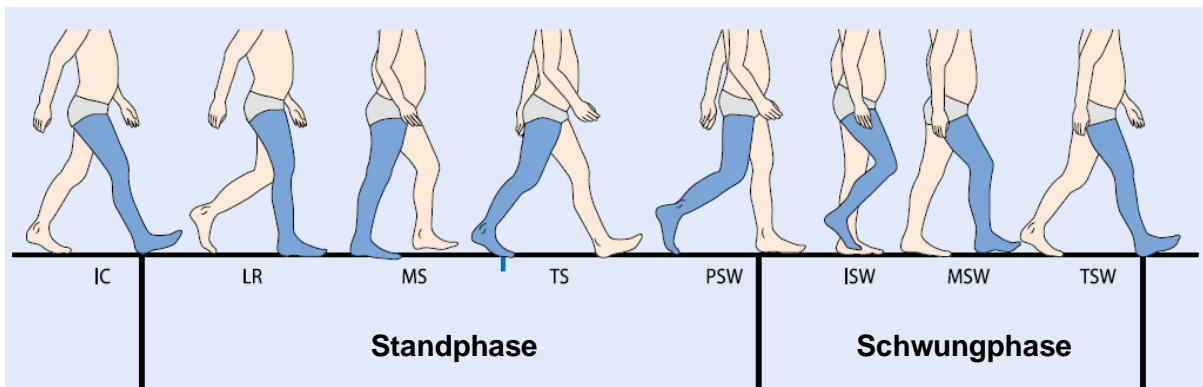


Abbildung 2: Der menschliche Gangzyklus vgl. [1]

Für die Lokomotionstherapie entscheidend ist ein physiologischer Bewegungsablauf. In Abbildung 3 werden die Winkel, welche die Gelenke in der Sagittalebene während des Gangzyklus vollziehen, analysiert.

Eine Flexion (Beugung) eines Gelenks ist für die Hüfte und die Knie als positiver Winkel definiert, während eine Extension (Streckung) einen negativen Winkel darstellt. Beim Sprunggelenk ist eine Dorsalextension einem positiven Winkel gleichzusetzen, während eine Plantarflexion durch einen negativen Winkel repräsentiert wird. [3]

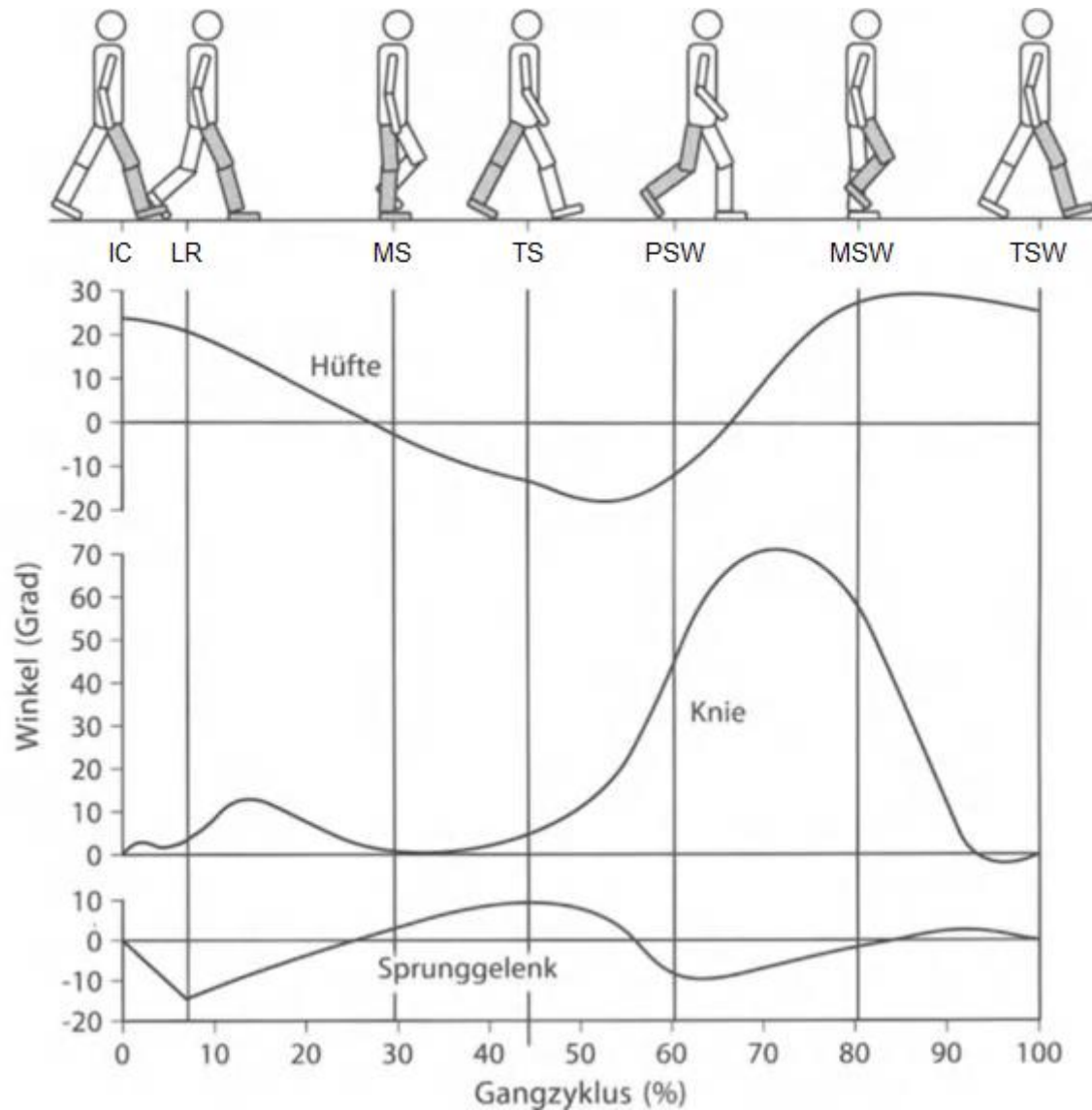


Abbildung 3: Winkelverläufe in Abhängigkeit des Gangzyklus vgl. [4]

Die in weitere Folge erforderlichen Winkelverläufe für die Hüfte und die Knie zur Generierung des Bewegungsmusters, wurden von der Klinik Judendorf-Straßengel zur Verfügung gestellt. Die Daten entstammen aus Ganganalysen mit 12 Kindern und Jugendlichen im Alter von 8 bis 15 Jahren.



## 1.2 Infantile Cerebralparese

Die infantile Cerebralparese ist eine Störung des Nerven- und Muskelsystems, die in Europa bei etwa zwei von tausend Kindern auftritt. Dabei gelten körperliche Behinderungen im Kindesalter mit Symptomen wie insbesondere einer Beeinträchtigung beim Gehen und bei der Bewegungskoordination als charakteristisch.

### 1.2.1 Definition und Häufigkeit

Die infantile Cerebralparese (ICP von lat. infantilis: kindlich, cerebrum: Gehirn und griech. palse: Lähmung), im Englischen als cerebral palsy bzw. cerebral paresis bezeichnet, stellt eine dauerhafte Bewegungs- bzw. Haltungsstörung dar, welche auf eine prä-, peri- oder postnatale Hirnschädigung bei Kleinstkindern zurückzuführen ist.

Eine umfangreiche Metastudie beziffert die Prävalenz mit 2,11 pro 1000 Lebendgeborenen, wobei aus statistischer Sicht bei Frühgeborenen das Risiko einer Cerebralparese um ein Vielfaches ansteigt. Obwohl sich die Überlebenschance bei frühgeborenen Säuglingen in den vergangenen Jahren verbessert hat, ist insgesamt kein Anstieg der Prävalenz von ICP zu verzeichnen. [5]

Zur infantilen Cerebralparese liegen divergierende Definitionen vor. Alle gemeinsam weisen jedoch auf eine frühkindliche Hirnschädigung hin, zu deren Auswirkungen Störungen des Nervensystems und der Muskulatur gehören, die wiederum Haltungs- und Bewegungsstörungen bei den Betroffenen hervorrufen. Die Hirnbeeinträchtigungen sind bleibend, hinsichtlich der unterschiedlich ausgeprägten Störungen der Motorik ist für die Patienten eine langfristige Betreuung notwendig, um die Folgen zu mildern. Das Krankheitsbild wird wie folgt definiert:

*„Cerebral palsy is an umbrella term covering a group of non-progressive, but often changing motor impairment syndromes secondary to lesions or anomalies of the brain arising in the early stages of development.“*

Zitiert aus [6]

### 1.2.2 Ätiologie

Die Ätiologie der ICP wird vom Zeitpunkt ihres Auftretens in prä-, peri- und postnatale Ursachen gegliedert. Am häufigsten liegt bei Kindern mit ICP eine pränatale Schädigung vor und nicht wie vermutet ein Sauerstoffmangel während, oder kurz nach der Geburt, obwohl auch die Hypoxie zu den Auslösern zählt. Pränatal können Chromosomenaberrationen, Hirnfehlbildungen während der Fetogenese im Mutterleib, Traumata sowie Gefäßverschlüsse der Plazenta Ursachen sein. Auch verschiedene Infektionskrankheiten während der Schwangerschaft können das Gehirn des ungeborenen Kindes schädigen. Zu diesen zählen

unter anderen Toxoplasmose, Cytomegalie oder Röteln. Im Übrigen stellen Medikamenten- oder Alkoholmissbrauch während der Schwangerschaft sowie Teratogene ebenfalls Risikofaktoren dar. Perinatale Ursachen können verzögerte bzw. komplizierte Entbindung als auch Durchblutungsstörungen (Nabelschnurumschlingung) sein. Herzstillstand, zerebrale Blutungen aber auch Hypoglykämie sind als ICP-Risikofaktoren nach der Geburt zu nennen. [1]

### 1.2.3 Klassifikation

Als internationaler Standard wird das sog. Gross Motor Function Classification System (GMFCS) anerkannt. Die Klassifizierung erlaubt es, den Schweregrad der Behinderung in 5 Stufen (GMFCS I–V) genau und zuverlässig zu beschreiben. Kinder der Stufe GMFCS I weisen nur leichte Einschränkungen auf, trotz welcher sie dennoch befähigt sind, ihren Tagesablauf gänzlich alleine zu bewältigen. Die Stufe GMFCS V steht für die stärksten Beeinträchtigungen, Betroffene haben weder Kopf- noch Rumpfkontrolle bzw. mehrfache Handicaps. [7] [8]

### 1.2.4 Symptome

Abhängig vom Bereich des Gehirns und dessen Entwicklungsphase, wo die entsprechende Schädigung eingetreten ist, werden als Symptome Spasmen, Ataxien (unkoordinierte Bewegungsabläufe), Athetosen (schlagartige Bewegungen) als auch Mischformen dokumentiert. Die Bewegungsstörungen betreffen am häufigsten eine Körperhälfte (Hemiparese), gefolgt von zwei Extremitäten (Diparese) oder allen vier Extremitäten (Tetraparese). [1]

Da sich die Schädigung des noch nicht vollständig entwickelten Gehirns nur selten auf die Mobilitätszentren allein beschränkt, können multiple Beeinträchtigungen auftreten [1]:

- Seh- und (seltener) Hörstörungen, wobei die Veränderungen des Sehens zentral (kortikal) oder peripher (z. B. Strabismus, Retinopathie) lokalisiert sein können
- Störungen der Intelligenz und der Aufmerksamkeit
- Verhaltensstörungen (exogene und endogene Ursachen)
- Epilepsien verschiedenster Ausprägungen mit lokalen und generalisierten Formen
- Dystrophien aufgrund von Kau- und Schluckproblemen
- Probleme im Zahn- und Kieferbereich
- intestinale Probleme (oft als gastroösophagealer Reflux)
- vegetative Störungen (Temperatur- und Schweißregulation)
- hormonelle Störungen (vorzeitige oder verzögerte Pubertät)
- Störungen der Körperwahrnehmung und der Schmerzempfindung
- Kommunikationsstörungen (expressiv oder rezeptiv)

### 1.2.5 Therapie

Die Behandlungsmöglichkeiten von ICP lassen sich überblicksmäßig in konservative und operative Techniken einteilen. Die Erstgenannten bilden die Basis der rehabilitativen medizinischen Behandlung und begleiten die Patienten vielfach ihr Leben lang. Die Therapiemöglichkeiten umfassen Physiotherapie, Ergotherapie, Logopädie, Orthopädie, Rehabilitationstechnik sowie Therapeutische Gipstechnik, medikamentöse Therapie und die Lokomotionstherapie. Häufig sind Kombinationen verschiedener Verfahren indiziert. [1] [8]

Die operativen Techniken umfassen Muskelverlängerungen, Sehnenverlängerungen, Sehnentranspositionen, Sehnenaugmentationen, Neurotomien, Osteotomien Epiphyseodesen (Wachstumslenkung oder -bremsung) Arthrodesen, Arthrorisen. [1]

## 1.3 Lokomotionstherapie

Unter dem Begriff der Lokomotionstherapie (von lat. locus: Ort, Stelle und motio: Bewegung) wird ein Gangtraining unter Einsatz eines Laufbandes oder Gangmaschinen mit partieller Körpergewichtsentlastung verstanden. Um den Patienten in einem aufrechten Stand über dem Laufband zu fixieren und ihn von einem Teil seines Körpergewichtes zu entlasten, gibt es verschiedenste Fixiervorrichtungen. Ein Gurtsystem, welches dem Patienten angelegt wird, verbunden mit einem Stahlseil, geführt über Rollensysteme aber auch pneumatische oder hydraulische Systeme sind im Einsatz. Oberstes Ziel der Lokomotionstherapie ist, das selbstständige Gehen im Alltag zu erlernen bzw. beizubehalten. Der Vorteil der Lokomotionstherapie gegenüber herkömmlichem Gehtraining mit einem Barren oder an Stöcken liegt im intensiveren Training der Muskeln als auch der Lokomotionszentren im Rückenmark. Das Verhindern von Muskelkontrakturen (permanente Muskelverkürzungen) und Gelenksteifen sowie Erhaltung bzw. Verbesserung der Gewebeelastizität werden der Lokomotionstherapie ebenfalls zugesprochen. Grundsätzlich wird zwischen manueller und automatisierter (roboter-assistierter) Lokomotionstherapie unterschieden. [1] [9]

Zu Beginn des manuellen Laufbandtrainings wird der Patient mit den entsprechenden Vorrichtungen idealerweise um mehr als die Hälfte seines Körpergewichtes entlastet. Mit zunehmender Gehfähigkeit kann diese Entlastung schrittweise wieder reduziert werden. Ein Physiotherapeut assistiert, indem er mit den Beinen ein physiologisches Gangmuster simuliert. Speziell in der anfänglichen Therapiephase stellt dies eine strapaziöse Tätigkeit dar, da der Patient die Bewegungen des Physiotherapeuten kaum oder gar nicht unterstützen kann. Somit ergeben sich Limitationen hinsichtlich der Dauer als auch Anzahl der Therapieeinheiten. Nachteilig könnte sich auch die Tatsache, dass das Gangmuster nicht exakt reproduzierbar ist, auswirken. [8] [9]

Die schlüssige Weiterentwicklung des manuellen Laufbandtrainings stellen automatisierte Varianten der Lokomotionstherapie dar. Der große Vorteil der roboter-gestützten Laufbandtherapie liegt gegenüber dem konventionellen Laufbandtraining in der Möglichkeit von längeren und intensiveren Trainingsphasen, unter Einhaltung eines annähernd physiologischen Gangbilds. Es existieren Konzepte mit bewegten Fußplatten<sup>1</sup> als auch bewegten Orthesen<sup>2</sup>. Im Falle von bewegten Fußplatten muss die Kniebewegung kontrolliert werden. Kritisch zu betrachten ist, dass sich die Fußsohlen sowohl in der Stand- wie auch in der Schwungphase auf den bewegten Fußplatten befinden und somit das Auftreten selbst, welches einen förderlichen Reiz der Lokomotionszentren darstellt, entfällt. [8] [9]

In Abbildung 4 sind die unterschiedlichen Konzepte von Lokomotionstherapie-Geräten ersichtlich.

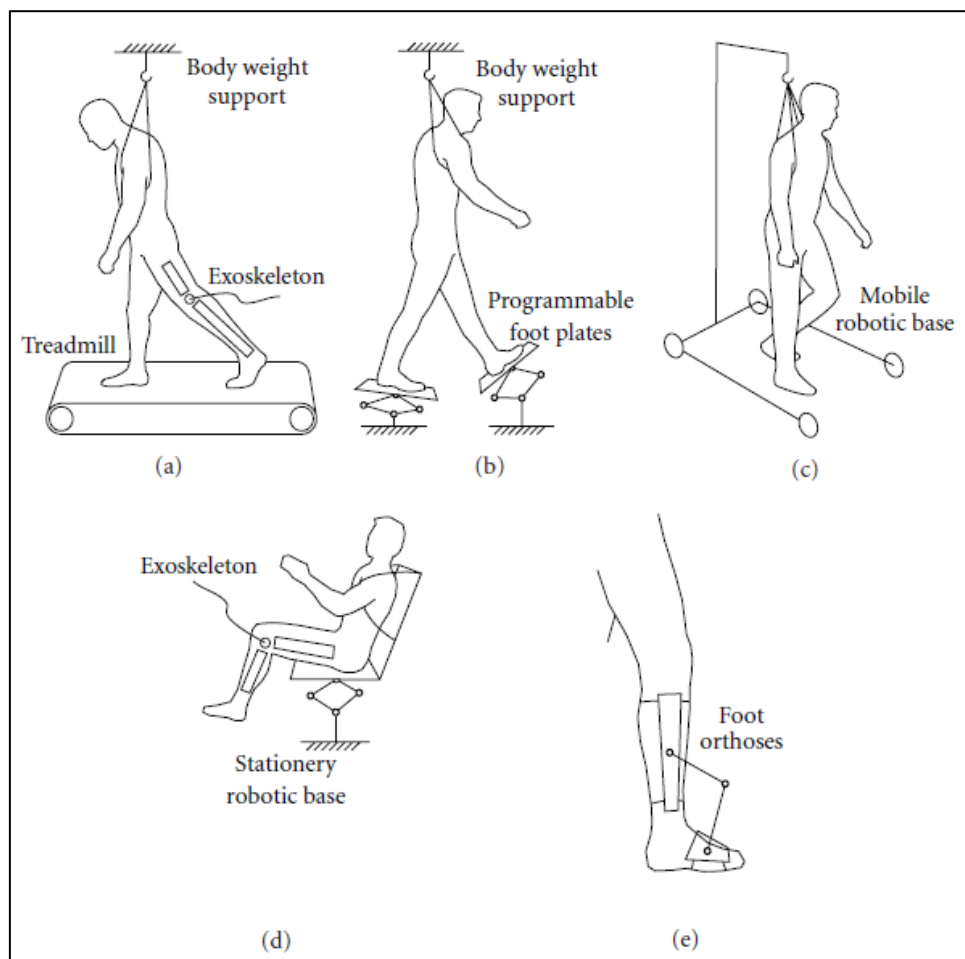


Abbildung 4: Verschiedene Systeme von Lokomotionstherapie-Geräten<sup>3</sup>

<sup>1</sup> Gait Trainer 1® (Reha-Stim, Deutschland)

<sup>2</sup> Lokomat® (Hocoma, Schweiz)

<sup>3</sup> Vgl. Bildquelle: [https://www.researchgate.net/publication/258380737\\_Lower-Limb\\_Robotic\\_Rehabilitation\\_Literature\\_Review\\_and\\_Challenges](https://www.researchgate.net/publication/258380737_Lower-Limb_Robotic_Rehabilitation_Literature_Review_and_Challenges)

## 1.4 Normative Rahmenbedingungen

Die normativen Rahmenbedingungen sind durch die Norm EN 62304: „Medizingeräte-Software - Software-Lebenszyklus-Prozesse“, welche eine Qualitäts- und Risikomanagement-Norm impliziert, gegeben. Des Weiteren kam die Maschinenrichtlinie im Zuge der Entwicklung zur Anwendung. Zweck der Norm 62304 ist die Definition von Anforderungen an den Lebenszyklus von Medizinprodukte-Software. Das Anwendungsgebiet deckt die Entwicklung und Wartung von Medizinprodukte-Software ab, wenn die Software selbst ein Medizinprodukt ist oder wenn die Software ein eingebetteter oder integraler Bestandteil des fertigen Medizinprodukts ist.

Die Norm definiert Medizinprodukte-Software als ein:

*„SOFTWARE-SYSTEM, das entwickelt wurde, um in das in der Entwicklung befindliche MEDIZINPRODUKT integriert zu werden, oder das für die Benutzung als selbständiges MEDIZINPRODUKT vorgesehen ist.“*

Zitiert aus [10]

Ein Software-System ist eine Sammlung von Software-Komponenten, die so organisiert ist, dass eine spezifische Funktion ausgeführt werden kann. Software-Komponenten stellen jedes identifizierbare Teil eines Computerprogramms (z.B. Quellcode) dar, während Software-Einheiten Software-Komponenten sind, die nicht weiter unterteilt werden können. [10]

Die Einhaltung der Norm EN 62304 basiert auf den allgemeinen Anforderungen der Medizinprodukterichtlinie 93/42/EWG, welche besagt:

*„Bei Produkten, die Software enthalten oder bei denen es sich um medizinische Software an sich handelt, muss die Software entsprechend dem Stand der Technik validiert werden, wobei die Grundsätze des Software-Lebenszyklus, des Risikomanagements, der Validierung und Verifizierung zu berücksichtigen sind.“*

Zitiert aus [11]

Beim Lokomotionsgerät handelt es sich um ein aktiv therapeutisches Medizinprodukt. Laut Medizinprodukterichtlinie 93/42/EWG ist dies ein:

*„Aktives Medizinprodukt, das entweder getrennt oder in Verbindung mit anderen Medizinprodukten eingesetzt wird und dazu bestimmt ist, biologische Funktionen oder Strukturen im Zusammenhang mit der Behandlung oder Linderung einer Krankheit, Verwundung oder Behinderung zu erhalten, zu verändern, zu ersetzen oder wiederherzustellen.“*

Zitiert aus [11]

Die Lokomotionsgeräte -Software gilt somit als integraler Bestandteil des aktiven therapeutischen Medizinprodukts.

Die allgemeinen Anforderungen der Norm EN 62304 verlangen im Zuge der Entwicklung der Software die Anwendung eines Qualitätsmanagement-Systems, eines Risikomanagement-Prozesses sowie die Durchführung einer Software-Sicherheitsklassifizierung. Das QM-System kann wahlweise ISO 13485, einer nationalen Norm für Qualitätsmanagement-Systeme oder einem Qualitätsmanagement-System, gefordert durch nationale Vorschriften entsprechen. Für den Risikomanagement-Prozess ist ISO 14971 anzuwenden. Der Hersteller ist verantwortlich für die Einordnung der Software in eine Sicherheitsklasse (A, B oder C), je nach dem Risiko eines Schadens, der aus der Gefährdungssituation resultiert. Der Umfang der Software-Lebenszyklus-Prozesse richtet sich nach der zuvor zugewiesenen Software-Sicherheitsklasse.

Die Norm EN 62304 definiert folgende Software-Lebenszyklus-Prozesse [10]:

- **Software-Entwicklungs-Prozess:** umfasst die Planung der Software-Entwicklung bis hin zur Freigabe der Software
- **Software-Wartungs-Prozess:** umfasst die Festlegung eines Plans für die Software-Wartung bis hin zur erneuten Freigabe der Software
- **Software-Risikomanagement-Prozess:** umfasst die Analyse von Software, die zu Gefährdungssituationen beiträgt bis hin zum Risikomanagement von Software-Änderungen
- **Software-Konfigurationsmanagement-Prozess:** umfasst die Identifizierung der Konfiguration bis hin zu Aufzeichnungen über den Status der Konfiguration
- **Problemlösungs-Prozess für Software:** umfasst das Erstellen von Problembereichten bis hin zum Inhalt der Prüfungsdokumentation

Im Rahmen der ursprünglichen Softwareerstellung (M. Kalkgruber [12]) wurde kein Qualitätsmanagement-System erstellt, um den Umfang jener Masterarbeit zu begrenzen. Dies wird auch in gegenständlicher Arbeit weiterhin so gehandhabt. Der Software-Wartungs-Prozess beschreibt die Wartung der Software nach deren Freigabe. Da das Ziel der vorangegangenen Arbeit im Entwickeln der Software bestand und Gegenstand dieser Arbeit eine Funktionserweiterung der Hardware und ein Software Re-Design darstellen, findet der der Freigabe nachgelagerte Software-Wartungs-Prozess in beiden Arbeiten keine Berücksichtigung.

## 2 Aufgabenstellung

Ziel dieser Arbeit ist es, eine verbesserte Steuerungs-Software eines neu entwickelten Lokomotionsgerätes für Kleinkinder zu realisieren. Dieses sogenannte Re-Design schließt eine Erweiterung der Funktionalität durch Implementieren neuer Hardware mit ein.

Folgende Anforderung an die Hardware sollen umgesetzt werden:

- Ergänzung von Kraftsensoren zur Bestimmung der Gewichtsbelastung
- Realisierung eines automatisierten (aktiven) Laufbandes
- Detektion der Betätigung des Not-Aus-Schalters
- Hinzufügen eines Fußschalters für die Aufmerksamkeitsüberwachung

Folgende Anforderung an die Software sollen umgesetzt werden:

- Überarbeitung der Usability der Software auf Grundlage der vorangegangenen Masterarbeit „Beurteilung der Gebrauchstauglichkeit eines Lokomotionsgerätes für Kleinkinder“ [13] von S. Langthaler
- Gewichtsmessung des Patienten durch Auswertung der Kraftsensoren
- Anpassung des Laufbandes an die Bewegungsprofile der Orthesen
- Alarmierung bei aktivem Not-Aus Schalter in der Software

Die Erweiterung der Software soll unter Berücksichtigung der Norm 62304 „Medical device software – Software life cycle processes“ umgesetzt werden. Die erstellten Software-Komponenten sollen umfassend unter Einhaltung der Anforderungen der Norm sowie des bestehenden Konzepts dokumentiert werden. Dies schließt die Aktualisierung des Risikomanagement-Prozesses für die Software, insbesondere in Bezug auf die neuen Funktionalitäten, mit ein. Abschließend wird die finalisierte Soft- und Hardware in das im Vorfeld neu entwickelte Lokomotionsgerät integriert.

### 3 Methoden

Im folgenden Abschnitt werden die theoretischen Grundlagen der Medizingeräte-Software-Norm EN 62304 beschrieben, sowie deren praktische Umsetzung hinsichtlich der im vorherigen Abschnitt festgehaltenen Aufgabenstellung.

#### 3.1 Risikomanagement

Die Norm EN 62304 [10] fordert die Anwendung eines Risikomanagement-Prozesses nach ISO 14971 [14] durch den Hersteller. In diesem Kapitel wird der generelle Ablauf des Risikomanagement-Prozesses nach EN ISO 14971 erläutert.

Demnach wird ein Risiko durch die Norm als Kombination der Wahrscheinlichkeit des Auftretens eines Schadens und des Schweregrads dieses Schadens definiert [14].

Für Medizinprodukte muss der Hersteller eine Risikomanagementakte anlegen und aktuell halten. Diese gewährleistet die Rückverfolgbarkeit jeder festgestellten Gefährdung. Ein Risikomanagementplan, welcher Teil der Risikomanagementakte ist, dokumentiert die entsprechend geplanten Tätigkeiten. [14]

Die Schritte des Risikomanagement-Prozesses sind in Abbildung 5 dargestellt und werden in den folgenden Kapiteln erklärt.

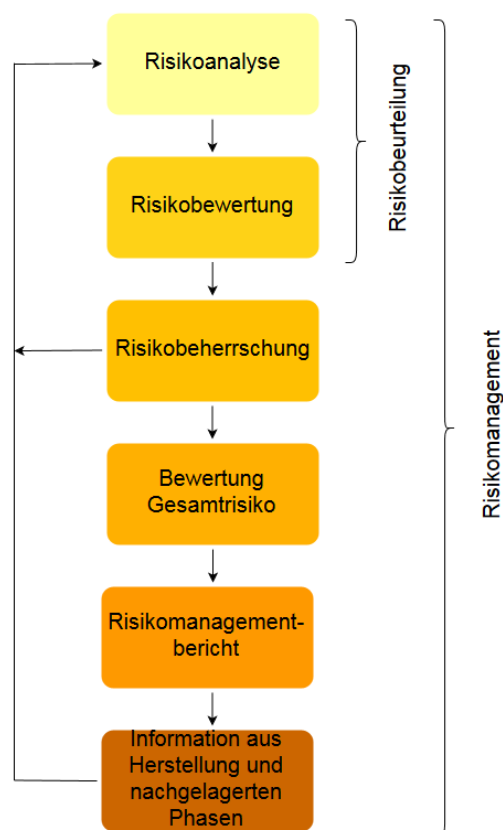


Abbildung 5: Ablaufdiagramm Risikomanagement-Prozess vgl. [14]



### 3.1.1 Risikoanalyse

Zuerst muss der Hersteller des jeweiligen Medizinprodukts die Zweckbestimmung festlegen und den vernünftigerweise vorhersehbaren Missbrauch dokumentieren. Unter dem Begriff Missbrauch versteht die Norm eine fehlerhafte oder ungeeignete Anwendung des Medizinprodukts. Im nächsten Schritt müssen Gefährdungen des Medizinprodukts identifiziert werden. Dazu muss der Hersteller bekannte und vorhersehbare Gefährdungen dokumentieren, die sowohl unter Normal- wie auch unter Fehlerbedingungen auftreten können. Entsprechende Hilfsmittel der Risikoanalyse werden von der Norm beschrieben, dazu gehören unter anderen die „Vorläufige Gefährdungsanalyse (PHA)“, die „Fehlerbaumanalyse (FTA)“, die „Fehlermöglichkeits- und Einflussanalyse (FMEA)“ und weitere. [14]

Die im Vorfeld bereits bestehende Risikoanalyse auf Basis der Software-Architektur wurde durch das Re-Design fortgeführt. Anhand der Anforderungen und Funktionen wurden für die Lokomotionsgerätesoftware mögliche Fehler, beziehungsweise deren Ursachen abgeleitet. Verwendet wurde hierfür die Fehlermöglichkeits- und Einflussanalyse (Failure Mode and Effects Analysis, FMEA).

Die FMEA ist ein Hilfsmittel mit welchem Fehlermöglichkeiten systematisch untersucht und bewertet werden können, dabei wird ein Bottom-Up Ansatz verfolgt. Es wird von einem möglichen Fehlerfall ausgegangen und dessen Auswirkungen über die einzelnen Ebenen der Wirkungskette bis zum endgültigen Schadensereignis betrachtet. Die Fehlerbaumanalyse geht den umgekehrten Weg. Es wird vom Schadensereignis ausgegangen und mit einem Top-Down Ansatz analysiert, welche Möglichkeiten zur Ursache führen könnten. [14] [15]

Eine Gegenüberstellung ist in Abbildung 6 ersichtlich.

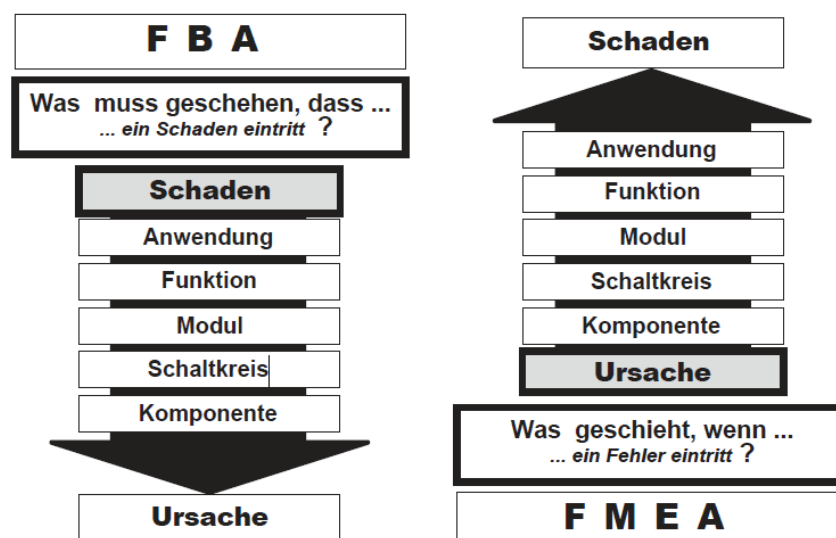


Abbildung 6: Fehlerbaumanalyse (FBA / FTA) und Fehlermöglichkeits- und Einflussanalyse (FMEA) [15]

Nachdem die Risiken identifiziert wurden, müssen diese hinsichtlich Eintrittswahrscheinlichkeit und Schadensfolge klassifiziert werden. Die sogenannte Risikobewertungsmatrix weist den Gefährdungen in Abhängigkeit der Eintrittswahrscheinlichkeit und der Schadensfolge eine Risikostufe (I-IV) zu, siehe Abbildung 7.

Eintrittswahrscheinlichkeit	6	häufig					Risikostufe
	5	manchmal				I	
	4	gelegentlich					
	3	selten				II	
	2	unwahrscheinlich				III	
	1	unglaublich				IV	
			gering	mittel	schwer	katastrophal	
			1	2	3	4	
			Schadensfolge				

Abbildung 7: Risikomatrix mit Risikostufen I bis IV

Für eine exaktere Einordnung der Schäden sind diese in Tabelle 2 spezifiziert.

Tabelle 2: Beschreibung der qualitativen Schadensfolgen zur Risikobewertung

Nummer	Bezeichnung	Bedeutung
1	gering	keine Verletzung
2	mittel	leichte Verletzung
3	schwer	schwere Verletzung oder Tod
4	katastrophal	schwere Verletzung oder Tod für mehrere Personen

Die im Zuge der Risikoanalyse durchgeführten Tätigkeiten und deren Ergebnisse sind gesammelt in der Risikomanagementakte zu dokumentieren.

### 3.1.2 Risikobewertung

Bei der Risikobewertung geht es darum, die Akzeptierbarkeit eines Risikos festzustellen. Anhand der zuvor beschriebenen Risikomatrix ergeben sich vier Risikostufen. Risiken der Stufe 1 sind nicht tolerierbar. Ausnahmefälle würden sich nur ergeben, wenn keine Alternative vorhanden ist und der Nutzen gegenüber dem Risiko in hohem Maße überwiegt. Risiken der Stufe 2 wären nur dann akzeptierbar, wenn das Risiko soweit minimiert wurde, wie es vernünftigerweise erreichbar ist. Dieses Prinzip wird ALARA (as low as reasonably achievable) genannt. Risiken der Stufe 3 können akzeptiert werden, wenn alle

vernünftigerweise praktikablen Möglichkeiten, das Risiko zu minimieren, unternommen wurden. Es gilt das ALARP-Prinzip (as low as reasonably practicable). Stufe 4 stellt akzeptierbare niedrige Risiken dar. Weitere Risikominderungen sollten im Fall von kostengünstigen bzw. einfachen Möglichkeiten realisiert werden. [15]

Zusammengefasst stellt Tabelle 3 die Risikostufen und ihre Bedeutung dar.

*Tabelle 3: Risikostufen (I bis IV) und ihre Bedeutung*

Risikostufe	Bedeutung
I	Nicht akzeptierbares Risiko
II	ALARA: kann nur toleriert werden, wenn eine Vermeidung nicht möglich ist
III	ALARP: kann nur toleriert werden, wenn eine Vermeidung nicht praktikabel ist
IV	Tolerierbar: vernachlässigbares Risiko

Für das Lokomotionsgerät werden im Gegensatz des zu erwartenden medizinischen Nutzens Risiken der Stufe I und II abgelehnt, auch wenn die Norm das in Betracht ziehen eventueller Ausnahmefälle vorsieht. Risiken der Stufe III werden nur dann toleriert, wenn deren Reduktion mit einem unproportional hohen Aufwand im Vergleich zum Nutzen steht.

### 3.1.3 Risikobeherrschung

Wird in der Risikobewertung festgestellt, dass eine Risikominderung erforderlich ist, müssen Aktivitäten zur Risikobeherrschung durchgeführt werden. Unter der Risikobeherrschung versteht man eine oder mehrere adäquate Maßnahmen, um das Risiko auf ein akzeptables Niveau zu mindern. Hierfür kann der Schweregrad des Schadens und/oder die Wahrscheinlichkeit seines Auftretens gemindert werden. [14]

Zur Durchführung der Risikobeherrschung eignet sich der Grundsatz der integrierten Sicherheit. Dieser besagt, dass die Sicherheitsmaßnahme mit der höchst möglichen Effizienz gewählt werden muss. Die Schutzmaßnahmen lassen sich dabei in drei Kategorien, mit absteigender Wirksamkeit, einteilen [15]:

1. **Unmittelbare (konstruktive) Sicherheit:** Ist überall dort sicherzustellen, wo dies möglich und wirtschaftlich vertretbar ist. Ein Beispiel für eine konstruktive Maßnahme im Falle des Lokomotionsgeräts wäre die Abdeckung rotierender Teile.
2. **Mittelbare (unterstützende) Sicherheit:** Sind konstruktive Maßnahmen nicht möglich oder sinnvoll, kommen unterstützende Maßnahmen zum Einsatz. Ein Beispiel für eine unterstützende Maßnahme im Falle des Lokomotionsgeräts stellt die Aufmerksamkeitssteigerung durch Quittierung dar.
3. **Hinweisende Sicherheit:** Dient der Ergänzung der Schutzmaßnahmen. Ein Beispiel hierfür ist die Gebrauchsanweisung zur Lokomotionsgerätesoftware.

Nach Durchführung der Maßnahmen der Risikobeherrschung muss das verbleibende Restrisiko erneut bewertet werden. Wird es noch immer als inakzeptabel beurteilt, müssen weitere Maßnahmen zur Risikobeherrschung umgesetzt werden. Sind weitere Maßnahmen nicht realisierbar muss der Hersteller den Nachweis erbringen, dass der medizinische Nutzen gegenüber dem Restrisiko überwiegt. Anzumerken ist, dass durch Risikobeherrschungsmaßnahmen neue Risiken entstehen können. Diese müssen erneut den Prozess durchlaufen. [14]

#### **3.1.4 Bewertung der Akzeptanz des Gesamt-Restrisikos**

Nach Umsetzung aller Risikobeherrschung-Maßnahmen muss verifiziert werden, ob das Gesamt-Restrisiko akzeptabel ist. Ein gleichzeitiges Eintreten von mehreren tolerierbaren Einzelrisiken kann zu einem inakzeptablen Gesamtrisiko führen. [14]

#### **3.1.5 Risikomanagementbericht**

Vor der Freigabe eines Medizinprodukts muss der Hersteller den Prozess des Risikomanagements überprüfen. Der Risikomanagementbericht bestätigt das Vorhandensein eines geeigneten Risikomanagementplans, eines akzeptierbaren Gesamt-Restrisikos und geeigneter Methoden, um Information aus Herstellung und nachgelagerter Phasen zu erhalten. Der Bericht ist Teil der Risikomanagementakte. [14]

#### **3.1.6 Informationen aus Herstellung und nachgelagerten Phasen**

Die Norm schreibt das Sammeln von Informationen über das Medizinprodukt während der Herstellung und der Herstellung nachgelagerten Phasen vor. Es sollten auch öffentlich zugängliche Informationen über vergleichbare auf dem Markt befindliche Produkte gesammelt werden. Die kumulierten Daten müssen hinsichtlich Sicherheitsrelevanz bewertet werden, um nicht erkannte Gefährdungen zu identifizieren bzw. nicht länger akzeptierbare Risiken zu erkennen. Der letztgenannte Fall kann sich durch eine Weiterentwicklung der Technik ergeben. [14]

### 3.2 Software-Sicherheitsklassifizierung

Die entwickelte Software muss laut EN 62304 einer Software-Sicherheitsklasse (A, B oder C) durch den Hersteller zugeordnet werden. Abhängig von dieser Zuordnung schreibt die Norm die durchzuführenden Aktivitäten sowie den Umfang der zu erstellenden Dokumentation in unterschiedlichem Ausmaß vor. Die Klassen unterscheiden sich nach dem Risiko eines möglichen potentiellen Schadens für den Patienten, den Anwender oder dritte Personen. Bei der Bestimmung der Sicherheitsklasse muss grundsätzlich ein Versagen der Software mit einer hundertprozentigen Wahrscheinlichkeit angenommen werden. Deshalb sind nur externe Risikobeherrschungsmaßnahmen, die nicht Teil der Software sind, hinsichtlich ihrer Effektivität zu betrachten. Abbildung 8 zeigt den Prozess des Zuweisens zu einer Sicherheitsklasse. [10]

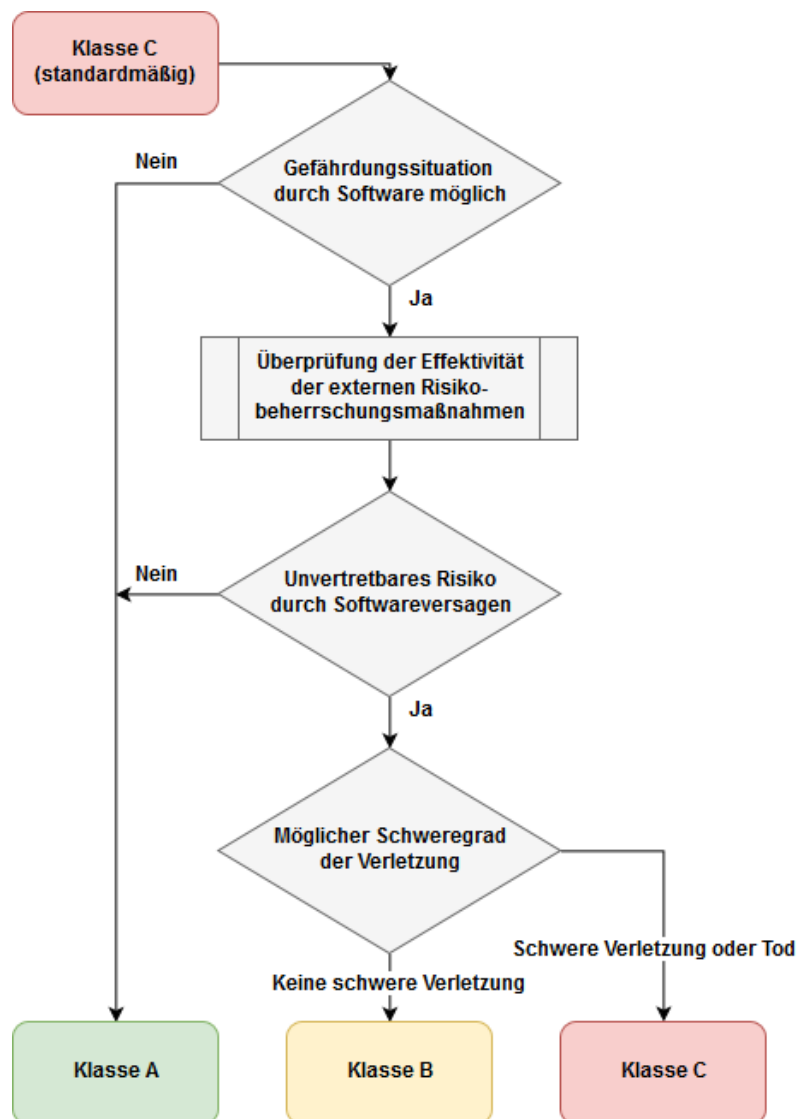


Abbildung 8: Software-Sicherheitsklassifizierungs-Prozess vgl. [10]

Die durchgeführte Software-Sicherheitsklassifizierung des Re-Designs der Lokomotionsgerätesoftware basiert auf der Bewertung der neuen Risiken einschließlich der vorhandenen Risiken laut dem Risikoprotokoll von M. Kalkgruber [12].

Das Lokomotionsgerät verwendet unter anderem hardwareseitige Vorrichtungen (z.B. Winkelbeschränkungen) als externe Risikobeherrschungsmaßnahmen.

Die Gefährdung, die von der Software der jeweiligen Klasse ausgeht, wird in der Norm wie folgt klassifiziert [10]:

- **Klasse A:** Das Software-System kann nicht zu einer Gefährdungssituation beitragen oder es trägt zu einer Gefährdungssituation bei, die externen Risikobeherrschungsmaßnahmen verhindern allerdings ein inakzeptables Risiko. Software der Klasse A kann demnach zu keiner Gefährdung der Gesundheit führen.
- **Klasse B:** Das Software-System kann zu einer Gefährdungssituation beitragen. Die externen Risikobeherrschungsmaßnahmen können ein unvertretbares Risiko nicht verhindern, jedoch stellt der Schaden keine schwere Verletzung dar.
- **Klasse C:** Das Software-System kann zu einer Gefährdungssituation beitragen. Die externen Risikobeherrschungsmaßnahmen können ein unvertretbares Risiko nicht verhindern. Der Schaden kann schwere Verletzung oder Tod sein.

Ein Software-System, das ursprünglich als Klasse B oder C eingestuft war, kann durch zusätzliche externe Risikobeherrschungs-Maßnahmen eventuell einer neuen Sicherheitsklasse zuzuordnen sein. [10]

Da der nachfolgende Software-Entwicklungs-Prozess maßgeblich durch die Software-Sicherheitsklasse beeinflusst wird, wird vorab das Ergebnis bekannt gegeben. Unter Berücksichtigung aller Risiken ergibt sich Klasse B für die Lokomotionsgeräte-Software, siehe Kapitel 4.1 Software-Sicherheitsklassifizierung.

### 3.3 Software-Entwicklungs-Prozess

Der Software-Entwicklungs-Prozess stellt den Hauptbestandteil der Norm EN 62304 dar. In Abhängigkeit der zugewiesenen Software-Sicherheitsklasse werden die im Zuge der Entwicklung durchzuführenden Aktivitäten und Aufgaben vorgeschrieben. Die Norm unterteilt den Prozess in folgende Aktivitäten [10]:

- Planung der Software-Entwicklung
- Analyse der Software-Anforderungen
- Design der Software-Architektur
- Detailliertes Software-Design
- Implementierung der Software-Einheiten
- Software-Integration und Integrationsprüfung
- Prüfung des Software-Systems
- Software-Freigabe

Die einzelnen Punkte werden auf den nachfolgenden Seiten detaillierter ausgeführt.

#### 3.3.1 Planung der Software-Entwicklung

Den Grundstein der Softwareentwicklung bildet der sog. Software-Entwicklungsplan, welcher die Durchführung der Aktivitäten des Software-Entwicklungs-Prozesses dokumentiert. Das verwendete Lebenszyklus-Modell der Software-Entwicklung muss im Plan definiert bzw. referenziert sein. Den Mindestanforderungen entsprechend müssen die Prozesse, die bei der Entwicklung zur Anwendung kommen, dokumentiert werden. Die von der Software zu erzielenden Ergebnisse und die dafür notwendigen Aktivitäten müssen ebenfalls in der Dokumentation enthalten sein. Die Rückverfolgbarkeit zwischen den System- und Software-Anforderungen, den Prüfungen sowie den Risikokontrollmaßnahmen muss gegeben sein. Ein Software-Konfigurations- und Änderungsmanagement muss implementiert werden. Letztendlich fordert die Norm noch eine schriftliche Software-Problemlösungs-Strategie und die Behandlung allfälliger Probleme der verschiedenen Lebenszyklus-Phasen. [10]

Für die bestehende Lokomotionsgerätesoftware wurde ein modifiziertes V-Modell als Entwicklungsstrategie gewählt. Das V-Modell besteht aus einem absteigenden linken Ast, welcher der konstruktiven Phase entspricht und einem aufsteigenden rechten Ast, welcher die Testphase darstellt. Sogenannte System Anforderungen, welche die übergeordneten Anforderungen in höchster Ebene darstellen, werden in der konstruktiven Phase genutzt, um immer detailliertere Anforderungen abzuleiten. Dies mündet im Detail-Design der einzelnen Software-Einheiten. Ausgehend von der untersten Ebene werden die Testphasen, die jeder konstruktiven Phase laut V-Modell gegenübergestellt sind, aufsteigend geprüft. Dabei

werden zuerst die Anforderungen des Detail-Designs mit den erstellten Software-Komponenten verglichen. In der obersten Ebene angelangt, werden abschließend die Software-Anforderungen durch eine Prüfung des Software-Systems verifiziert. Dies führt letztendlich zur Freigabe der Software. Im Anhang in Abbildung 62 ist das vollständige V-Modell ersichtlich. Eine ausführlichere Beschreibung befindet sich in [12].

### 3.3.2 Analyse der Software-Anforderungen

In diesem Abschnitt geht es um die Ableitung der Software-Anforderungen aus den übergeordneten System-Anforderungen und deren Dokumentation. Dargestellt ist der Ablauf in Abbildung 9.

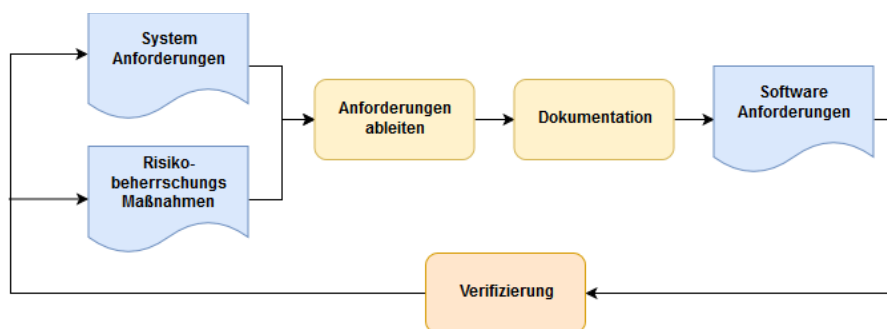


Abbildung 9: Ableitung der Software-Anforderungen

Die Norm spezifiziert den Inhalt der Software-Anforderungen, wobei einzelne Punkte ineinander übergreifen dürfen. Die Aufnahme der Risikobeherrschungs-Maßnahmen in die Anforderungen wird ebenfalls vorgeschrieben. [10]

Die System- und Software-Anforderungen inklusive der Risikomaßnahmen wurden ausführlich dokumentiert und im sogenannten Traceability-Dokument referenziert. Dies gewährleistet die geforderte Rückverfolgbarkeit der EN 62304. Eine dokumentierte System-Anforderung resultiert in zumindest einer Software-Anforderung, es können jedoch auch mehrere daraus hervorgehen. Die Verifizierung wird durch sog. Software-System-Tests durchgeführt. Ihre Beschreibung und deren Ergebnis befindet sich ebenso im Traceability Dokument. Siehe auch [12].

Anmerkung: Für die Entwicklung der Hard- und Software des Lokomotionsgeräts wurde die Maschinenrichtlinie („Richtlinie 2006/42/EG des Europäischen Parlaments und des Rates vom 17.Mai 2006 über Maschinen und zur Änderung der Richtlinie 95/16/EG“) [16] eingehalten. [12] [17] [18]

Zutreffende Vorschriften der Richtlinie wurden mittels entsprechender Anforderungen umgesetzt.



### 3.3.3 Design der Software-Architektur

Aus den Software-Anforderungen muss eine dokumentierte Software-Architektur erstellt werden. Diese beschreibt die Struktur der Software und identifiziert die Software-Komponenten. [10]

Der prinzipielle Prozess ist in Abbildung 10 dargestellt.

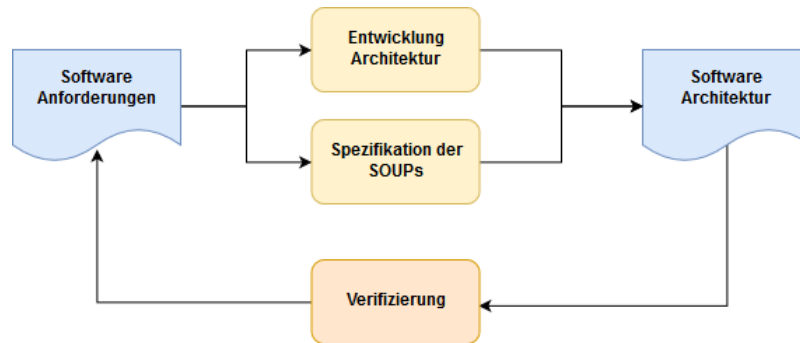


Abbildung 10: Ableitung der Software-Architektur

Die Architektur wurde in eine statische und dynamische Betrachtung unterteilt. Die statische Sicht beschreibt die Software-Komponenten und ihre Schnittstellen untereinander bzw. zur Hardware. Die dynamische Sicht beschreibt den Ablauf als auch die Funktionalität des Software-Systems. Flussdiagramme werden zur dynamischen Darstellung verwendet. Die zur Spezifikation der SOUPs (Software unbekannter Herkunft) vorgeschriebenen Funktions- und Leistungsanforderungen sowie die erforderliche Hardware und Software des Systems wurde dokumentiert. Für die abschließende Verifizierung wurde eine Checkliste generiert, deren Kriterien erfüllt werden müssen. [12]

Die Medizingeräte-Software EN 62304 bezeichnet Software unbekannter Herkunft (Software of unknown provenance) als SOUP und definiert sie folgendermaßen:

*„Software-Komponente, die bereits entwickelt und allgemein verfügbar ist und die nicht entwickelt wurde, um in das Medizinprodukt integriert zu werden (auch bekannt als "Off-The-Shelf-Software"), oder zuvor entwickelte Software-Komponente, für die angemessene Aufzeichnungen zum Entwicklungs-Prozess nicht verfügbar sind.“*

Zitiert aus [10]

### 3.3.4 Detailliertes Software-Design

Die entwickelte Software muss so lange weiter unterteilt werden, bis sie durch Software-Einheiten dargestellt wird. Eine Software-Einheit wird definiert als Software-Komponente, die nicht in weitere Komponenten unterteilt ist. Die Feinheit der Unterteilung obliegt dem Hersteller. Ausschließlich für Software der Sicherheitsklasse C muss ein Design Dokument für jede Software-Einheit entwickelt werden. Es muss detailliert genug sein, um eine korrekte Implementierung zu ermöglichen. [10]

Es wurde, obwohl die Lokomotionsgerätesoftware der Sicherheitsklasse B entspricht und es somit nicht von der Norm gefordert wäre, ein Detail-Design für jede Einheit angelegt, um die Implementierung, Wartung und das Testen zu vereinheitlichen. Zusätzlich kann die Dokumentation nachfolgenden Entwicklern eine enorme Hilfestellung darstellen.

Das für jede Software-Einheit angelegte Detail-Design-Dokument beinhaltet folgende Informationen:

- Autor der Software bzw. des Dokuments
- Datum der Erstellung
- Version der Software
- Änderungen zur vorherigen Version der Software
- Software-Sicherheitsklasse
- Beschreibung der Software-Komponente
- Anforderung an die grafische Benutzeroberfläche (Frontpanel)
- Beschreibung der Funktionalität
- Verweis auf eventuelle umgesetzte Software-Anforderung
- Definition der Schnittstellen:
  - Speicherzugriff
  - Verwendete Unterprogramme (sog. SubVIs)
  - Verwendete SOUPs
  - Externe Schnittstellen
  - Variablen:
    - Name
    - Schnittstellen-Typ
    - Daten-Typ
    - Bemerkung

Zusätzlich zu den genannten Punkten haben Detail-Design-Dokumente von modifizierten Software-Komponenten, im Zuge des Re-Designs einen Eintrag für die Person, die die letzte Änderung durchgeführt hat und das Datum der letzten Änderung erhalten.

In Abbildung 11 ist ein Detail-Design-Dokument für eine neu entwickelte Softwarekomponente dargestellt.

DD-0.4 Select Printer-V-0.docx

**Detaildesign: DD-0.4 Select Printer-V-0.docx**

Autor: Matthias Prim  
Datum: 19.10.2018  
Version: V-0  
Änderungen zur vorherigen Version:  
 Erstversion  
Software-Sicherheitsklasse: B

Beschreibung:  
 Das VI zeigt eine GUI zur Auswahl eines verfügbaren Druckers an.

Frontpanel:

- Buttons:
  - Ring: Dropdown-Menü zur Druckerauswahl
  - Boolean: Drucker auswählen
  - Boolean: Abbrechen

Funktionalität:

1. Es kann ein verfügbarer Drucker aus dem Dropdown-Menü ausgewählt werden.
2. Durch Drücken von „Drucker auswählen“ wird der entsprechende Druckername in Form eines Strings retourniert.
3. Durch Drücken von „Abbrechen“ wird „Kein Drucker gewählt“ in Form eines Strings retourniert.

Umgesetzte Anforderungen:

	keine
--	-------

Schnittstellen:

- Speicherzugriff: kein Speicherzugriff
- SubVI: keines
- SOUP: keine Verwendung von SOUPs

• Variablen:

Name	Schnittstellen-Typ	Daten-Typ	Bemerkung
Error	Eingang / Ausgang / Intern	Error-Cluster	Cluster mit Fehlerinformationen
Auswahl	Intern	Ring U16	Zeigt die verfügbaren Drucker an
Drucker auswählen	Intern	Boolean	Beendet VI und gibt jeweiligen Druckernamen zurück
Abbrechen	Intern	Boolean	Beendet VI und gibt „Kein Drucker gewählt“ zurück
Druckername	Ausgang	String	Name (Bezeichnung) des Druckers

Seite 1 von 1

Abbildung 11: Exemplarisches Detail-Design-Dokument (0.4 Select Printer-V-0)

### 3.3.5 Implementieren–Verifizieren-Integrieren

Da entwickelte Programme mit LabVIEW eine hierarchische Struktur aufweisen, wurden die von der Norm EN 62304 geforderten Aktivitäten „Implementierung der Software-Einheiten“ sowie „Software-Integration und Integrationsprüfung“ zu einer iterativen Aktivität mit dem Namen „Implementieren-Verifizieren-Integrieren“ (IVI) zusammengefasst. Die Vorgehensweise orientiert sich an einer Bottom-Up-Integrationsstrategie, siehe Abbildung 12.

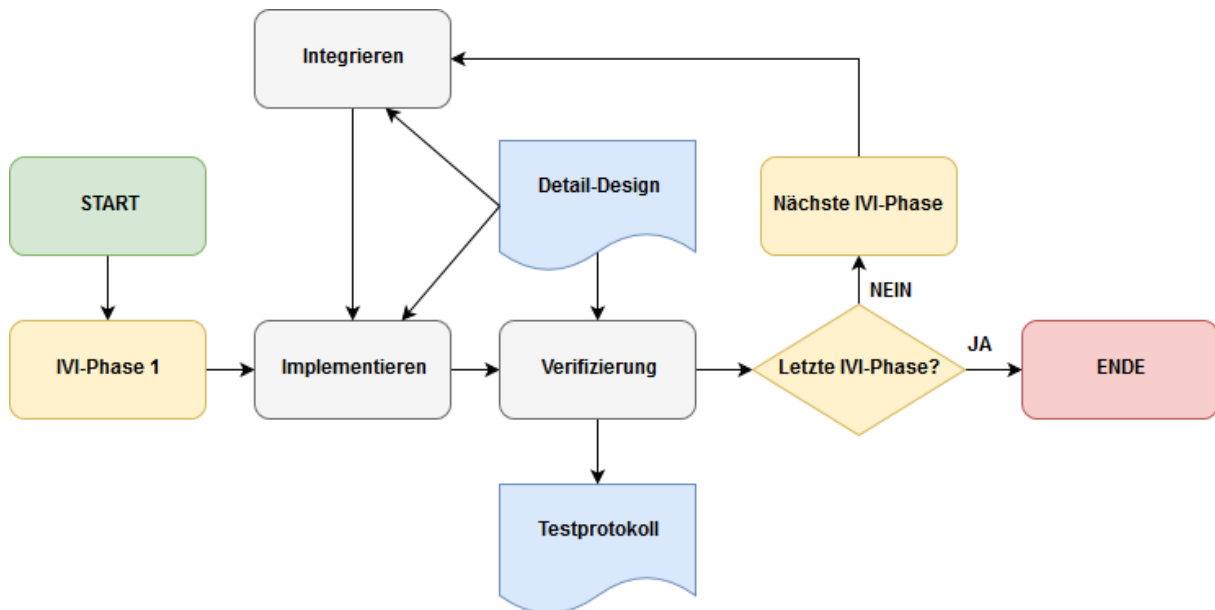


Abbildung 12: Implementieren-Verifizieren-Integrieren Aktivität

Im ersten Durchgang werden alle benötigten Software-Einheiten der untersten Ebene implementiert. Nach der vollständigen Umsetzung werden die Einheiten verifiziert. Erst nachdem die Verifizierung vollständig ist darf mit der nächsthöheren Phase begonnen werden. Die zuvor verifizierten Einheiten können nun in die darüber liegende Software-Komponente integriert werden. [12]

Die Verifizierung der einzelnen Software-Komponenten wird in Testprotokollen dokumentiert. Ein sogenanntes Codereview stellt neben einem korrekten Programmierstil inklusive Dokumentation des Codes auch die Einhaltung der Anforderungen des Detail-Designs sicher. Anschließend an das Codereview wird ein Funktionstest der Software-Komponente durchgeführt. Hierfür sind exakt durchzuführende Anweisungen und die erwarteten Ereignisse vorgegeben. Treffen sämtliche Ereignisse zu, hat die Komponente die Verifizierung erfolgreich bestanden. Das Gesamtergebnis der Software-Komponente wird zu Beginn des Testprotokolls vermerkt. Ein Beispiel für ein Testprotokoll einer modifizierten Software-Komponente ist in Abbildung 13 und Abbildung 14 ersichtlich.

Testprotokoll: TP-1.3 Zugangsbeschränkung-V-0.2-1.docx

## Testprotokoll: TP-1.3 Zugangsbeschränkung-V-0.2.docx

Test Autor: Matthias Prim

Datum: 10.11.2018

Freigabe: Matthias Prim

Datum: 10.11.2018

Software-Tester: Matthias Prim

Datum: 10.11.2018

**Testergebnis:** PASS

Testobjekt: 1.3.Zugangsbeschränkung-V-0.2

Detail-Design: DD-1.3.Zugangsbeschränkung-V-0.2

### Code-Review:

Nr.	Beschreibung	Ja/Nein	Anmerkung:
CR-1.3-1	Der Informationsfluss erfolgt von links nach rechts.	JA	
CR-1.3-2	Alle Funktionsblöcke sind sichtbar.	JA	
CR-1.3-3	Die Schnittstellen der Software entsprechen den Vorgaben im Detail-Design (Name/Daten-Typ).	JA	
CR-1.3-4	Am linken oberen Rand des Blockdiagramms befindet sich eine kurze Beschreibung der Software als Kommentar.	JA	
CR-1.3-5	In der Beschreibung ist das Datum der Implementierung angegeben	JA	
CR-1.3-6	In der Beschreibung ist der Autor der Implementierung angegeben	JA	
CR-1.3-7	Am linken oberen Rand des Blockdiagramms befindet sich eine Beschreibung der Teilaufgaben dieser Software.	JA	
CR-1.3-8	Jedem Funktionsblock ist mittels Kommentar einer Teilaufgabe zugeordnet.	JA	
CR-1.3-9	Die im Detail-Design geforderten Elemente sind im Frontpanel vorhanden und eindeutig identifizierbar.	JA	
CR-1.3-10	Die Subsoftwarekomponenten/-einheiten wurden laut Detail-Design bzw. Software-Architektur eingefügt.	JA	

### Funktionstest:

Die Funktion wird überprüft indem die Zugangsbeschränkung ausgeführt wird und die Rückgabewerte korrekt übergeben werden. Weiters wird das Verhalten der Buttons und Anzeigeelemente überprüft.

#### Testumgebung:

- TC-1.3-1
- “~\Daten SW-Test\1.3 Zugangsbeschränkung\Daten\AnwenderDaten.xml”
- Anmerkung: Es kann direkt mit AnwenderDaten.xml gearbeitet werden, ohne zuvor eine Kopie zu erstellen.

#### Ablauf:

1. Laden Sie das VI „TC-1.3-1“
2. Geben Sie im Blockdiagramm im Standard-Pfad folgenden Pfad an:
  - “~\Daten SW-Test\1.3 Zugangsbeschränkung“
3. Starten Sie das VI „TC-1.3-1“

Seite 1 von 2

Abbildung 13: Exemplarisches Testprotokoll Seite 1 (1.3 Zugangsbeschränkung-V-0.2)

Testprotokoll: TP-1.3 Zugangsbeschränkung-V-0.2-1.docx

Nr.	Erwartetes Ereignis	Ja/Nein	Anmerkung
FT-1.3-1	Das VI „1.3 Zugangsbeschränkung“ erscheint mit folgenden Werten der Elemente: <ul style="list-style-type: none"> <li>• <b>Benutzername:</b> <i>leer</i></li> <li>• <b>Passwort:</b> <i>leer</i></li> </ul>	JA	
FT-1.3-2	Der Einloggen Button ist ausgegraut und deaktiviert	JA	
4. Geben Sie folgende Werte ein: <ul style="list-style-type: none"> <li>○ <b>Benutzername:</b> Testperson</li> <li>○ <b>Passwort:</b> Test1</li> </ul>			
FT-1.3-3	Der Einloggen Button ist nicht ausgegraut und anwählbar	JA	
5. Drücken Sie Einloggen			
FT-1.3-4	Es erscheint folgende Meldung: <ul style="list-style-type: none"> <li>• <b>Anwender nicht vorhanden oder Passwort falsch!</b></li> </ul>	JA	
6. Bestätigen Sie die Meldung.			
FT-1.3-5	Die Elemente besitzen folgende Werte: <ul style="list-style-type: none"> <li>• <b>Benutzername:</b> <i>leer</i></li> <li>• <b>Passwort:</b> <i>leer</i></li> </ul>	JA	
7. Geben Sie folgende Werte ein: <ul style="list-style-type: none"> <li>○ <b>Benutzername:</b> Testperson1</li> <li>○ <b>Passwort:</b> Test</li> </ul>			
8. Drücken Sie Eingabe			
FT-1.3-6	Es erscheint folgende Meldung: <ul style="list-style-type: none"> <li>• <b>Anwender nicht vorhanden oder Passwort falsch!</b></li> </ul>	JA	
9. Bestätigen Sie die Meldung.			
FT-1.3-7	Die Elemente besitzen folgende Werte: <ul style="list-style-type: none"> <li>• <b>Benutzername:</b> <i>leer</i></li> <li>• <b>Passwort:</b> <i>leer</i></li> </ul>	JA	
10. Geben Sie folgende Werte ein: <ul style="list-style-type: none"> <li>○ <b>Benutzername:</b> Testperson1</li> <li>○ <b>Passwort:</b> Test1</li> </ul>			
FT-1.3-8	Das VI wird beendet	JA	
FT-1.3-9	Im Frontpanel von „TC-1.3-1“ enthält das Anzeigeelement Aktueller-Anwender folgende Werte: <ul style="list-style-type: none"> <li>• <b>Benutzername:</b> Testperson1</li> <li>• <b>Passwort:</b> Test1</li> <li>• <b>Adminrechte:</b> <i>True</i></li> </ul>	JA	
11. Starten Sie das VI „TC-1.3-1“			
12. Drücken Sie den Button Abbrechen			
FT-1.3-10	Das VI hat sich beendet	JA	
13. Starten Sie das VI „TC-1.3-1“			
14. Drücken Sie Esc			
FT-1.3-11	Das VI hat sich beendet	JA	

**Sonstige Anmerkungen:**

Seite 2 von 2

Abbildung 14: Exemplarisches Testprotokoll Seite 2 (1.3 Zugangsbeschränkung-V-0.2)

### **3.3.6 Prüfung des Software-Systems**

Wie von der Norm gefordert müssen nach der Entwicklung des Software-Systems Prüfungen festgelegt und durchgeführt werden. Die Durchführung der Software-System-Prüfung ist an den vorangegangenen Schritt „Implementieren–Verifizieren-Integrieren“ angelehnt. Der Unterschied besteht darin, dass nun die Anforderungen an das gesamte Software-System durch den Entwickler überprüft werden. Der Test muss so ausgelegt sein, dass die Erfüllung sämtlicher Software-Anforderungen überprüft werden kann. [10]

Besonderes Augenmerk soll an dieser Stelle auch auf die Verifizierung der Bewegungsmuster gelegt werden, da vor allem nicht nur die Software, sondern auch die Hardware des Lokomotionsgeräts geändert wurde.

Das Ergebnis der Verifizierung des Software-Systems befindet sich in Kapitel 4.7.

### **3.3.7 Software-Freigabe**

Bevor die Software eine Freigabe seitens des Herstellers erhält, muss sichergestellt werden, dass die Verifizierung vollständig ist. Restliche Anomalien müssen dokumentiert und hinsichtlich ihres Risikos bewertet sein, um unvertretbare Risiken zu vermeiden.

Freizugebende Software-Versionen sind zu dokumentieren, einschließlich ihrer Erzeugung sowie die Umgebung, in der sie erzeugt wurden. Die Aktivitäten des Software-Entwicklungsplans müssen abgeschlossen sein. Die Software und die zugehörige Dokumentation müssen archiviert werden. Abschließend fordert die Norm die Sicherstellung der zuverlässigen Auslieferung der freigegebenen Software. [10]

Die letzte Forderung der Norm, der zuverlässigen Auslieferung ohne Beschädigung oder nicht autorisierter Änderung, wird dadurch sichergestellt, dass die Software nur in Verbindung mit dem Lokomotionsgerät veröffentlicht wird. Kritische Punkte wie Vervielfältigung, Kennzeichnung bis hin zur Auslieferung entfallen somit gänzlich.

### 3.4 Problemlösungs-Prozess für Software

Der Problemlösungs-Prozess für Software dient der systematischen Dokumentation und Bewältigung von Problemen. Das Vorgehen ist in Abbildung 15 dargestellt.

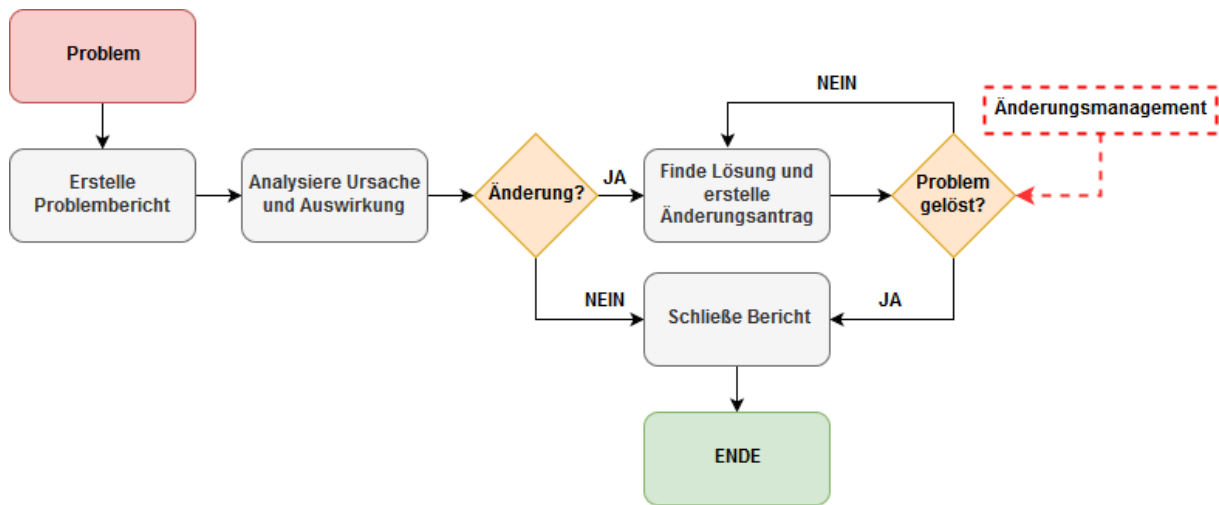


Abbildung 15: Problemlösungs-Prozess für Software

Zu in Kenntnis geratenen Problemstellungen, die vor oder nach der Freigabe auftreten, müssen unabhängig von der identifizierenden Person jeweils Problemberichte erstellt werden. Der Fehler muss analysiert werden, um möglichenfalls die Ursache zu identifizieren. Mit Hilfe des Software-Risikomanagement-Prozesses müssen die Auswirkungen auf die Sicherheit untersucht werden. Problemberichte müssen eine Aussage über die Kritikalität treffen. [10]

Die festgelegten Kritikalitäts-Stufen für den Problemlösungs-Prozess sind in Tabelle 4 zusammengefasst [12]:

Tabelle 4: Kritikalitäts-Stufen des Problemlösungs-Prozesses

Stufe	Beschreibung	Auswirkung
1	Idee	Kein Fehler, sondern Verbesserungsvorschlag für das Produkt
2	Änderungswunsch	Kein Fehler, sondern nur Änderungswunsch
3	Leichter Fehler	Lösung des Fehlers, wenn nötige Ressourcen vorhanden
4	Mittlerer Fehler	Soll im nächsten Release gelöst werden
5	Schwerer Fehler	Muss im nächsten Release gelöst werden

Für das aufgetretene Problem muss eine Änderungsanforderung erstellt werden, um den Fehler zu korrigieren oder alternativ eine schriftliche Begründung abgegeben werden, dass keine Aktion erfolgt. Der Problembericht und die Lösung des Problems einschließlich der Verifizierung der Lösung müssen dokumentiert und archiviert werden. [10]

Ein exemplarischer Problembericht ist in Abbildung 16 zu sehen.



Problembereich: PB\_2.docx

### **Problembereich:**

**Aktueller Status:** Änderungsantrag gestellt

(Bericht erstellt / Ursache identifiziert / Auswirkung analysiert / (kein) Änderungsantrag gestellt)

**Identifikation:** PB-2

**Autor:** Matthias Prim

**Datum:** 11.11.2018

**Kritikalität:** 3 (1-5, Idee, Änderungswunsch, Leichter, Mittlerer, Schwerer Fehler)

**Beschreibung:**

- 1) Die Liste der Patienten soll nach Vorname, Nachname und Geburtsdatum auf- bzw. absteigend sortiert werden können.
- 2) Das VI weist einen Fehler auf, der das Programm bei Doppelklick auf die Spaltenüberschriften bzw. eine Leerzeile zum Absturz bringen kann.
- 3) Die Booleschen Variablen der Buttons könnten unter Verwendung von "latch when released" für mehr Übersicht in die jeweiligen Cases verschoben werden. Das Rücksetzen auf default wäre überflüssig.

**Ursache des Problems:**

**Autor:** Matthias Prim

**Datum:** 11.11.2018

- 1) keine, neue Funktion
- 2) Bei Doppelklick auf die Spaltenüberschriften bzw. eine Leerzeile wird der Wert -1 retourniert. Der Doppelklick-Case wird allerdings verwendet um das Patientenpfad-Array auszulesen. Da das Array mit -1 indiziert wird, kommt es zu einem Error.
- 3) keine

**Auswirkung auf die Sicherheit:**

**Autor:** Matthias Prim

**Datum:** 11.11.2018

- 1) keine
- 2) Programmabsturz
- 3) keine

**Falls Auswirkung auf Sicherheit. Welche Risikokontrollmaßnahmen wurden abgeleitet:**

- keine

**Soll eine Änderung durchgeführt werden:** JA

**Autor:** Matthias Prim

**Datum:** 11.11.2018

a) **Falls keine Änderung durchgeführt wird. Beschreibung der Begründung:**

b) **Falls Änderung durchgeführt werden soll. Verweis auf Änderungsantrag:**

- AEA-2

Seite 1 von 1

Abbildung 16: Exemplarischer Problembereich (2 Patientenverwaltung-V-0.3.vi)

### 3.5 Software-Konfigurationsmanagement-Prozess

Der Software-Konfigurationsmanagement-Prozess dient der Identifizierung und Dokumentation der Entwicklung der Änderungen, die eine Softwareentwicklung durchläuft. Wie bereits vermerkt muss der Software-Entwicklungsplan ein Software-Konfigurations- und Änderungsmanagement, einschließlich SOUP-Konfigurationselementen (SOUP, Software Of Unknown Provenance), enthalten. Dabei versteht man unter einem Konfigurationselement eine Einheit (z.B.: Code, Dokument), die an einem gegebenen Referenzpunkt eindeutig identifiziert werden kann. Der Prozess soll einerseits einen Überblick über das gesamte Software-System ermöglichen und andererseits Änderungen inkl. deren Freigaben leichter nachvollziehbar werden lassen. Die Konfigurationselemente und deren Versionen müssen eindeutig identifizierbar sein und dokumentiert werden. Verwendete SOUP-Konfigurationselemente müssen mit Titel, dem Hersteller und einer eindeutigen Kennzeichnung (z.B.: Versionsnummer, Freigabedatum usw.) ebenfalls dokumentiert werden. [10]

Ein eindeutig festgelegtes Dateibezeichnungs-Schema zur Identifizierung und Versionierung ist im Dokument Konfigurationsmanagement\_Re-Design.docx [19] des digitalen Anhangs enthalten. Die einzige Änderung im Vergleich zum ursprünglichen Namensschema betrifft die Problembenachrichtigungen und Änderungsanträge. Beide Dokumente hatten jeweils nur eine fortlaufende Zahl im Namensschema. Allerdings wurde die Nummer der Änderungsanträge je nach Abarbeitung vergeben. So kam es, dass Problembenachrichtigungen und zusammenhängende Änderungsanträge unterschiedlich nummeriert waren, was eine schwierige Handhabung ergab. Da durch das Re-Design die Probleme und Änderungen einer Software-Komponente ohnehin gesammelt und nicht mehr getrennt durchgeführt und dokumentiert wurden, wurde ein neues Namensschema verwendet. Ein ursprüngliches Beispiel lautet folgendermaßen, Problembenachrichtigung 20 verweist auf Änderungsantrag 9. Ein Beispiel nach dem Re-Design für die Abänderung der Softwarekomponente 3.0.8 lautet hingegen:

- PB\_20.docx → AEA-09.docx
- PB\_3.0.8.docx → AEA-3.0.8.docx

Die Abkürzungen PB und AEA stehen für Problembenachrichtigung und Änderungsantrag.

Änderungen an der Software dürfen nur durch genehmigte Änderungsanforderungen durchgeführt werden. Alle Aktivitäten, die durch eine Änderung beeinflusst werden, vor allem auch die Software-Sicherheitsklassifizierung, müssen wiederholt werden. Getätigte Änderungen müssen verifiziert werden, einschließlich der Verifizierungen, die durch die Änderung ungültig geworden sind. Die Norm fordert die Aufzeichnung und Aufbewahrung

von Dokumenten wie dem Problebericht, der daraus entstehenden Änderungsanforderung und der Genehmigung der Änderung. [10]

Im Zuge des Designs der Software wurde ein Änderungsantragsformular entwickelt. Der Änderungskontroll-Prozess ist in Abbildung 17 ersichtlich. Eine sogenannte Impaktanalyse hilft die Konfigurationselemente, die geändert werden müssen, zu erkennen. Berücksichtigt werden dabei nicht nur die Software, sondern auch die Softwaretests und die vollständige Dokumentation, von den Software-Anforderungen bis hin zu den Detail-Designs. Nachdem die Änderungen durchgeführt wurden müssen jene getestet und verifiziert werden. Wurde das Problem gelöst kann der Problebericht geschlossen werden. Andernfalls muss ein weiterer Änderungsantrag erstellt werden. [12]

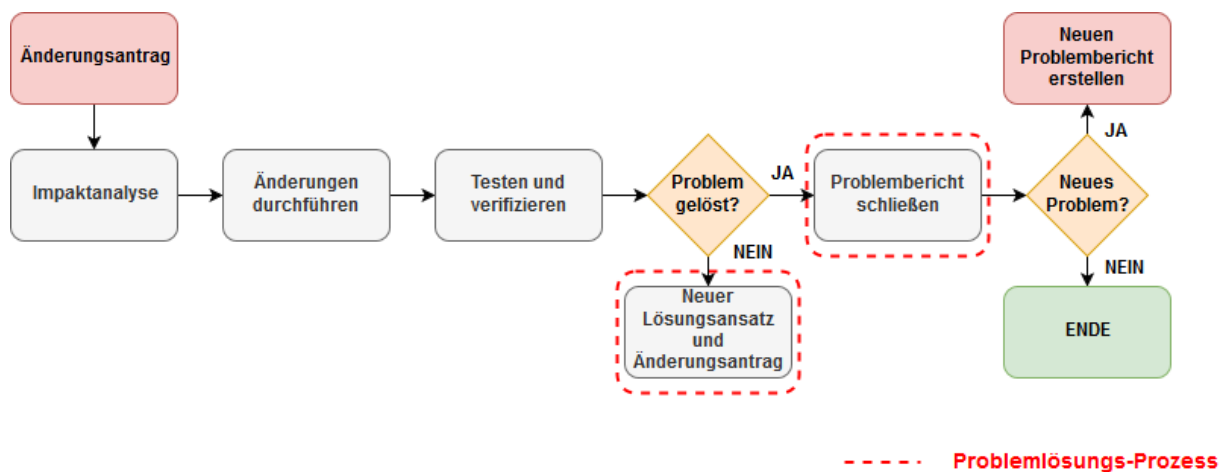


Abbildung 17: Änderungskontroll-Prozess

Der Änderungsantrag von Softwarekomponente „2 Patientenverwaltung“, auf welchen der Problebericht aus dem vorigen Kapitel verweist, ist in Abbildung 18 und Abbildung 19 zu sehen.

Änderungsantrag: AEA-2.docx

## Änderungsantrag:

Autor: Matthias Prim

Datum: 11.11.2018

Verweis auf Problembereich: PB-2

Identifizierung: AEA-2

Kritikalität: 3 (1-5, Idee, Änderungswunsch, Leichter, Mittlerer, Schwere Fehler)

Kurze Beschreibung der Änderung?

- 1) Die Liste der Patienten soll nach Vorname, Nachname und Geburtsdatum auf- bzw. absteigend sortiert werden können.
- 2) Das VI weist einen Fehler auf, der das Programm bei Doppelklick auf die Spaltenüberschriften bzw. eine Leerzeile zum Absturz bringen kann.
- 3) Die Booleschen Variablen der Buttons könnten unter Verwendung von "latch when released" für mehr Übersicht in die jeweiligen Cases verschoben werden. Das Zurücksetzen auf default wäre überflüssig.

### Impaktanalyse:

Autor: Matthias Prim

Datum: 11.11.2018

Software-Anforderung:

Müssen Software-Anforderungen geändert werden? NEIN

Element	Beschreibung:	aktuelle Version	Datum d. Änderung	Autor

### Software- Architektur:

Muss die Software-Architektur geändert werden? NEIN

Element	Beschreibung:	aktuelle Version	Datum d. Änderung	Autor

### Software-Detail-Designs:

Muss das Detail-Design geändert werden? JA

Element	Beschreibung:	aktuelle Version	Datum d. Änderung	Autor
DD-2. Patientenverwaltung-V-0.2.docx	Ergänzung der Beschreibung der Funktionalität und Hinzufügen von neuem SUBVI „0.3 Sort 2D Array“	0.3	11.11.2018	Prim

### Software-Implementierung:

Welche Software muss direkt geändert werden?

Element	Beschreibung:	aktuelle Version	Datum d. Änderung	Autor
2. Patientenverwaltung-V-0.2.vi	Anlegen der Cases Vorname, Nachname und Geburtsdatum um mit dem neuen VI „0.3 Sort 2D Array“ auf- bzw. absteigend sortieren zu können	0.3	11.11.2018	Prim
2. Patientenverwaltung-V-0.2.vi	Case Structure verhindert, dass das Array mit -1 indexiert wird	0.3	11.11.2018	Prim
2. Patientenverwaltung-V-0.2.vi	"latch when released" erspart das zurücksetzen der Variablen; diese wurden in die Cases verschoben	0.3	11.11.2018	Prim

Seite 1 von 2

Abbildung 18: Exemplarischer Änderungsantrag Seite 1 (2 Patientenverwaltung-V-0.3.vi)

Änderungsantrag: AEA-2.docx

Welche Software muss indirekt geändert werden? (Änderung der Konfiguration /SubVIs)

Element	Beschreibung:	aktuelle Version	Datum d. Änderung	Autor

**Software –Test:**

Welche Testprotokolle müssen aktualisiert werden?

Element	aktuelle Version	Datum d. Änderung	Autor
TP-2.Patientenverwaltung-V-0.2.docx	0.3	11.11.2018	Prim

Welche Testcases müssen geändert / erstellt werden?

Element	Beschreibung:	aktuelle Version	Datum d. Änderung	Autor
-	Testen des Sortierens von Vorname, Nachname und Geburtsdatum	0.3	11.11.2018	Prim

Welche Tests müssen durchgeführt /wiederholt werden? Wann wurde der Test durchgeführt?

Testprotokoll	Ergebnis	Datum d. Test	Tester
TP-2.Patientenverwaltung-V-0.3.docx	PASS	11.11.2018	Prim

**Ergebnis:**

Name: Matthias Prim

Datum: 11.11.2018

Wurde das Problem gelöst? JA

Problembereich geschlossen? JA

Wurden zusätzliche Probleme verursacht? NEIN

### 3.6 Ausgangssituation der Arbeit

Die technischen Rahmenbedingungen der Hardware waren durch die Diplomarbeit „Entwicklung eines Lokomotionsgerätes für Kleinkinder“ [17] von A. Tilp, sowie in weiterer Folge durch die Arbeit „Re-Design eines Lokomotionsgerätes für Kleinkinder“ von S. Steinecker [18] vorgegeben. Der Anwendungsbereich wurde durch die Klinik Judendorf-Straßengel auf ein Alter der Patienten von 1 bis 4 Jahren begrenzt. Ebenfalls wurde die minimale Kadenz mit 20 Schritten pro Minute und die maximale Kadenz mit 120 Schritten pro Minute vorgegeben. In Tabelle 5 sind die Daten für den vorhergesehenen Anwendungsbereich, die sich aus dem Alter ableiten, dargestellt.

*Tabelle 5: Vorgesehener Anwendungsbereich des Lokomotionsgeräts (Re-Design)*

	Minimum	Maximum	Einheit
<b>Alter</b>	1	4	Jahre
<b>Gewicht</b>	6,5	16	kg
<b>Körpergröße</b>	67	90	cm
<b>Oberschenkellänge</b>	16	25	cm
<b>Unterschenkellänge mit Fuß</b>	15	27	cm

Das aktuelle Lokomotionstherapiegerät ist auf einem stationären Therapietisch aufgebaut. Die Stützvorrichtung für den Patienten wurde durch eine Aufhängevorrichtung mit einem Gurtsystem umgesetzt. Musste in der letzten Version das Kind noch mittels des Gurtes in einen aufrechten Stand gebracht werden, wurde dieser nun durch einen zusätzlichen Seilzug und einer manuell betriebenen Winde realisiert. Beim Vorgängermodell war es für den Therapeuten teilweise nicht möglich, den Patienten alleine in die gewünschte Position zu bringen, daher stellt diese Weiterentwicklung eine erhebliche Erleichterung dar. Die Beine werden jeweils in einer Ober- und Unterschenkelorthese, welche in der Höhe sowie im Umfang an die Anatomie des Kindes individuell angepasst werden können, fixiert. Das Gerüst der Orthesen wird automatisiert durch vier Motoren angetrieben, um die erforderlichen Bewegungsabläufe der Knie -sowie Hüftgelenke für einen physiologischen Gang zu ermöglichen. In der ursprünglichen Version wurde durch Anbringung von Rollen an den Schuhen die auftretende Reibung des Schuhwerks mit der Bodenoberfläche minimiert. In der darauffolgenden Entwicklung wurde ein passives Laufband integriert, welches die Rollen ersetzte und das Gleiten der Füße am Untergrund gewährleistete. Beim gegenständlichen Re-Design hingegen kommt ein aktiv gesteuertes Laufband zur Anwendung, siehe Kapitel 3.12. Dieses Laufband ist auf einer im Tisch integrierten

Hebebühne eingebettet, um es ideal an die Körpergröße des Kleinkinds anzupassen. Insgesamt ist das Re-Design durch Verstrebungen stabiler gebaut und eine zusätzliche Gewichtsreduktion der bewegten Massen ermöglicht, die Hüft- und Kniemotoren noch effizienter einzusetzen. Während der Therapie kann der Patient seine Hände auf der dafür vorgesehenen Handablage positionieren. Das Re-Design wurde mit einer vergrößerten und stabiler ausgeführten Handablage ausgestattet, welche für den Fixiervorgang einfach mittels Klettverschlusses gelöst werden kann. Die Bedienung erfolgt nun über einen Laptop, welcher in einer frei beweglichen Halterung am Lokomotionsgerät angebracht ist. Der Beistelltisch mit Desktoprechner, Bildschirm und Eingabegeräten ist beim neuen Modell hinfällig und führt hier zu einer Platzeinsparung. Während die Vorgängerversion mit einem externen Schaltschrank bestückt war, ist beim Re-Design die Elektrik im Gerät verbaut und nur nach Abnahme der Seitenwände zugänglich. In die Front integriert ist eine Schublade, hier kann Zubehör wie die Kurbel für die Seilwinde und Hebebühne oder auch die Dokumentationsunterlagen aufbewahrt werden. Abbildung 20 zeigt das finale Konzept in der Form des neu entstandenen Aufbaus des Lokomotionsgeräts.

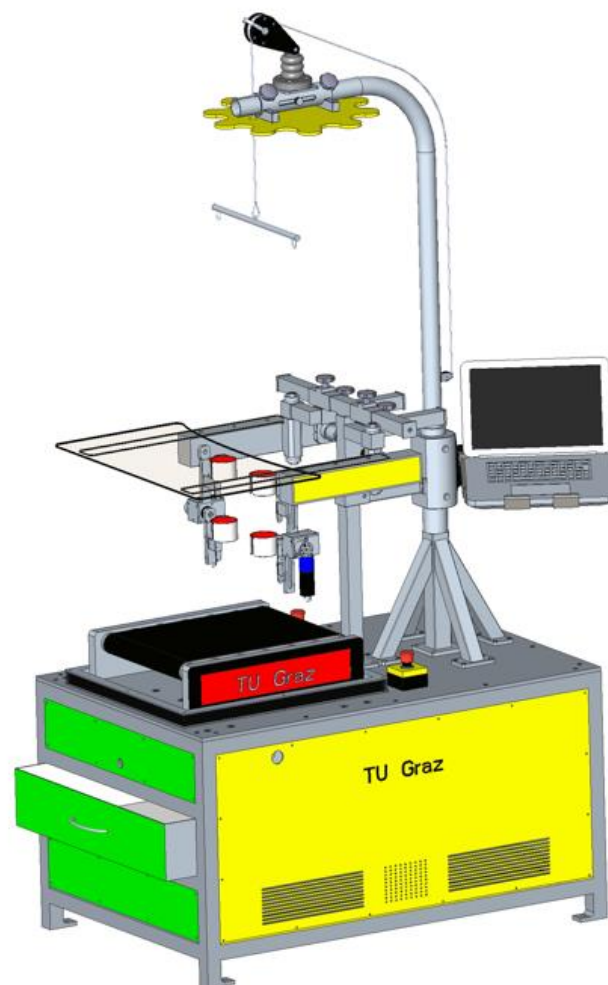


Abbildung 20: Re-Design des Lokomotionstherapiegeräts (3D-Entwurf) [18]

Als Grundlage für das Re-Design der Software wurde die Diplomarbeit „Entwicklung einer Software für ein Lokomotionsgerät“ [12] von M. Kalkgruber, sowie „Beurteilung der Gebrauchstauglichkeit eines Lokomotionsgerätes für Kleinkinder“ [13] von S. Langthaler herangezogen.

Das bereits bestehende Bewegungskonzept soll an dieser Stelle noch einmal kurz erläutert werden. Abbildung 21 zeigt die Ansteuerung einer Orthese des Lokomotionsgeräts.

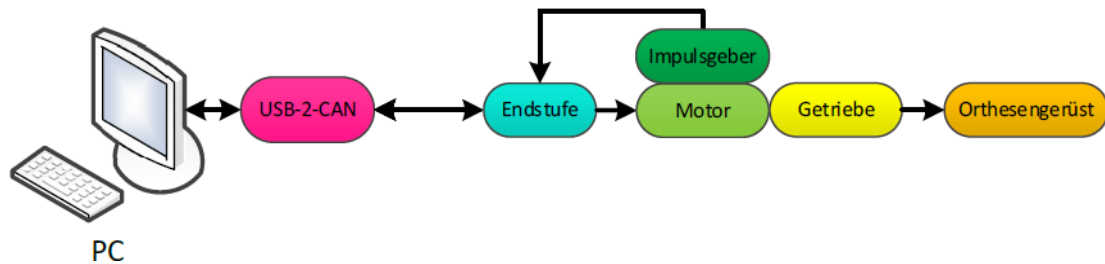


Abbildung 21: Steuerungskonzept der Orthesen vgl. [12]

Zur Durchführung der Bewegung der Orthesen kommen vier permanent erregte Gleichstrommotoren zum Einsatz, welche mittels Getriebe mit dem Orthesengerüst verbunden sind. Die Ansteuerung der Motoren erfolgt durch Endstufen (Motion Controller) in Kombination mit optischen Impulsgebern, um die jeweilige Position zu erfassen. Die Kommunikation zwischen dem Laptop und den Reglern wurde durch einen USB-to-CAN Adapter bewerkstelligt. Zusammengefasst sind die verbauten Komponenten und deren Verwendung in Tabelle 6. Um das benötigte Drehmoment des Knies zu gewährleisten wurde eine Kombination aus Planeten- und Schneckengetriebe verbaut.

Tabelle 6: Komponenten des Antriebssystems

Stk.	Typ	Hersteller	Modellbezeichnung	Anwendung
4	Motion Controller	Faulhaber	MCDC 3006 S CF	Steuerung der Motoren
4	Impulsgeber	Faulhaber	HEDS 5500 A	Winkelbestimmung
2	DC Kleinstmotor	Faulhaber	3863 H 024 CR	Motoren für Hüfte
2	Planetengetriebe	Faulhaber	38A	Getriebe für Hüfte
2	DC Kleinstmotor	Faulhaber	3863 H 024 C	Motoren für Knie
2	Planetengetriebe	Faulhaber	38/2S	Getriebe für Knie
2	Schneckengetriebe	ELRA	SR25	Getriebe für Knie
1	Usb-Can-Konverter	IXXAT	USB 2 CAN V2 Compact V1.1	Verbindung zum CAN-Bus

Weiterführende Informationen der verbauten Antriebskomponenten befinden sich in den entsprechenden Datenblättern. [20] [21] [22] [23] [24] [25] [26] [27]



### 3.7 LabVIEW

Die Entwicklungsumgebung LabVIEW (**L**aboratory **V**irtual **I**nstrumentation **E**ngineering **W**orkbench) und seine zugehörige graphische Programmiersprache „G“ unterscheiden sich maßgeblich von anderen Konzepten, wie z.B. objektorientierten textbasierten Programmiersprachen. Erstellte Programme werden „Virtuelle Instrumente“ (VIs) genannt und unterteilen sich in ein sogenanntes Frontpanel und Blockdiagramm. Die Bezeichnung Frontpanel nimmt dabei Bezug auf die Ähnlichkeit zu Messgeräten. Während der eigentliche Programmiervorgang im Blockdiagramm durch Platzieren von Elementen und anschließendem „verdrahten“ stattfindet, werden am Frontpanel die Anzeigen und Steuerungen für die Bedienung angeordnet. Die Abarbeitung von Programmen erfolgt nach dem Datenfluss-Prinzip, dabei warten Funktionsblöcke solange ab, bis alle Eingangsdaten vorhanden sind, bevor diese ausgeführt werden. Auf diese Weise wird die Reihenfolge der Schritte definiert. [28] [29]

Abbildung 22 zeigt exemplarisch ein Frontpanel und Blockdiagramm des Re-Designs der Lokomotionssoftware.

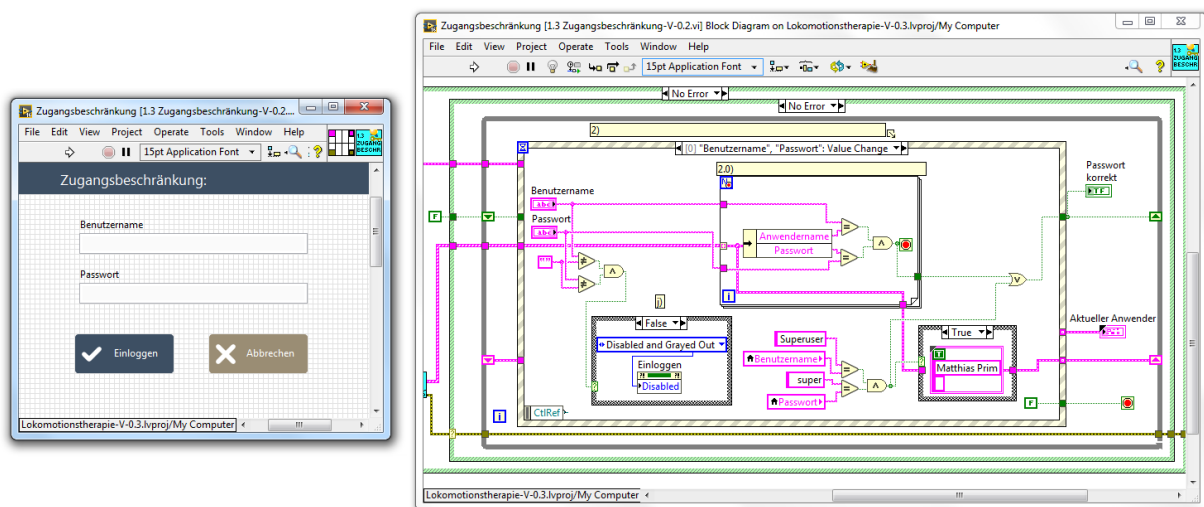


Abbildung 22: Lokomotionstherapie-Software Ausschnitt: Frontpanel rechts, Blockdiagramm links

Die ursprüngliche Lokomotionsgerät-Software wurde in LabVIEW 2012 32 Bit (National Instruments Corp., Austin, USA) programmiert. Aufgrund einer Aktualisierung der Entwicklungssoftware und der ohnehin gegebenen Abwärtskompatibilität wurde in weiterer Folge mit LabVIEW 2016 32-Bit gearbeitet. Das Endgerät (Laptop), auf welchem die Software zum Einsatz kommt, musste dafür vorbereitet werden. Für eine Unterstützung von ausführbaren Dateien, erstellt mit LabView, benötigt es die sog. LabVIEW Run-Time Engine (2016, 32-bit). Zusätzlich musste auf dem Endgerät NI-DAQmx installiert werden, um Datenerfassungsgeräte von NI zu unterstützen, siehe Kapitel 3.9.

## 3.8 Portierung der Lokomotionsgeräte-Software

Bevor mit dem eigentlichen Re-Design der Lokomotionstherapie-Software begonnen werden konnte, musste diese auf die neue Hardware adaptiert werden, da die Software zu Beginn ohne entsprechende Anpassungen nicht lauffähig war. Die Portierung der Lokomotionsgerätesoftware Version 0.2 auf das neu entwickelte und gebaute Lokomotionsgerät wurde in insgesamt fünf Phasen bewerkstelligt, die wie folgt beschrieben werden:

### 3.8.1 USB-to-CAN Konverter

In der ersten Phase musste die grundlegende Kommunikation zwischen dem Entwicklungsgerät und der Lokomotionsgeräte-Hardware hergestellt werden. Der ursprüngliche USB 2 CAN Konverter war am Markt nicht mehr erhältlich und so wurde die nachfolgende Generation verbaut, siehe Abbildung 23.



Abbildung 23: IXXAT USB-to-CAN V2 compact<sup>4</sup>

Die bis dato verwendete Treiber-Version ist mit dem neu verbauten USB 2 CAN V2 Compact Modul des Herstellers Ixxat [27] nicht mehr kompatibel und somit war es notwendig, auf aktuelle Treiber zurückzugreifen. Verwendet wurde Version 3.5.4 der dem Produkt beiliegenden CD.

### 3.8.2 Verifizieren der Verbindung

In der zweiten Phase wurden die Grundfunktionen der verbauten Hardware und korrekte Verbindungen zwischen Motion Controller, Motoren und Inkrementalgeber überprüft. Dazu wurde die offizielle Software des Herstellers namens „Faulhaber Motion Manager“ verwendet. Auch die Grundeinstellungen der Motion Controller wurden mit dieser Software

---

<sup>4</sup> Bildquelle: [https://www.ixxat.com/images/librariesprovider8/product-images/pc-can-interfaces/interfaces-usb-to-can-2-compact-subd.png?sfvrsn=c9ec8d6\\_6](https://www.ixxat.com/images/librariesprovider8/product-images/pc-can-interfaces/interfaces-usb-to-can-2-compact-subd.png?sfvrsn=c9ec8d6_6)

konfiguriert. Nach erfolgreichem Verbindungsaufbau werden in der Herstellersoftware in der linken Spalte die Typenbezeichnungen der verbauten Motoren sowie die verbundenen Motion Controller angezeigt, siehe Abbildung 24. Ein Motor bildet dabei mit einem Controller jeweils einen Knoten (Node 1-4) im CAN-Netzwerk. Um die Funktionsfähigkeit der einzelnen Komponenten zu prüfen, werden die einzelnen Motoren mit einer konstanten Drehzahl angesteuert. Ein Doppelklick im linken Fenster auf einen Knoten aktiviert diesen. Wird nun auf Motorauswahl gedrückt, erscheint eine Abfrage, ob der Motion Controller in den Operationszustand versetzt werden soll. Nach Bestätigung der Meldung wird das erscheinende Fenster wieder geschlossen. Nun können mit der Kommandozeile in der linken oberen Ecke die Motoren angesteuert werden. Zum Ansteuern des Motors mit der Knotennummer 1 mit einer konstanten Drehzahl von 50 U/min müssen folgende Befehle gesendet werden: 1EN, 1V50. Die Orthesenhalterung sollte sich daraufhin in Bewegung setzen. Um den Motor wieder anzuhalten, wird der Befehl 1DI verwendet. Die Abkürzungen der Befehle stehen für Enable, Velocity und Disable. Die Eingaben sind im rechten Fenster der Software ersichtlich. Analog funktioniert das Testen der übrigen Motoren unter vorheriger Aktivierung des jeweiligen Knotens. Aufgrund der unterschiedlichen Übersetzungen von Knien bzw. Hüfte bewegen sich die Halterungen, bei Vorgabe derselben Geschwindigkeit, langsamer oder schneller. Bewegt sich ein Motor nicht, sollten in erster Instanz die Versorgungsleitungen überprüft werden. Vollzieht eine Orthesenhalterung trotz Eingabe derselben Drehzahl eine viel schnellere Bewegung als ihre komplementäre Knie- oder Hüftorthese, liegt eine mögliche Fehlerquelle in der Zuleitung der Inkrementalgeber zum Motion Controller. Ist eine der vier Adern der Signalleitung unterbrochen, ist letztgenanntes Verhalten beobachtbar.

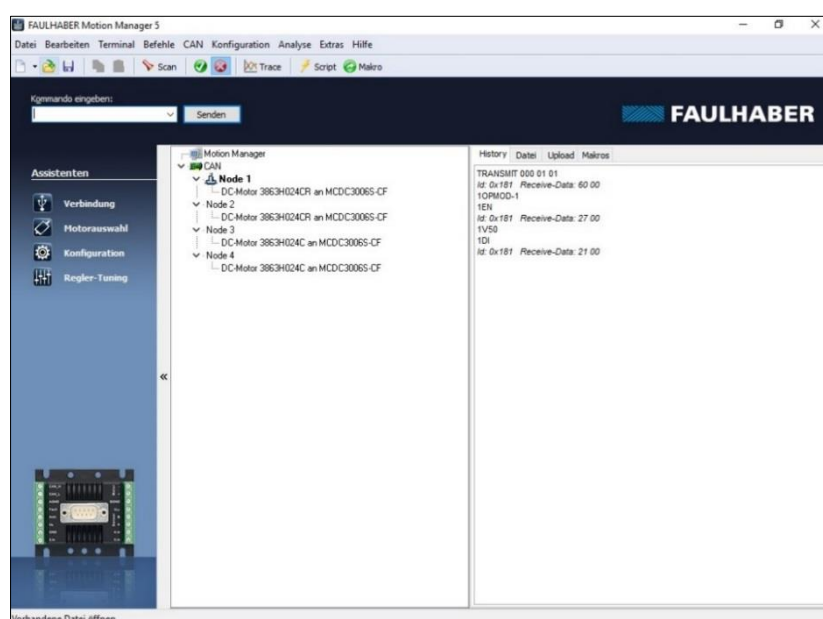


Abbildung 24: Faulhaber Motion Manager (Dr. Fritz Faulhaber GmbH & Co. KG, Deutschland)

### 3.8.3 Initialisierung der USB-CAN-Kommunikation

In der dritten Phase musste ein immer wiederkehrendes Problem mit der Initialisierung der USB-CAN-Kommunikation behoben werden. Im Therapiemodus der Software wird zu Beginn durch Betätigen eines Buttons die USB-CAN Kommunikation initialisiert. Handelte es sich um die erste Initialisierung nach dem Einschalten des Geräts, funktionierte dies anstandslos. Trat allerdings in weiterer Folge ein Fehler auf, wurde das Programm teils wieder in den uninitialisierten Zustand zurückversetzt. Eine erneute Initialisierung war ohne weiteres Zutun unmöglich. Abhilfe schaffte zu diesem Zeitpunkt allein, das Ein- und Ausstecken der USB-Verbindung, bevor im Therapiefenster erneut die Kommunikation initialisiert werden konnte. Bei dem sich bereits im Einsatz befindlichen Lokomotionsgerät wurde immer die USB Schnittstelle programmatisch resettiert, um eine etwaige noch vorhandene Verbindung, welche das Re-Initialisieren verhinderte, zu trennen. Notwendig war dieser Vorgang lediglich, da beim Beenden des Programms auf das Schließen der USB-CAN-Verbindung verzichtet wurde. Das eigentliche Problem resultierte aus der Verwendung der SOUP „devcon.exe“. Dabei handelte es sich um einen Gerätemanager auf Kommandozeilen-Basis, der unter anderem das Zurücksetzen von USB-Verbindungen erlaubt. Der Nachteil dabei ist, dass „devcon.exe“ von der Systemarchitektur (x86 oder x64) abhängt und trotz korrektem System häufig zu Problemen führt. Auf dem Entwicklungssystem sowie dem Endgerät, auf welchen die Lokomotionstherapie-Software zum Einsatz gekommen ist, war „devcon.exe“ nicht funktionstüchtig und der USB-to-CAN-Converter musste regelmäßig an- bzw. abgesteckt werden. Die Lösung des Problems lag im Ersetzen der Struktur durch die vorgesehenen VIs des Herstellers zum Schließen der Verbindung (CANChannelClose, CanControlClose und VCIDeviceClose). Somit konnte gänzlich auf die SOUP verzichtet werden. Erneutes Initialisieren der Kommunikation und auch Beenden und anschließendes Neustarten des Programms sind nun problemlos möglich. Der Vorteil der Elimination der SOUP besteht einerseits in der Unabhängigkeit der Software von der Systemarchitektur und dass ab sofort nicht mehr händisch an- bzw. abgesteckt werden muss. Ein letzter Umstand in welchem die USB-to-Can Schnittstelle händisch resettiert werden muss, ergibt sich, wenn das Programm insgesamt „abstürzt“. Dabei gehen die CAN Handles (Referenzwert zu einer vom Betriebssystem verwalteten Systemressource), welche in einer Variable gespeichert werden, verloren. Um auch diese seltene Möglichkeit auszuschließen und den Initialisierungsvorgang zu automatisieren, kann angedacht werden, die Handles permanent zu jedem Start der Software in einer temporären Datei auf der Festplatte abzulegen.

### 3.8.4 Spitzen- und Dauerstrombegrenzung

In der vierten Phase der Portierung musste eine Problemstellung bezüglich der Konfiguration der Motion Controller zu Programmstart behoben werden. Bevor die eigentliche Therapie startet und sich die Orthesen in Bewegung setzen, werden einige Befehle an die verbauten Faulhaber Motion Controller MCDC 3006 S CF [20] gesendet, siehe Abbildung 25.

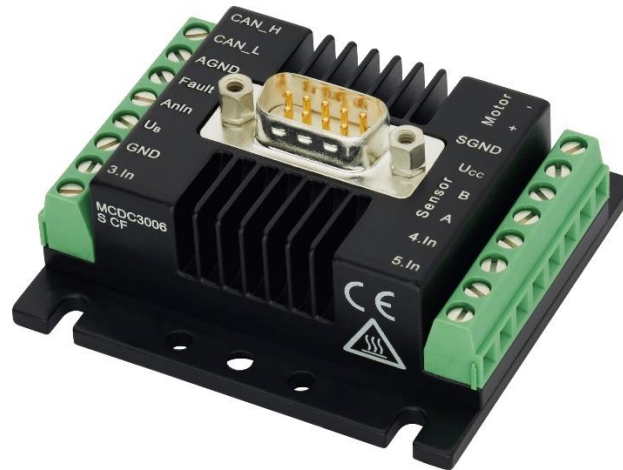


Abbildung 25: Faulhaber Motion Controller MCDC 3006 S CF<sup>5</sup>

Bei jedem Therapiestart wurden die Werte bezüglich der Spitzenstrombegrenzung (Peak Current Limit) und der Dauerstrombegrenzung (Continuous Current Limit) im jeweiligen EEPROM der Motion Controller gespeichert. Dies trat einerseits bei der sog. Referenzierung ein, wodurch die aktuelle Position der Orthesen bestimmt wird, als auch beim eigentlichen Therapievorgang, wo die Grenzen aufgrund der erhöhten Belastung angehoben wurden. Das Problem resultierte aus der Abfrage der Antworten (Replies) auf die übertragenen Befehle, welche nie übereinstimmten und somit als Fehler in der Lokomotionssoftware behandelt wurden. Indem die Lokomotionssoftware bedingt, bis zum Setzen der Ströme, ausgeführt wurde und anschließend die gespeicherten Werte der EEPROMs mit der Herstellersoftware ausgelesen wurden, konnte bewiesen werden, dass das ursächliche Setzen der Stromgrenzen funktioniert hat. Da die Lokomotionssoftware das Setzen der Ströme mit inkorrekten bzw. fehlenden Antworten nicht verifizieren kann, wurde der Dauer- bzw. Spitzenstrom einmalig mit dem Faulhaber Motion Manager in den EEPROMs gesetzt und die jeweiligen Befehle in der Lokomotionsgerätesoftware wurden deaktiviert. Die Deaktivierung des Schreibvorgangs genannter Grenzen durch die Software behob den Fehler. Ebenfalls kommt dies der Lebensdauer der Motion Controller zugute, da EEPROMs nur für begrenzte Schreibzyklen ausgelegt sind. Als nachteilig zu nennen ist, dass keine Differenzierung der Stromgrenzen für die Referenzierung und den Therapievorgang mehr vorgenommen werden

<sup>5</sup> Bildquelle: <https://www.faulhaber.com/de/produkte/serie/mcdc-3006-s/>

kann. Zukünftige reale Einsatzszenarien, die während der Entwicklung nicht gegeben waren, könnten eine Anhebung der Stromgrenzen erforderlich machen. Für die Hüfte wurde eine Dauer- und Spitzenstromgrenze von 3000mA eingestellt, für die Knie 2000mA. Abbildung 26 zeigt stellvertretend die durchgeführten Einstellungen der Hüfte. Durch den Button EEPSAV wird die Konfiguration permanent in den EEPROMs gespeichert.

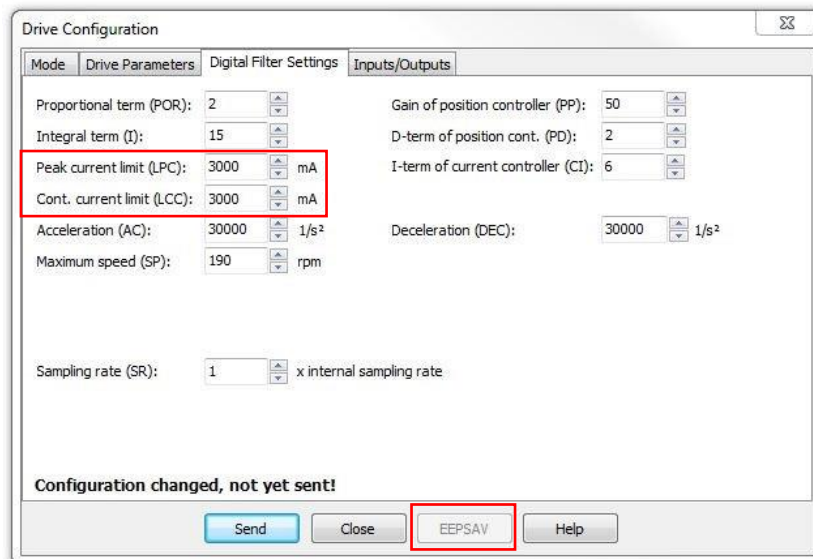


Abbildung 26: Motion Controller Strombegrenzung

Der Grund für das Ausbleiben der Replies konnte, obwohl das neue Lokomotionsgerät abgesehen vom USB-to-CAN Converter baugleich mit dem sich im Einsatz befindlichen ist, bis zum Abschluss der Arbeit nicht eruiert werden. Mit hoher Wahrscheinlichkeit liegt die Ursache nicht am verwendeten USB-to-CAN Converter, da sämtliche Befehle mit Ausnahme des Spitzen- und Dauerstroms und auch die Bewegungsabläufe korrekt funktionieren, siehe Kapitel 4.7.

### 3.8.5 Korrektur der Referenzierungs-Offsets

Die fünfte und letzte Phase umfasste, bevor mit dem eigentlichen Re-Design der Software begonnen werden konnte, die Anpassung der Referenzierungs-Offsets. Vorab der Beschreibung der Problemstellung, wird an dieser Stelle der Referenzierungsvorgang erläutert. Zur Positionsbestimmung kommt der optische Impulsgeber HEDS 5500 A [21] der Firma Faulhaber zum Einsatz. Anders als Endlagenschalter liefern optische Impulsgeber Relativpositionen, deshalb muss vor jeder Verwendung des Gerätes eine Referenzierung aller Antriebsachsen durchgeführt werden. Für die Referenzierung wird ein vorläufiges Koordinatensystem festgelegt. Die Orthesen bewegen sich langsam in Richtung des mechanischen Winkelanschlags, dabei führen die Hüften und Knie eine Flexion durch. Beim Erreichen der mechanischen Beschränkung steigt der Strom in kürzester Zeit sprunghaft an.

Überschreitet der durchschnittliche Motorstrom einen zuvor definierten Schwellwert, wird die jeweilige Position als Endlage detektiert. Der Ursprung des Koordinatensystems wird exakt in jener Position neu gesetzt. Nach dem Erkennen der Endlage werden die Orthesengerüste in die Nulllage bewegt und der Referenzierungsvorgang ist abgeschlossen. Die Nulllage stellt die Ausgangsposition der Orthesen zu Therapiebeginn dar, dabei stehen sie senkrecht zum Boden. Die Referenzierungs-Offset-Winkel sind in der Software vorgegeben und beschreiben die erforderliche Drehung, um von den Anschlägen die Nulllage zu erreichen. Die Grenzwerte der Ströme zur Endlagendetektion wurden auf niedrige Werte eingestellt, um den mechanischen Aufbau des Gerätes zu schonen. Die Stromgrenze zur Detektion der Endlage der Hüfte beträgt 1800 mA und die der Knie 600 mA. Das Hüftgelenk benötigt insofern mehr Strom, da es die Orthesenkonstruktion und die Kniemotoren während des Bewegungsablaufes mit anheben muss. Als redundante Sicherheitsmaßnahme greifen bei einem Softwareausfall die Strombegrenzungen ein, welche in den EEPROMs der Motion Controller gespeichert sind. Die endstufeninterne Begrenzung beträgt 3000 mA für die Hüfte und 2000 mA für die Knie. Wie bereits ausgeführt, entstand das Problem durch die Referenzierungs-Offsets. Wurde das Lokomotionsgerät ohne Korrektur der Winkel in Betrieb genommen, kam es bei Bewegungsausführung dazu, dass die Orthesen in die mechanischen Winkelanschläge fuhren und so eine Überschreitung des Drehmoments als Fehler detektiert wurde. In Folge dessen haben sich alle Antriebe deaktiviert. Die Begründung der Unterscheidung der Referenzierungs-Offsets zwischen der neuen und der vorherigen Generation des Lokomotionsgeräts liegt in den Positionen der Endanschläge. Produktionsbedingt konnten diese nicht im exakt selben Winkel platziert werden. Zusätzlich weisen die Hüftwinkel untereinander eine Abweichung durch den genannten Grund auf. Die Referenzierungs-Offsets des aktuellen Lokomotionsgeräts wurden empirisch für die einzelnen Hüft- und Kniegelenke ermittelt. Zur endgültigen Anpassung war die Generierung eines neuen Ini-Files (Init.xml) notwendig, aus welchem die Software die Referenzierungswinkel auslesen kann. Die eingestellten Werte sind in Tabelle 7 ersichtlich.

Tabelle 7: Referenzierungswinkel der Hüfte und Knie

	Hüfte	Knie
rechts	40°	63°
links	36°	63°

Die Richtungsbezeichnungen beziehen sich auf die Sicht des Patienten. Nach Durchführung aller genannten Änderungen war die ursprüngliche Lokomotionstherapie-Software erstmals uneingeschränkt auf dem neuen Lokomotionstherapiegerät lauffähig. Anschließend konnte mit dem eigentlichen Re-Design begonnen werden.

### 3.9 Hardware

Im folgenden Kapitel wird auf die verwendete Hardware zur Umsetzung der Aufgabenstellung näher eingegangen. Das USB-6001 DAQ-Device dient der Umsetzung der Gewichtsbestimmung, der Not-Aus-Detektion sowie der Ansteuerung des Laufbands. Letztgenanntes wird aufgrund des Umfangs im eigenständigen Kapitel 3.12 beschrieben.

#### 3.9.1 NI USB-6001

Beim USB-6001 von National Instruments handelt es sich um ein Data Acquisition-Device (A/D-Wandler), welches für die Verbindung zwischen der Peripherie und dem Laptop mit der Steuerungssoftware gewählt wurde, siehe Abbildung 27.



Abbildung 27: NI USB-6001<sup>6</sup>

Wie der Name bereits verdeutlicht, läuft der Datenaustausch via USB-Port ab. Das Device bietet sowohl analoge als auch digitale Ein- und Ausgänge. Eine 5V Spannungsquelle, welche maximal 150 mA liefern kann, ist ebenfalls integriert. Alle Anschlüsse sind als Schraubklemmen ausgeführt. Die analogen Eingänge weisen eine Auflösung von 14-Bit auf und mit einem erlaubten Eingangssignal von  $\pm 10V$  ergibt sich für die kleinste noch detektierbare Änderung des Signals ( $\Delta U$ ):

$$\Delta U = \frac{10V - (-10V)}{2^{14}} = \frac{20V}{16384} \approx 1,2mV \quad (1)$$

---

<sup>6</sup> Bildquelle: <https://www.conrad.de/de/national-instruments-ni-daqmx-datenerfassung-einzellizenz-1-schueler-1152109.html>



Die absolute Genauigkeit über die gesamte Skala und den gänzlich zulässigen Temperaturbereich gibt der Hersteller mit 26mV an. Hierbei handelt es sich jedoch um ein Worst-Case-Szenario, welches nicht den Regelfall darstellt. Der Hersteller berücksichtigt bei genanntem Wert mögliche Quantisierungs-Fehler, Offset-Fehler, Nicht-Linearitäts-Fehler, Verstärkungs-Fehler und ein Rauschen des Systems. [30]

### 3.9.2 Gewichtsbestimmung

Zur Gewichtsbestimmung<sup>7</sup> des Patienten kommen sog. Wägezellen (engl.: load cells) zum Einsatz, welche so auch in herkömmlichen Personenwaagen verwendet werden. Charakteristisch ist, dass eine einzelne Wägezelle dabei aus zwei oder vier Dehnmessstreifen (DMS, engl.: strain gauge) besteht, die bereits auf einem Verformungskörper angebracht (verklebt) sind. Ein solcher Folien-DMS besteht aus einer Trägerfolie, in welcher mäanderförmig ein elektrischer Leiter eingearbeitet ist. Wirkt nun eine mechanische Kraft von außen ein, resultiert daraus eine elastische Verformung des Trägermaterials, welche sich durch die Verklebung auf die DMS ausbreitet und somit den Widerstand der DMS verändert. Gleichung (2) beschreibt den Zusammenhang des ohmschen Widerstands eines Leiters in Abhängigkeit von seinen geometrischen Maßen:

$$R = \rho \frac{l}{A} [\Omega] \quad (2)$$

Der ohmsche Widerstand  $R$  ist vom spezifischen Widerstand  $\rho$  (Materialkonstante), der Leiterlänge  $l$  und der Querschnittsfläche  $A$  abhängig. Wird ein DMS gedehnt, so nimmt seine Leiterlänge zu und sein Querschnitt ab. Insgesamt ergibt sich eine Erhöhung des Widerstands. Im Falle einer Stauchung (auch negative Dehnung genannt) nimmt seine Leiterlänge ab und sein Querschnitt zu. Im Gegenteil zur Dehnung wird der Widerstand insgesamt kleiner. Der in diesem Zusammenhang in Datenblättern häufig dokumentierte, einheitslose  $k$ -Faktor steht für die Empfindlichkeit eines DMS. Er beschreibt den Zusammenhang zwischen der relativen Widerstandsänderung und der relativen Längenänderung, auch Dehnung  $\varepsilon$  genannt, siehe Gleichung (3):

$$\frac{\Delta R}{R} = k \frac{\Delta l}{l} = k \cdot \varepsilon \quad (3)$$

---

<sup>7</sup> Anmerkung: Im messtechnischen Sinne handelt es sich um eine Massen- und nicht Gewichtsbestimmung, dennoch wird hier und in weitere Folge der umgangssprachliche Terminus „Patientengewicht“ und nicht „Patientenmasse“, für ein leichteres Verständnis, verwendet.

Um die Bestimmung der Masse auf eine Dehnungsmessung zurückzuführen, kommt eine Messbrücke zum Einsatz. Vier der bisher beschriebenen Wägezellen werden zu einer Wheatstone'schen Brückenschaltung, einer Vollbrücke, verschaltet. Die Vollbrücke bietet die höchste erzielbare Empfindlichkeit mit Brückenschaltungen. Sie ermöglicht, sehr kleine Widerstandsänderungen zu messen, wobei ihr größter Vorteil in der Kompensation von Temperatureinflüssen liegt. Dehnmessstreifen mit gleichsinnigen Widerstandsänderungen werden diagonal zueinander angeordnet, während Dehnmessstreifen mit gegensinnigen Widerstandsänderungen gegenüber bzw. in Serie angeordnet werden. Die Brücke wird im „Ausschlagverfahren“ betrieben. Eine angelegte Versorgungsspannung verursacht eine Diagonalspannung der Brücke, welche von den Widerstandsverhältnissen abhängig ist. Im abgeglichenen Zustand, wenn alle Widerstandswerte gleich groß sind, ergibt sich eine Diagonalspannung von 0 Volt. Kommt es zu einer Widerstandsänderung weicht die Diagonalspannung ab, man spricht von einer Verstimmung der Messbrücke. [31] [32]

Nach anschließender Verstärkung der Spannung mit einer geeigneten Schaltung kann das Signal mit einem Analog-Digital-Converter zur Gewichtsbestimmung verwendet werden.

### **Durchführung:**

Zur Konstruktion der digitalen Waage wurden einer herkömmlichen Personenwaage vier Wägezellen entnommen, siehe Abbildung 28 und in geeigneter Ausführung zu einer Vollbrücke verbunden.

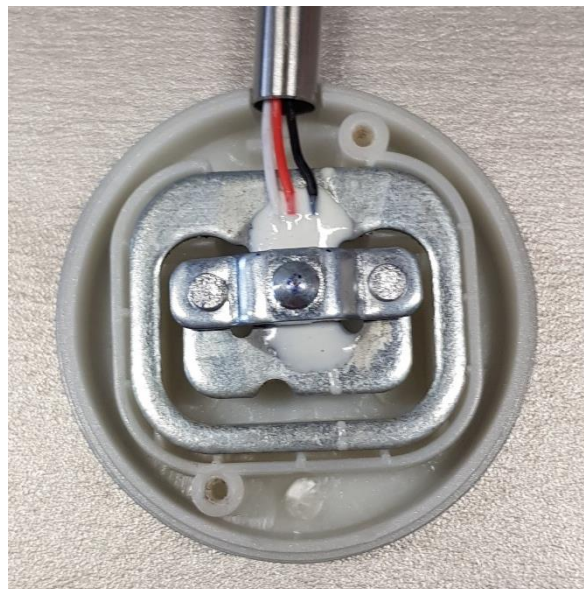


Abbildung 28: Wägezelle

Ein einzelner Dehnmessstreifen weist im unbelasteten Zustand einen elektrischen Widerstand von circa 1000 Ohm auf, wobei auch andere Größen handelsüblich sind. Aufgrund des kleinen Signals im Mikrovolt-Bereich war eine Verstärkung vorab der

Digitalisierung durch das USB-DAQ-Device unerlässlich. Diese erfolgte durch einen Instrumentenverstärker, welcher als diskretes Bauteil erhältlich ist. Eigens für die Versorgung und Auswertung von Brückenschaltungen konzipiert ist das IC von Texas Instruments mit der Bezeichnung INA 125P [33]. Abbildung 29 zeigt die Beschaltung des Bauteils und die Verwendung in Verbindung mit der Vollbrücke.

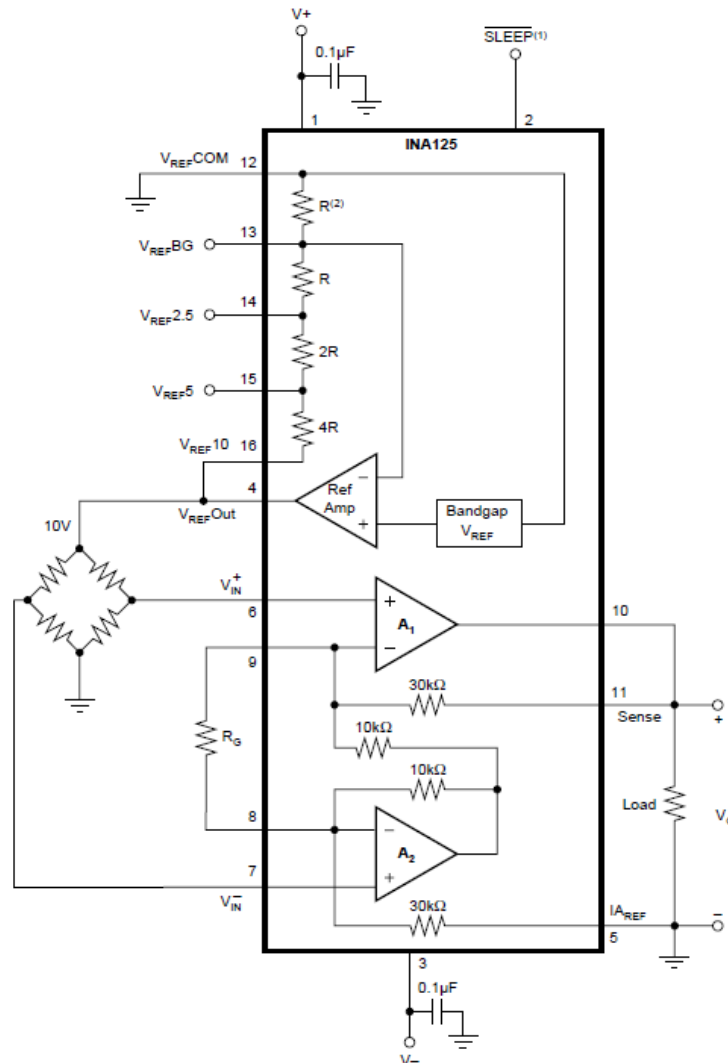


Abbildung 29: Instrumentenverstärker INA 125P mit Vollbrücke [33]

Der gewählte Instrumentenverstärker kann sowohl mit einem Single Supply (2.7V to 36V) als auch mit einem Dual Supply ( $\pm 1.35V$  bis  $\pm 18V$ ) [33] betrieben werden. Da das vorhandene Netzteil des Lokomotionsgeräts eine positive Ausgangsspannung von +24V liefert, war ein Betrieb mit Single Supply vorgegeben. Eine Einschränkung durch die Versorgungsart hätte sich ergeben, wenn negative Spannungen verstärkt werden hätten sollen. Da die Brückenspannung sich allerdings nur im positiven Bereich bewegt, konnte dies vernachlässigt werden. Um eine zuverlässige Verbindung und einfache Einstellung bzw. Wartung zu gewährleisten, wurde eine Platine für den Instrumentenverstärker, siehe

Abbildung 30, entwickelt. Die entworfene Platine diente ausschließlich der Softwareentwicklung. Für einen permanenten Einsatz, muss die Norm EN 60601 -1 Medizinische elektrische Geräte [34] eingehalten werden. Hierzu müsste die Einhaltung der vorgeschriebenen Luft- und Kriechstrecken garantiert werden.

Die zugehörige Konstruktionszeichnung ist im Anhang in Abbildung 63 ersichtlich. Ein Entwurf für ein Printed Circuit Board (PCB) für eine duale Versorgung ist im digitalen Anhang beigefügt.

Bei den Anschlüssen kamen 2-polige Schraubklemmen zum Einsatz. Die Klemme auf der linken Seite dient der Spannungs-Versorgung des ICs. Bei dem Kondensator C1 handelt es sich um einen Entkoppelkondensator, welcher Störungen vorbeugen soll. Auf Empfehlung des Datenblattes [33] wurde dieser mit  $0,1\mu\text{F}$  bemessen. Die 2-polige Klemme mit der Bezeichnung SIG IN, stellt den Eingang für die Brückenspannung (Diagonalspannung) dar.

Der Ausgang, gekennzeichnet durch die 2-polige Klemme mit der Benennung SIG OUT, liefert das verstärkte Signal für das USB-DAQ Device. Die Stiftleiste (Header), auf der rechten Seite des Boards ersichtlich, ermöglicht durch Setzen eines Jumpers die Versorgungsspannung der Brücke, welche vom IC generiert wird, zu wählen. Zur Verfügung stehen dabei 2,5; 5 oder 10 Volt bzw. eine Bandgap-Spannung. Diese Brücken-Versorgungs-Spannung (Bridge excitation voltage) ist an der 2-poligen Klemme mit der Bezeichnung BRIDGE VOLTAGE abgreifbar.

Gewählt wurde unter Berücksichtigung des Sicherheitsaspektes der Wert 5 Volt, da herkömmliche Wägezellen charakteristisch mit Versorgungsspannungen bis zu 10 Volt betrieben werden können, aber kein Datenblatt zu den verwendeten Wägezellen vorliegt.

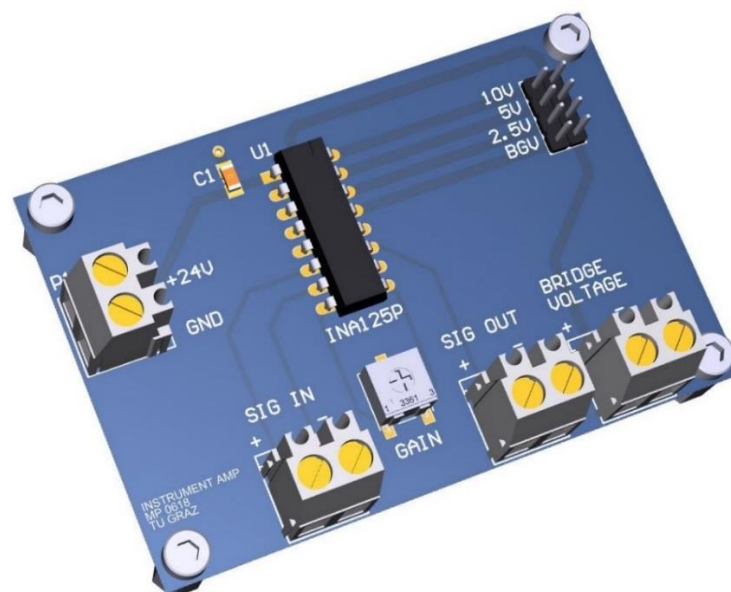


Abbildung 30: 3D Modell Instrumentenverstärker-Board

Das Trimpotentiometer ermöglicht die Einstellung der Verstärkung (Gain). Einmal vor dem Kalibriervorgang gesetzt, darf dieses anschließend nicht mehr variiert werden. Gleichung (4) beschreibt den Zusammenhang der Verstärkung  $G$ , welche sich durch die interne Beschaltung sowie dem externen Widerstand  $R_G$  ergibt [33]:

$$G = 4 + \frac{60k\Omega}{R_G} \quad (4)$$

Um eine möglichst exakte Aufzeichnung des Gewichts zu gewährleisten, sollte der zulässige Eingangsbereich des USB-DAQ-Device ( $\pm 10V$ ) vollständig ausgenutzt werden. Würde das Laufband mit der darunterliegenden Waage mit dem höchst zulässigen Patientengewicht von 16 kg belastet werden und das Potentiometer so lange justiert, bis am Ausgang 10V detektierbar wären, würde das einer exakten Messung am nächsten kommen. Berücksichtigt man jedoch eine mögliche Überlast, muss eine Spannungsreserve eingeplant werden. Ein typischer Fall einer Überlastung kann sich durch das anfängliche Wiegen eines Patienten, ob eine Therapie mit dem Gerät überhaupt durchführbar ist, ergeben. Denkbar wäre auch, dass dritte Personen während des Stillstandes des Laufbands sich auf dieses stützen, und somit die darunterliegenden Wägezellen belasten.

Unter Berücksichtigung der vorangegangenen Diskussion wurde die Verstärkung  $G$  so eingestellt, dass sich bei einer Belastung von 15kg eine verstärkte Spannung von 7,5 Volt am Eingang des DAQ-Devices ergibt.

Zur Erstellung des Teilprogramms, welches im Stande ist, das Gewicht zu messen, musste im Anschluss an das korrekte Einstellen der Verstärkung eine Kalibrierung vorgenommen werden. Dazu wurden Prüfgewichte verwendet, auf der Wagenkonstruktion platziert und die jeweils zugehörigen Spannungen aufgezeichnet. Wichtig dabei war, die Abstände der einzelnen Messpunkte nicht zu groß zu wählen, den gesamten zulässigen Bereich abzudecken und die Spannung der Waage auch im unbelasteten Zustand zu messen. Mittels linearer Regression wurde der mathematische Zusammenhang zwischen dem applizierten Gewicht und der gemessenen Spannung bestimmt. Die sich daraus ergebende Geradengleichung wurde in der Software implementiert, um damit das Gewicht zu ermitteln. Der beschriebene Zusammenhang ist in Abbildung 31 verdeutlicht. Anzumerken ist, dass die Abbildung die Geradengleichung für die finale Konstruktion mit den jeweiligen Prüfgewichten darstellt. Das Laufband wurde über eine Ausgleichsplatte auf den Wägezellen platziert und dadurch, dass die Kalibrierung mit demselben stattgefunden hat, fließt dieses Tara-Gewicht nicht in die Messung ein.

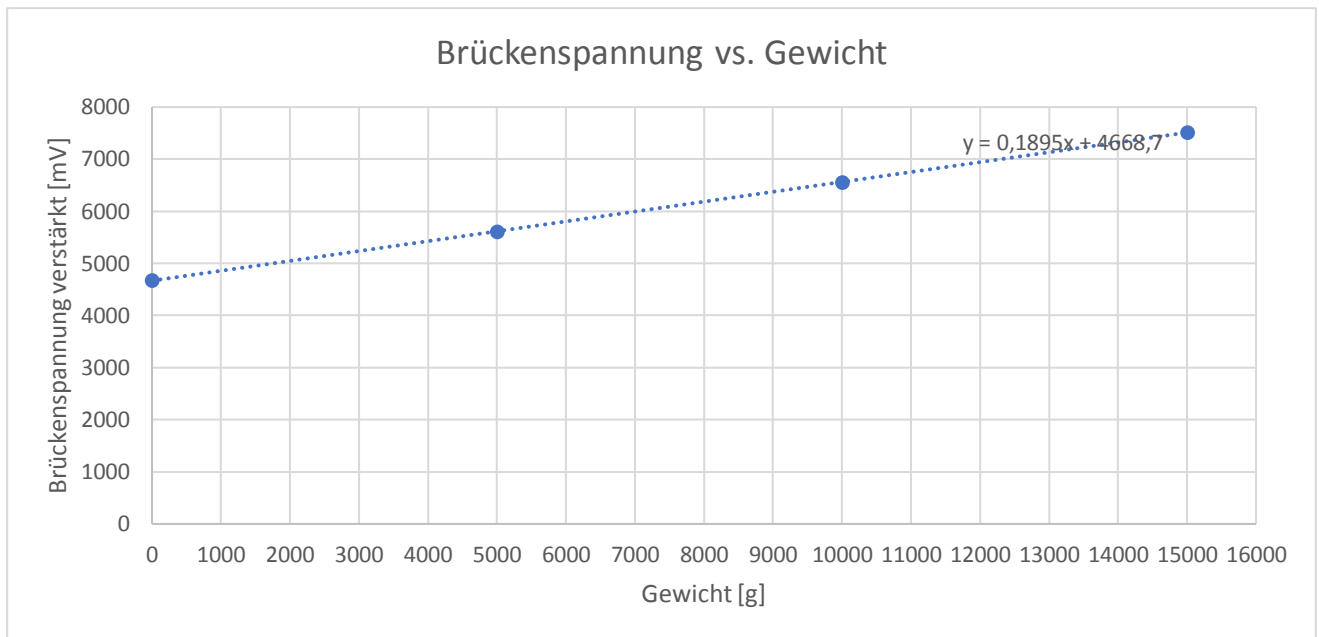


Abbildung 31: Verstärkte Brückenspannung in Abhängigkeit des Gewichts

Die verstärkte Brückenspannung  $U$  wird durch folgende lineare Funktion beschrieben:

$$U[mV] = 0,1895 \cdot \text{Gewicht}[g] + 4668,7[mV] \quad (5)$$

Die gesuchte Relation zur Bestimmung der Gewichtsbelastung ergibt sich durch Umstellung von Formel (5):

$$\text{Gewicht}[g] = \frac{U[mV] - 4668,7[mV]}{0,1895} \quad (6)$$

Unter Betrachtung der maximal zulässigen Eingangsspannung von 10 Volt, ergibt sich für die einkalkulierte Überlast:

$$\text{Gewicht} = \frac{10000 - 4668,7}{0,1895} g \quad (7)$$

$$\text{Gewicht} \approx 28133 g \approx 28 kg \quad (8)$$

Die zulässigen 16 kg des Lokomotionsgeräts dürfen somit um 12 kg überschritten werden. In Prozent ausgedrückt entspricht dies einer 75%igen Überlast, ohne dass ein Schaden eintreten kann. Zusätzlich weist das USB-DAQ-Device im eingeschalteten Zustand einen Überspannungsschutz bis zu 30 Volt auf [30].

### 3.9.3 Detektion des Not-Aus-Schalters

Die aktuelle Version des Lokomotionsgeräts ist mit zwei Not-Aus-Schaltern versehen. Die rot-gelben Druckknöpfe sind jeweils an der linken bzw. rechten Seite des Therapietisches angebracht und gewährleisten somit im Falle des Eintretens einer Gefährdung ein sofortiges Abschalten der Antriebe. Hierzu lösen die Schalter einen Schütz aus, der die Stromzufuhr unmittelbar unterbricht. Ein Schütz ist ein elektromagnetischer Schalter, welcher mittels Magnetfeldes einer Spule Kontakte öffnen bzw. schließen kann. Wird ein Not-Aus betätigt öffnet dies die Kontakte der Phase bzw. des Neutralleiters. Zur Detektion, ob ein Not-Aus-Schalter betätigt wurde, wird ein verbleibender nicht belegter Ein- und Ausgang des Schützes, mit einer Öffner-Funktion verwendet. Zum Einsatz kam ein Schütz des Herstellers Schrack mit der Bezeichnung BZ 326 463, welcher drei Öffner und einen Schließer aufweist [35]. In Abbildung 32 sind der genannte Schütz sowie der zugehörige Schaltplan ersichtlich.



Abbildung 32: Schütz Schrack 25A, 3S+1Ö, 230VAC und Schaltbild [35]

An den Eingang, Klemme Nr. 5, wurden 5 Volt, die vom bereits genannten USB-6001 DAQ-Device bereitgestellt werden, angelegt. Der zugehörige Ausgang, Klemme Nr. 6, wird mit einem analogen Eingang des DAQ-Devices verbunden. Befindet sich das Lokomotionsgerät im Normalbetrieb, sind alle Öffner geschlossen und am jeweiligen Ausgang können in etwa 5 Volt detektiert werden. Durch Drücken eines Not-Aus-Schalters werden die Kontakte geöffnet und die zuvor angelegte Spannung ist nicht mehr detektierbar. Zusammengefasst ist die Logik in Tabelle 8.

Tabelle 8: Not-Aus-Schalter Logiktable

Zustand	Not-Aus-Schalter	Öffner	Detektierbares Signal
1	nicht betätigt	geschlossen	high (5V)
2	betätigt	geöffnet	low (0V)

Die Betriebsspannung des Lokomotionsgeräts in der Höhe von 24 Volt, welche vom Netzteil bereitgestellt wird, darf nicht zur Detektion des Not-Aus Schalters verwendet werden, da der analoge Eingangsbereich des USB-6001 DAQ-Device  $\pm 10$  Volt beträgt und eine Beschädigung in vorhersehbarer Zeit eintreten würde. [30]

Das entwickelte VI misst, ob die Spannung kleiner als 2,5 Volt ist, da dies der halben Versorgungsspannung entspricht. Eine Abfrage ob die Spannung 0 Volt entspräche, wäre niemals zielführend, da bedingt durch äußere Störeinflüsse und den Aufbau des USB-DAQ Devices immer eine elektrische Spannung im Mikrovolt-Bereich zu messen sein wird. Auch der Verlauf der Messleitung gebietet zur Vorsicht. Bei einer Führung knapp entlang der Netzleitung wurden Spannungen größer als die überprüften 2,5 Volt induziert, wodurch hiermit die Not-Aus-Detektion gestört wurde. Durch Verwendung einer abgeschirmten Leitung mit zusätzlichem Mantel konnten die Störeinflüsse behoben werden. Die entsprechende Fehlermeldung, welche bei Betätigung generiert und dem Anwender auf dem Bildschirm angezeigt wird, weist auf das Lösen des Not-Aus-Schalters hin, um den Normalbetrieb wieder aufzunehmen. Anzumerken ist in Hinblick auf die im Zuge der Softwareentwicklung durchgeführten Risikoanalyse, dass die Funktion des VIs rein visuell ist und keinen sicherheitstechnischen Aspekt hat.



### 3.10 Aufmerksamkeitsüberwachung

Da Patienten bei einer Therapie mit dem Lokomotionsgerät nicht unbeaufsichtigt sein sollen, wurde eine Aufmerksamkeitsüberwachung implementiert. Innerhalb definierter Zeitabstände muss der Therapeut seine Anwesenheit via Tastenbetätigung bestätigen, ansonsten werden die Antriebe abgeschaltet und die Therapie gestoppt.

Das bestehende Konzept baut auf einen Countdown, der in Sekundenschritten von einem definierten Startwert abläuft. Die Anwesenheit wird dabei durch Bewegung der Maus bestätigt, wodurch die Zeit auf ihren Startwert resettiert wird. Bei Überschreiten der ersten Zeitschwelle ertönt ein einzelner kurzer Signalton, um den Therapeuten auf die Bestätigung seiner Anwesenheit hinzuweisen. Wird diese bis zum Erreichen der zweiten Schwelle nicht bestätigt, kommt es zu einem hohen kurzen Signalton, der sich im Sekundentakt wiederholt. Nach Ablauf des Timers ohne Bestätigung des Therapeuten, wird ein langer hoher Signalton ausgegeben, die Therapie angehalten und die Antriebe werden abgeschaltet.

Die jeweiligen Werte der Zeitschwellen können in der verschlüsselten Datei „Init.xml“ vorgegeben werden. Folgende Werte sind implementiert:

- Countdown Startzeit: 40 Sekunden
- 1. Zeitschwelle: 10 Sekunden
- 2. Zeitschwelle: 5 Sekunden

Als größter Nachteil der Aufmerksamkeitsüberwachung per Maus kann die händische Bedienung angesehen werden. Spezielle Situationen während der Therapie können den Einsatz beider Hände des Therapeuten erfordern. Um die Aufmerksamkeitsüberwachung trotzdem zu gewährleisten, wurde ein Fußschalter [36], siehe Abbildung 33, zusätzlich zur vorhandenen Bedienung in die Steuerung integriert.



Abbildung 33: USB-Fußschalter, f-pro<sup>8</sup>

---

<sup>8</sup> Bildquelle: <https://www.f-pro.de/wp-content/uploads/AUDIOTR-20180502AAlexander-334-W-GELB-1.jpg>

Die Funktionsweise des USB-Fußschalters liegt in der Simulation der Betätigung spezifischer Tasten der Tastatur begründet. Da keine native Unterstützung für beliebige USB-Geräte von National Instruments gegeben ist und dies für Programmiersprachen auch nicht gängig ist, wird zum Konfigurieren des Fußschalters die Software des Herstellers verwendet. Da durch die Verwendung von SOUP (Software unbekannter Herkunft) auch das Management von Risiken, die aus SOUPs resultieren können durch die Norm EN 62304 vorgeschrieben wird, ist es im Allgemeinen sinnvoll, diese wenn möglich durch einen eigenen Code zu ersetzen. Im vorliegenden Fall wurde vorerst versucht, eine DLL (dynamic link library), welche durch den Hersteller<sup>9</sup> zur Verfügung gestellt wird, zu analysieren bzw. in die Lokomotionsgeräte-Software einzubinden. Die DLL stellte insofern ein Problem dar, da sie in der Programmiersprache C++ geschrieben wurde und somit sämtliche Übergabeparameter und Return-Werte der einzelnen enthaltenen Funktionen bekannt sein müssen und man auf die Dokumentation angewiesen ist. Nach mehreren Fehlversuchen wurde auf die Möglichkeit mit dem herstellereigenen Konfigurationstool gewechselt. Einmalig konfiguriert, kann der somit simulierte Tastendruck programmatisch detektiert werden. Wichtig ist anzumerken, dass es sich hierbei nicht um Verwendung einer SOUP handelt, da die Software-Komponente nicht in die Lokomotionstherapie-Software integriert wird. Somit können sich keine zusätzlichen Risiken aufgrund der SOUP ergeben. Dies war mitunter ein Grund, von der DLL-Implementation Abstand zu nehmen. Die Aufmerksamkeitsüberwachung im Risikomanagement wurde hinsichtlich des Fußschalters aktualisiert, siehe Kapitel 4.2. Würde dieser aufgrund eines Defekts ausfallen, bliebe immer noch die Möglichkeit, die Anwesenheit per Mausklick zu bestätigen. Im Falle einer technischen Störung beider Varianten oder menschlichen Versagens endet die Therapie für den Patienten sicher nach dem voreingestellten Countdown.

---

<sup>9</sup> Eigentlich handelt es sich um eine Co-Produktion zweier Hersteller. Das Konfigurationstool stand auf der ersten Herstellerwebsite zur Verfügung während die DLL sich auf der zweiten Homepage zum Download befand. Zu bemängeln ist, dass nicht ausdrücklich ausgewiesen ist welche Produkte von der DLL unterstützt werden.

### 3.11 Usability und neue Funktionen

Im folgenden Kapitel werden einerseits die Änderungen aufgrund des Usability-Tests, sowie die Neuerungen und Verbesserungen des Re-Designs der Lokomotionstherapie-Software präsentiert. Im Zuge einer vorangegangenen Arbeit wurde die Usability (Gebrauchstauglichkeit) des Lokomotionsgeräts bzw. dessen Software durch S. Langthaler [13] bewertet. Die Aufgabenstellung des Re-Designs der Lokomotionsgerätesoftware umfasst unter anderem die Überarbeitung der bemängelten Punkte des Usability-Tests. Nachfolgend werden ausgewählte Verbesserungen im Detail vorgestellt. Eine Liste mit allen Usability-Schwachpunkten der Software, eine Kurzbeschreibung der jeweiligen Problemlösung und eine Bewertung der durchgeführten Maßnahmen befindet sich im Anhang in Kapitel C.

Sämtliche Frontpanele, die eine User Interaktion benötigen, wurden mit einem neuen Design versehen, welches insbesondere auf eine verbesserte Usability abzielt. Für eine intuitivere Bedienbarkeit ist es fortan möglich, sämtliche Eingaben mit Betätigung der Return-Taste zu bestätigen. Zuvor trat fälschlicherweise bei der Eingabe von Daten ein Zeilenumbruch auf. Der Esc-Taste hingegen wurden allen Abbruch- und Zurück-Buttons zugewiesen. Mit der Tabulator-Taste ist es möglich zwischen den Eingabefeldern zu wechseln, wie es vom Usability-Test gefordert wird. Abbildung 34 stellt das neue und alte Design des Hauptmenüs gegenüber.

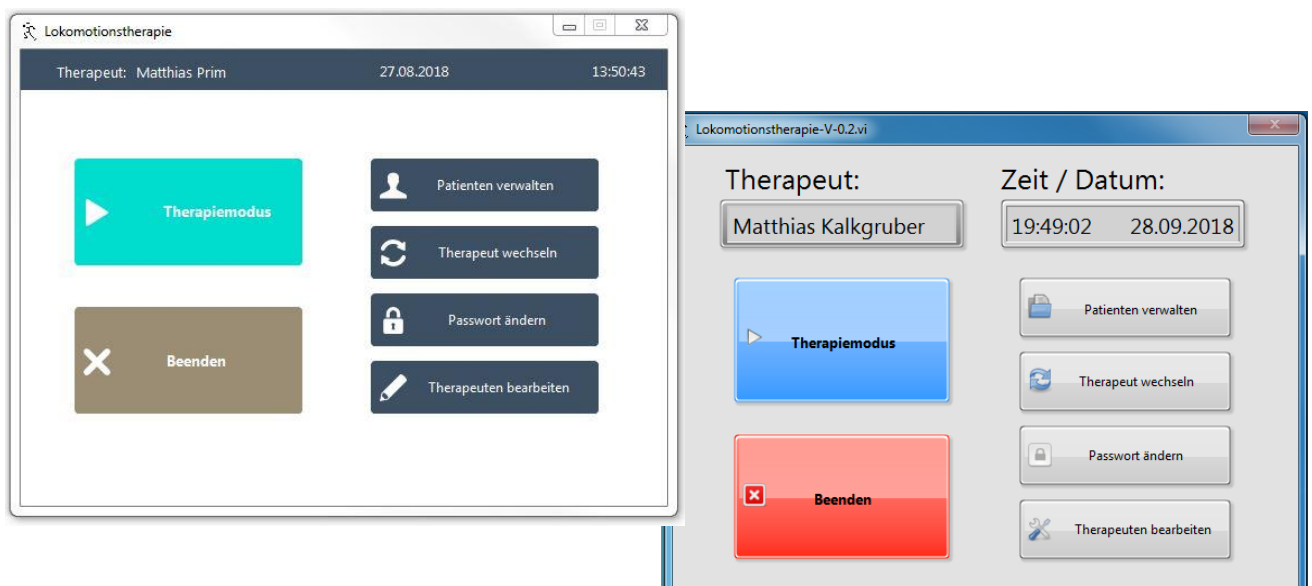


Abbildung 34: Neues und altes Design des Hauptmenüs

Die Anzeige des behandelnden Therapeuten, des aktuellen Datums und der aktuellen Uhrzeit wurde zu einer besseren Übersichtlichkeit in einer neu erstellten Statusleiste am oberen Rand des Hauptmenüs platziert.

### 3.11.1 Anwenderverwaltung

Die Anwenderliste stellte beim Erstaufwurf die Benutzernamen unsortiert dar. Fortan erscheinen die User in alphabetischer Reihenfolge. Zusätzlich ist es nun möglich, auf- bzw. absteigend zu sortieren. Der neue Benutzername-Button wurde mit einem Doppelpfeil ausgestattet, um auf diese Möglichkeit noch deutlicher hinzuweisen, siehe Abbildung 35.

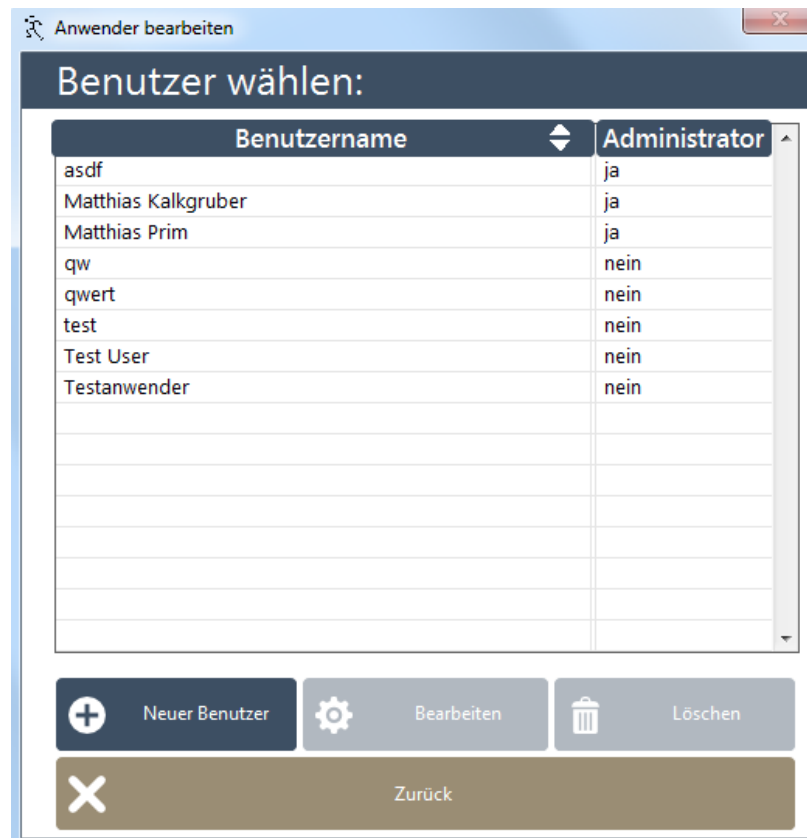


Abbildung 35: Anwenderverwaltung

Eine weitere Modifizierung betrifft die Zugangsbeschränkung. Diese wird einerseits zu Programmstart, um vor unautorisiertem Zugriff zu schützen, als auch zum Wechseln des behandelnden Therapeuten in der Software, verwendet. Bisher endete beim Schließen der Zugangsbeschränkung das Programm. Nunmehr kann das Fenster, wenn fälschlicherweise ein Wechsel des Therapeuten in der Software eingeleitet wurde, problemlos geschlossen werden. Wird das Fenster zu Programmstart geschlossen, endet die Software wie zuvor.

Der aktive User kann sich durch das Re-Design nicht mehr selbst löschen und somit versehentlich ausschließen. Ohne diese Sicherstellung wäre es in letzter Konsequenz sogar möglich gewesen, sämtliche Anwender zu löschen und damit die Unbenutzbarkeit des Programms herbeizuführen.

Möchte der Therapeut sein Passwort ändern, muss er das neue Passwort eingeben und anschließend die Eingabe nochmals verifizieren. Durch einen Programmfehler war es zuvor

möglich, dass trotz nicht identer Eingaben, das Programm die Änderung des Passworts zuließ.

### 3.11.2 Patientenverwaltung und Patientenakte

Die Patientenverwaltung erzeugte in der vorherigen Version eine Liste der Patienten, welche alphabetisch nach Vornamen sortiert wurde. Zudem waren die zusätzlich erfassten Parameter in ihren Spalten nicht sortierbar. Das Re-Design der Software ermöglicht nun, die Listenelemente, Vorname, Nachname und Geburtsdatum auf- bzw. absteigend zu sortieren. Dies vereinfacht den gewünschten Patienten zu lokalisieren, auch eine größere Patientendatenbank ließe sich somit komfortabel verwalten. Abbildung 36 zeigt das neue Design der Patientenverwaltung. Die Doppelpfeile der einzelnen Spaltenüberschriften sollen die neue Sortierfunktion noch besser verdeutlichen.

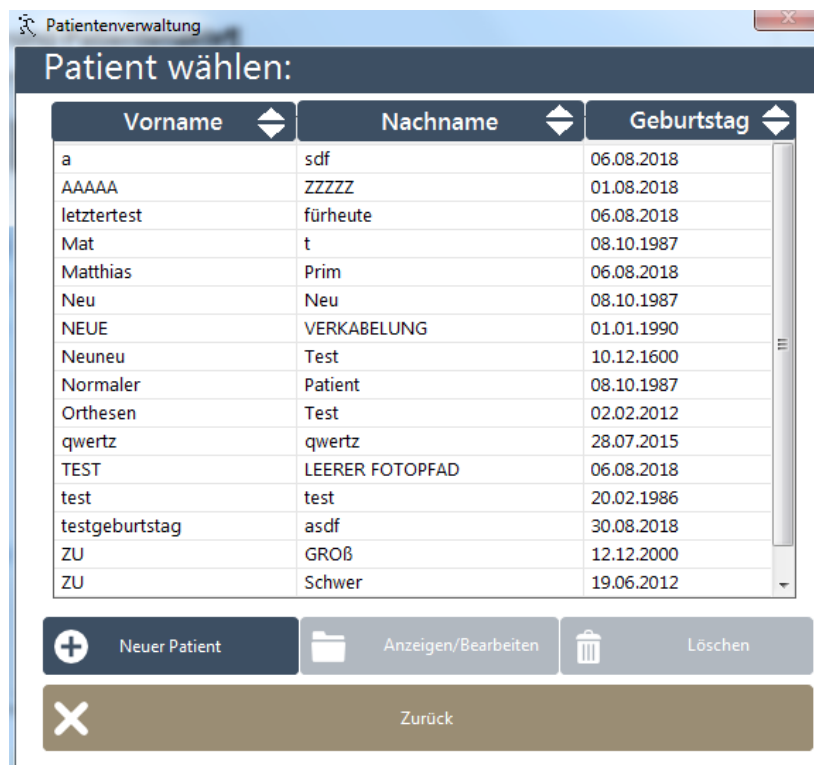


Abbildung 36: Patientenverwaltung

Wird ein neuer Patient angelegt oder die Daten abgeändert, ist es ab sofort möglich, im Zuge dessen das Gewicht zu messen. Der Patient muss sich hierfür auf dem Laufband befinden und der Therapeut den jeweiligen Button anwählen, siehe Abbildung 37. Der Wert des gemessenen Gewichts kann nach Bestätigung der Meldung direkt in die Patientendaten übernommen werden. Eine weitere Neuerung stellt das Patientenfoto dar. Ist kein Patientenfoto in der Datenbank vorhanden, enthält das jeweilige Fenster ein Beispielbild mit dem Text „Kein Foto vorhanden“. Dadurch entfiel die dementsprechende, auf ein Fehlen hinweisende Meldung, die ohnehin bloß quittiert werden konnte. Da die

Lokomotionstherapie-Software nur Patientenfotos im JPEG-Format unterstützt, wird beim Hinzufügen eines Bildes die Dateieindung kontrolliert und bei anderen Dateitypen ein entsprechender Hinweis ausgegeben.

Patientendaten konnten in vielerlei Hinsicht inkonsistent sein. Es war möglich, Patienten ohne Messwert für Körpergröße bzw. Gewicht, also jeweils mit der Zahl 0 versehen, zu erfassen. Weiters konnten das Geburtsdatum in der Zukunft liegen und bereits eingegebene Daten gelöscht werden, sodass die Anlage von unvollständigen Patientenakten möglich war. Durch das Re-Design wurden alle genannten Vorkommnisse behoben.

Abbildung 37: Patientendaten ändern

Wird bei Eingabe der Größe oder des Gewichts das jeweils zulässige Maximum des Lokomotionsgeräts überschritten, enthält die Fehlermeldung nunmehr auch die max. zulässige Größe bzw. das max. zulässige Gewicht, welches vorab durch den zulässigen Anwendungsbereich definiert wurde.

Die durch den Usability-Test geforderte Option zum Ausdrucken der Therapiedaten wurde ebenfalls umgesetzt. Die Patientenakte kann grundsätzlich mittels zwei verschiedener Arten exportiert werden, siehe Abbildung 38. Wird der Button „Patientenakte speichern“ angewählt, erstellt das Programm aus den vorhandenen Daten ein Word Dokument. Ein direkter PDF-Support konnte nicht realisiert werden, da LabView dies nicht unterstützt. Durch Drücken des Buttons „Patientenakte drucken“ erscheint ein neu erstelltes Auswahlfenster der vorhandenen Drucker, womit die Daten exportieren werden.

The screenshot shows a software window titled 'Patientenakte anzeigen bearbeiten' for patient 'Matthias Prim'. The main area is divided into two sections: 'Patientendaten:' and a right-hand sidebar with action buttons.

**Patientendaten:**

- Vorname: Matthias
- Nachname: Prim
- Geburtsdatum: 06/08/2018, Alter in Jahre: 0
- Orthesenlängen:
  - Oberschenkelänge [cm]: links 2, rechts 2
  - Unterschenkelänge [cm]: links 2, rechts 2
  - Höhe Orthesenhalterung [cm]: 2
- Größe in cm: 10, Größe zulässig:
- Gewicht in kg: 10, Gewicht zulässig:
- KEIN FOTO VORHANDEN
- Bemerkung: (empty text area)

**Right-hand sidebar:**

- Ansicht ändern**
  - Therapiedetails anzeigen
- Exportieren der Patientenakte**
  - Patientenakte speichern
  - Patientenakte drucken
- Patient bearbeiten**
  - Patientendaten bearbeiten

At the bottom, there is a close button 'X' and the text 'Ansicht schließen'.

Abbildung 38: Patientenakte

### 3.11.3 Therapiemodus

Zusätzlich zu den bereits beschriebenen Änderungen durch die Portierung der Lokomotionsgeräte-Software, der Aufmerksamkeitsüberwachung und der Not-Aus-Überwachung hat der Therapiemodus noch eine Vielzahl weiterer Verbesserungen erhalten. Es war möglich bei der Therapiedauer den Wert 0 bzw. negative Werte einzustellen, woraufhin sich die Therapie sofort beendete. Die untere Grenze wurde zweckentsprechend mit einer Minute in den Variablenoptionen festgelegt. Die obere Grenze war nicht gesetzt. Es war möglich, eine Therapiedauer über dem globalen Limit, welches vom Programm zu Beginn aus „lnit.xml“ geladen wird, einzustellen. Nachdem das globale Zeitlimit jedoch erreicht war, endete die Therapie. Durch das Re-Design wird die Obergrenze der einstellbaren Therapiedauer nun direkt vom globalen Limit übernommen.

Mit dem neuen Button „Patientendaten anzeigen“ kann zwischen der Ansicht der Patientendaten und dem Schrittdiagramm gewechselt werden, siehe Abbildung 39.

The screenshot displays the therapy interface for Matthias Prim. At the top, it shows 'Therapeut: Matthias Prim', 'Patient: Matthias Prim', and the date/time '26.08.2018 19:48:11'. The interface is divided into several sections:

- Checkliste zum Therapiestart:** A list of six steps with corresponding status buttons:
  - Kommunikation initialisieren (Status: Kommunikation AUS)
  - Referenzieren (Status: keine Referenz)
  - Starte Antriebe (Status: Motoren AUS)
  - Startposition anfahren (Status: Therapiestart nicht möglich)
  - Therapie starten (Status: GESTOPPT)
  - Pause / Beenden
- Einstellungen:** Includes a 'Therapiedauer [min]' set to 20 and a 'Geschwindigkeit [Schritte/min]' slider set to 1.
- Anzeigen:** Features a speedometer for 'Geschwindigkeit [Schritte/min]' (0-120), 'Absolvierte Zeit' (00:00), and 'Durchgeführte Schritte' (0). A patient portrait of Matthias Prim is shown.
- Maximale Kraft [%]:** Four vertical sliders for 'links' and 'rechts' for 'Hüfte' and 'Knie'.
- Graph:** A grid for 'Kraft [%]' and 'Kadenz in Schritte/min' over 'Schritt Nr.' (0-11).
- Buttons:** A red 'NOT-STOPP' button, a 'Patientendaten anzeigen' button, and a 'Therapiemodus verlassen' button.

Abbildung 39: Re-Design des Therapiemodus

Die Einstellung der Schrittgeschwindigkeit (Ziel Kadenz) war vom Aufruf des Therapiemodus an verstellbar und setzte sich zudem nach jeder bestätigten Eingabe der „Checkliste zum Therapiestart“ zurück. Die Einstellung wurde erst unmittelbar vor Therapiestart übernommen. Dies konnte Irritationen beim Anwender hervorrufen, weshalb die Schrittgeschwindigkeit nun erst mit der tatsächlichen Übernahme des Wertes einstellbar ist. Vor dieser Übernahme ist der Wert in der Ansicht ausgegraut und nicht verstellbar. Nach Beenden der Therapie ist die Schrittgeschwindigkeit wiederum ausgegraut, jedoch wird die Einstellung nicht sofort erneut rückgesetzt, sondern kann noch abgelesen werden.

Vom Aufruf des Therapiemodus an war die Anzeige der Schrittgeschwindigkeit (Ist-Kadenz) optisch sichtbar. Anwender versuchten, teilweise mit der Anzeige anstatt der Einstellung die Geschwindigkeit vorzugeben. Auch hier ist die Anzeige nun zu Beginn ausgegraut und erst ab Bewegungsbeginn der Orthesen sichtbar bzw. zum Therapieende wieder ausgegraut.

Der Not-Stop-Button war bereits zu Beginn des Therapiemodus anwählbar. Wurde dieser vor Initialisierung der Kommunikation gedrückt, führte dies unsachgemäß zur Anzeige



„Kommunikation-EIN“. Nun ist der Button erst nachdem die Kommunikation initialisiert wurde anwählbar.

Das Auftreten gewisser Fehler bedingt, dass das Frontpanel des Therapiemodus in den Zustand vor Therapiestart zurückversetzt wird. Während die Orthesen sofort anhielten, stoppte in dieser Situation die absolvierte Therapiedauer nicht. Durch das Re-Design wurde der Fehler behoben.

### Mathscript RT

Für die Bewegungserstellung der Orthesen wurde ein kostenpflichtiges Add-On für LabView mit der Bezeichnung MathScript RT verwendet. Es erweitert LabView um einen nativen Compiler für \*.m-Dateien, die in der Software MATLAB (Mathworks Inc., Natick, USA). oder GNU Octave entwickelt wurden. Zusätzlich ermöglicht es textbasierte Programmierung.

Abbildung 40 zeigt den verwendeten Codeausschnitt. Eingangsvariablen sind links, Ausgangsvariablen auf der rechten Seite ersichtlich.

```

Kadenz_soll 1  if (timer-timer_out)> 150
2      delta=Kadenz_soll-Kadenz_ist;
3      %% maximale Ängerung der Geschwindigkeit =1
4      if delta > 1
5          delta = 1;
6      end
7      if delta < -1
8          delta = -1;
9      end
10     if abs(delta)< 0.5
11         delta = 0;
12     else
13         Kadenz_ist = Kadenz_ist + delta;
14         % Sendeperiode
15         Periode = uint32(floor(50*120/Kadenz_ist));
16     end
17     timer_out = timer;
18 end
19
Kadenz_ist
Periode
timer_out
timer

```

Abbildung 40: Mathscript RT

Die blaue Umrandung der Ein- und Ausgangsvariablen steht für den Integer-Datentyp (ganze Zahl). In Zeile 10 ist ein logischer Fehler ersichtlich. Die Variable „delta“ repräsentiert das Ergebnis der Subtraktion der Ist-Kadenz von der Soll-Kadenz, siehe Zeile 2. Eine Subtraktion eines Integers von einem anderen kann erneut nur einen Integer ergeben. Ein Vergleich ob der Betrag von „delta“ kleiner 0,5 ist, ist somit sinnfrei. Auswirkungen auf den Programmablauf waren keine gegeben, da die if-Abfrage ohnehin nur für „delta“ = 0 richtig wäre und „delta“ erneut 0 zugewiesen worden wäre.

Es war möglich, den eigentlichen Quellcode mit Standard-Funktionen von LabView zu ersetzen. Der sich daraus ergebende Vorteil ist, dass die Installation des Add-Ons, auf dem Endgerät entfällt. Darüber hinaus ist die Lizenzgebühr für das Add-On obsolet.

### 3.11.4 Fehlerbehandlung

Die Fehlerbehandlung wurde im Zuge des Re-Designs ebenfalls erweitert. Ein neuer Error Case wurde hinzugefügt welcher eintritt, wenn kein USB-DAQ Device für die NOT-AUS Detektion bzw. die Gewichtsmessung verbunden ist. Wird der Not-Aus Schalter im Therapiemodus betätigt, erscheint ebenfalls ein neuer Fehler, welcher auf das Lösen des Schalters hinweist, um den Normalbetrieb wieder aufzunehmen. Wie durch den Usability-Test gefordert, wurde eine Hilfestellung bei fehlerhafter Initialisierung der Kommunikation in die Fehlermeldung eingefügt, siehe Abbildung 41.

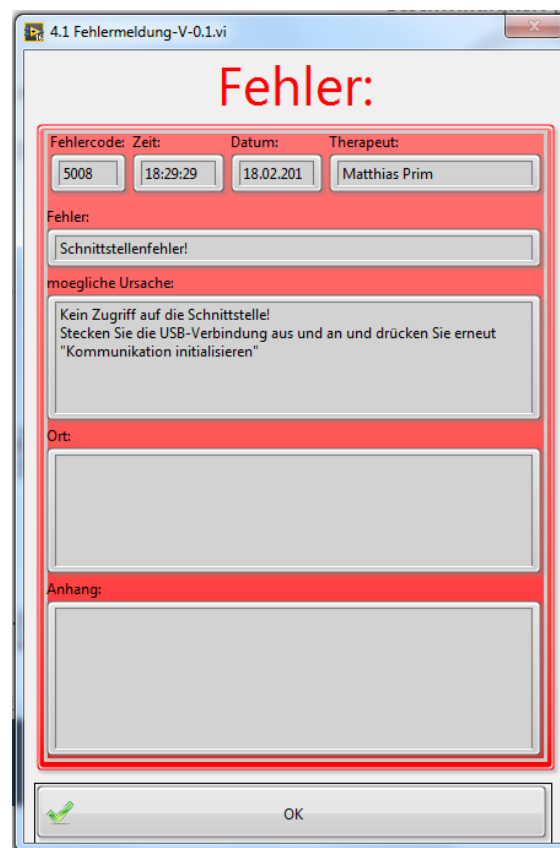


Abbildung 41: Fehlermeldung der Lokomotionsgerätesoftware

### 3.12 Realisierung eines automatisierten Laufbands

Zur Auswahl eines geeigneten Antriebs für ein automatisiertes Laufband musste vorweg die Gehgeschwindigkeit bestimmt werden, da bisher nur eine Kadenz spezifiziert war.

#### 3.12.1 Berechnung der Gehgeschwindigkeit

Für die nachfolgenden Berechnungen wurden die Maße des Re-Designs des Lokomotionsgeräts verwendet, welche sich von der ersten Version unterscheiden. Das Maximum für die Oberschenkellänge hat sich um 3 cm erhöht, während das Minimum für den Unterschenkel mit Fuß um 4 cm verringert wurde. Zusammengefasst sind die differenzierten Längen und die sich daraus ergebenden Gesamtbeinlängen in Tabelle 9 dargestellt.

Tabelle 9: Zulässige Beinlängen des Lokomotionsgeräts (Re-Design)

	Minimum	Maximum	Einheit
<b>Oberschenkellänge</b>	16	25	cm
<b>Unterschenkellänge (mit Fuß)</b>	15	27	cm
<b>Beinlänge</b>	31	52	cm

Berechnet wurde die minimale Beinlänge ( $l_{B,min}$ ) durch Addition der minimalen Oberschenkellänge ( $l_{O,min}$ ) mit der minimalen Unterschenkellänge inklusive Fuß ( $l_{U,min}$ ):

$$l_{B,min} = l_{O,min} + l_{U,min} [m] \quad (9)$$

$$l_{B,min} = 16cm + 15cm \quad (10)$$

$$l_{B,min} = 31 \text{ cm} = 0,31 \text{ m} \quad (11)$$

Die maximale Beinlänge ( $l_{B,max}$ ) wurde durch Addition der maximalen Oberschenkellänge ( $l_{O,max}$ ) mit der maximalen Unterschenkellänge inklusive Fuß ( $l_{U,max}$ ) bestimmt:

$$l_{B,max} = l_{O,max} + l_{U,max} [m] \quad (12)$$

$$l_{B,max} = 25cm + 27cm \quad (13)$$

$$l_{B,max} = 52 \text{ cm} = 0,52 \text{ m} \quad (14)$$

Zudem ist für die Berechnung der Hüftwinkel, den die Orthesenhalterungen zu Therapiestart einnehmen, entscheidend. Dieser konnte dem sog. „Bewegungsverlauf“- Array entnommen werden. Der Flexionswinkel beträgt  $33,4^\circ$ , während sich der Extensionswinkel auf  $8,6^\circ$  beläuft. Der maximale Hüftwinkel summiert sich somit zu  $42^\circ$ , siehe Abbildung 42.

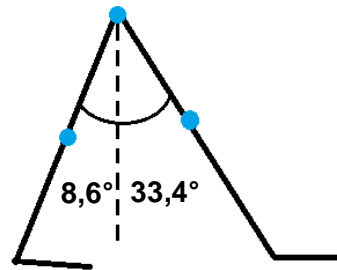


Abbildung 42: Maximaler Hüftwinkel Orthosenhaltungen

### Minimale Gehgeschwindigkeit:

Um die Geschwindigkeit zu berechnen, muss die Kadenz und die Länge eines Einzelschrittes (step length) multipliziert werden. Alternativ kann auch die Länge eines Doppelschrittes (stride length) verwendet werden. Die Abschätzung basiert darauf, dass die Beine mit dem Boden ein gleichschenkliges Dreieck bilden. Betrachtet man den maximalen Hüftwinkel von  $\gamma = 42^\circ$  als eingeschlossenen Winkel, stellt die step length ( $l_s$ ) die Basis des gleichschenkeligen Dreiecks dar, siehe Abbildung 43. Durch halbieren des Dreiecks erhält man zwei idente rechtwinkelige Dreiecke, woraus sich die step length mit einer Winkelfunktion bestimmen lässt.

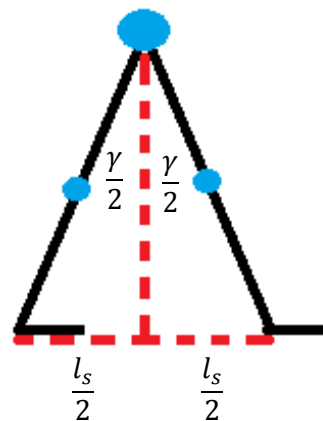


Abbildung 43: Berechnung der Step Length (Schrittlänge)

Wichtig bei der Berechnung der minimalen Gehgeschwindigkeit ist, die kürzeste zulässige Beinlänge zu verwenden. Würde man für die Berechnung eine längere Beinlänge verwenden, würde sich die Schrittlänge erhöhen. Stellt man die Gleichung für den Sinus eines der beiden Dreiecke auf und formt auf die minimale step length ( $l_{s,min}$ ) um, erhält man die gesuchte Größe:

$$\sin\left(\frac{\gamma}{2}\right) = \frac{GK}{HYP} \quad (15)$$

$$\sin\left(\frac{\gamma}{2}\right) = \frac{l_{s,min}}{2} \cdot \frac{1}{l_{B,min}} \quad (16)$$

$$l_{s,min} = 2 \cdot \sin\left(\frac{\gamma}{2}\right) \cdot l_{B,min} [m] \quad (17)$$

$$l_{s,min} = 2 \cdot \sin\left(\frac{42^\circ}{2}\right) \cdot 0,31m \quad (18)$$

$$l_{s,min} \approx 0,222 m \quad (19)$$

Mit der ermittelten minimalen Schrittlänge wird die gesuchte minimale Gehgeschwindigkeit ( $v_{min}$ ) berechnet. Hierfür wird vom Minimum der zulässigen Kadenz (20 Schritte pro Minute) ausgegangen.

$$v = \text{Kadenz} \cdot \text{Step Length} = \frac{\text{Kadenz} \cdot \text{Stride Length}}{2} \left[\frac{m}{s}\right] \quad (20)$$

$$v_{min} \approx 20 \frac{\text{Schritte}}{\text{min}} \cdot 0,222m \quad (21)$$

$$v_{min} \approx 4,44 \frac{m}{\text{min}} \approx 0,074 \frac{m}{s} \quad (22)$$

Anmerkung: Die Vorgehensweise zur Abschätzung der Schrittlänge, wie auch Formel (17) ist in [37] enthalten. Formel (20) ist in abgewandelter Form in [3] enthalten.

### **Maximale Gehgeschwindigkeit:**

Um die maximale step length zu bestimmen, wird von der maximal zulässigen Beinlänge, die zuvor berechnet wurde, ausgegangen. Die Abschätzung basiert darauf, dass die Beine mit dem Boden ein gleichschenkeliges Dreieck bilden. Bei Betrachtung des maximalen Hüftwinkels von  $\gamma = 42^\circ$  als eingeschlossenen Winkel, stellt die step length die Basis des gleichschenkeligen Dreiecks dar. Durch Halbieren des Dreiecks resultieren zwei idente rechtwinkelige Dreiecke, woraus sich die step length bestimmen lässt. Stellt man die Gleichung für den Sinus eines der beiden Dreiecke auf und formt auf die maximale step length ( $l_{s,max}$ ) um, ergibt dies den gesuchten Wert:

$$\sin\left(\frac{\gamma}{2}\right) = \frac{GK}{HYP} \quad (23)$$

$$\sin\left(\frac{\gamma}{2}\right) = \frac{l_{s,max}}{l_{B,max}} \quad (24)$$

$$l_{s,max} = 2 \cdot \sin\left(\frac{\gamma}{2}\right) \cdot l_{B,max} [m] \quad (25)$$

$$l_{s,max} = 2 \cdot \sin\left(\frac{42^\circ}{2}\right) \cdot 0,52m \quad (26)$$

$$l_{s,max} \approx 0,373 m \quad (27)$$

Mit der ermittelten maximalen Schrittlänge wird die gesuchte maximale Gehgeschwindigkeit ( $v_{max}$ ) berechnet. Hierfür wird vom Maximum der zulässigen Kadenz (120 Schritte pro Minute) ausgegangen.

$$v = \text{Kadenz} \cdot \text{Step Length} = \frac{\text{Kadenz} \cdot \text{Stride Length}}{2} \left[\frac{m}{s}\right] \quad (28)$$

$$v_{max} \approx 120 \frac{\text{Schritte}}{\text{min}} \cdot 0,373m \quad (29)$$

$$v_{max} \approx 44,76 \frac{m}{\text{min}} \approx 0,746 \frac{m}{s} \quad (30)$$

Anmerkung: Die Vorgehensweise zur Abschätzung der Schrittlänge, wie auch Formel (25) ist in [37] enthalten. Formel (28) ist in abgewandelter Form in [3] enthalten.

Zusammengefasst sind die Ergebnisse der Berechnungen in Tabelle 10.

Tabelle 10: Berechnete Ergebnisse der Gehgeschwindigkeit

	Minimum	Maximum
<b>Schrittlänge (step length)</b>	0,222 m	0,373 m
<b>Kadenz</b>	20 Schritte/min	120 Schritte/min
<b>Gehgeschwindigkeit</b>	0,074 m/s (4,44 m/min)	0,746 m/s (44,76 m/min)

### 3.12.2 Auswahl des Antriebs

Zur Realisierung eines automatisierten Laufbands musste ein entsprechender Antrieb gewählt werden. Ein Rollenantrieb mit 42 cm Länge, einem Durchmesser von 5 cm, einer Versorgung mit 24 Volt DC und speziell die stufenlose Geschwindigkeitsregelung stellten die Rahmenbedingungen dar. Die Entscheidung fiel zugunsten des Rollerdrives EC310 [38] der Firma Interroll, ersichtlich in Abbildung 44.



Abbildung 44: Interroll Rollerdrive EC310<sup>10</sup>

Der Rollenantrieb wird mit verschiedenen Getriebeuntersetzungen angeboten. Abhängig davon ergibt sich der Geschwindigkeitsbereich, der abgedeckt wird, das Nennmoment, das Anlaufmoment und das Haltemoment. Um eine synchrone Bewegung mit den Orthesen zu ermöglichen, musste die richtige Untersetzung und damit der richtige Geschwindigkeitsbereich gewählt werden. Die minimale und maximale Gehgeschwindigkeit wurden im vorangehenden Kapitel 3.12.1 mit 0,074 m/s respektive 0,746 m/s berechnet. Bei einer gewählten Untersetzung von 16:1 beträgt die minimale Geschwindigkeit des Rollenantriebs 0,05 m/s und die maximale Geschwindigkeit 0,98 m/s. Der benötigte Geschwindigkeitsbereich wird somit vollständig abgedeckt. Die wichtigsten Kennzahlen des Rollerdrive EC310 wurden dem Datenblatt entnommen und sind in Tabelle 11 ersichtlich. [38]

---

<sup>10</sup> Bildquelle: <http://www.hsssearch.co.uk/orgfiles/ZORGF000011/HSS/Images/Interroll%20RollerDrive%20EC310.jpg>

Tabelle 11: Technische Daten Rollerdrive EC310 [38]

KenngroÙe	Wert
Nennspannung	24 VDC
Spannungsbereich	18 bis 28 VDC
Leerlaufstrom	0,4 A
Maximaler Anlaufstrom	5 A
Maximale Anzahl Start/Stopp pro Minute	30
Rollendurchmesser	50 mm
Rollenlänge	500 mm
Schutzklasse	IP66
Getriebeübersetzung	16:1
Geschwindigkeitsbereich	0,05 bis 0,98 m/s
Nennmoment	0,81 Nm
Anlaufmoment	1,95 Nm
Haltemoment	0,64 Nm

Eine Möglichkeit zur Steuerung des Antriebs stellt ein sogenanntes Drivecontrol [39], ebenfalls von Interroll dar. Diese Steuerung weist drei digitale Eingänge zum Einstellen der Geschwindigkeit auf, woraus sich  $2^3 = 8$  diskrete Geschwindigkeitseinstellungen ergeben. Liegt an allen Eingängen ein Low-Potential, entspricht dies einer Geschwindigkeit von 0 m/s, wodurch sich die wählbaren Geschwindigkeiten auf effektiv 7 Einstellungen reduzieren. Dies war mit der Forderung einer stufenlosen Steuerung nicht vereinbar. Aus diesem Grund wurde auf genannten Drivecontrol verzichtet. Die Ansteuerung erfolgt nun über den analogen Geschwindigkeitsanschluss des Rollerdrives EC310. Die Pinbelegung des Motorsteckers des Rollerdrives ist in Abbildung 45 ersichtlich, die zugehörigen Funktionen in Tabelle 12.

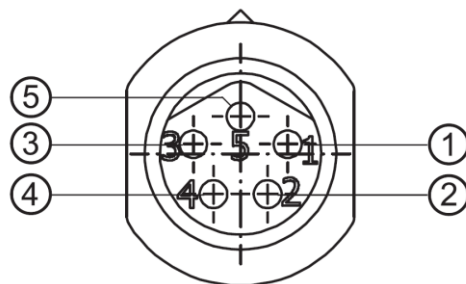


Abbildung 45: Rollerdrive EC310 Pinbelegung [38]



Tabelle 12: Rollerdrive EC310 Motorstecker Pinbeschreibung

Pin Nummer	Farbe	Funktion	Wert
1	braun	Spannungsversorgung (+)	Nennspannung 24 VDC Spannungsbereich: 18 bis 28 VDC
2	weiß	Rotationsrichtung	$U < 4 \text{ V}$ = gegen den Uhrzeigersinn $U > 7 \text{ V}$ = Uhrzeigersinn
3	blau	Masse Spannungsversorgung und Signal (-)	Masse
4	schwarz	Störungsausgang	Fehler: Signal Low Kein Fehler: Signal High
5	grau	Analoges Geschwindigkeitssignal	Stopp: 0 bis 2,3 VDC Steigung der Geschwindigkeit zwischen 2,3 VDC bis 10 VDC: 740 U/min pro Volt Max. Geschwindigkeit 10 VDC bis 24 VDC

Die Sicherstellung der korrekten Polarität war deshalb besonders wichtig, da die Pins 1 und 3 nicht verpolsicher sind, wodurch eine Beschädigung des Motors hätte auftreten können.

Die Berechnung der Geschwindigkeit des Laufbands (Fördergeschwindigkeit) wird wie folgt dargelegt. Ab 2,3 Volt beginnt sich der Motor im inneren der Rollerdrive mit 305 U/min<sup>11</sup> zu drehen. Eine Erhöhung der Spannung um 1 Volt resultiert in einer Erhöhung der inneren Drehzahl um 740 U/min. Ab 10 Volt erreicht die innere Drehzahl ihr Maximum mit 6000 U/min<sup>12</sup>. Um die äußere Drehzahl zu bestimmen, muss die Getriebeuntersetzung eingerechnet werden. Das Über- bzw. Untersetzungsverhältnis ( $i$ ) ist der dimensionslose Quotient der Drehzahlen vor und nach der Umwandlung oder anders ausgedrückt ist es das Verhältnis der inneren Drehzahl ( $n_1$ ) zur äußeren ( $n_2$ ):

<sup>11</sup> Durch Rücksprache mit Interroll wurde das innere Drehzahlminimum, welches im Datenblatt nicht explizit ausgewiesen war, mit 305 U/min in Erfahrung gebracht.

<sup>12</sup> Ebenfalls wurde das innere Drehzahlmaximum, welches im Datenblatt nicht explizit ausgewiesen war, mit 6000 U/min in Erfahrung gebracht.

$$i = \frac{n_1}{n_2} [ ] \quad (31)$$

Um die äußere Drehzahl des Rollerdrives ( $n_2$ ) zu berechnen muss die Motordrehzahl im Inneren ( $n_1$ ) durch die Untersetzung ( $i$ ) dividiert werden:

$$n_2 = \frac{n_1}{i} \left[ \frac{1}{\text{min}} \right] \quad (32)$$

Der für die Berechnung der Fördergeschwindigkeit benötigte Umfang ( $U$ ) errechnet sich aus dem Durchmesser ( $d$ ) und der Kreiszahl ( $\pi$ ):

$$U = d \cdot \pi [m] \quad (33)$$

Die Fördergeschwindigkeit ( $v$ ) ergibt sich aus dem Produkt der äußeren Rollendrehzahl ( $n_2$ ) mit dem Umfang ( $U$ ):

$$v = n_2 \cdot U \left[ \frac{m}{s} \right] \quad (34)$$

Die Berechnung der minimalen und maximalen Fördergeschwindigkeit sowie die Fördergeschwindigkeit in Abhängigkeit der Spannung wird jeweils anhand eines Beispiels im Anhang in Kapitel B demonstriert. Tabelle 13 stellt den Zusammenhang der angelegten Spannung und der resultierenden Geschwindigkeit der Rolle bzw. des Laufbands dar.

Tabelle 13: Rollerdrive EC310 Drehzahl und Geschwindigkeit

Spannung	Umdrehung innen	Umdrehung innen	Umdrehung außen	Umdrehung außen	Geschwindigkeit
V	1/min	1/s	1/min	1/s	m/s
2,3	305	5,08	19,06	0,32	0,05
3,3	1045	17,42	65,31	1,09	0,17
4,3	1785	29,75	111,56	1,86	0,29
5,3	2525	42,08	157,81	2,63	0,41
6,3	3265	54,42	204,06	3,40	0,53
7,3	4005	66,75	250,31	4,17	0,66
8,3	4745	79,08	296,56	4,94	0,78
9,3	5485	91,42	342,81	5,71	0,90
10	6000	100,00	375,00	6,25	0,98

Abbildung 46 verdeutlicht die innere und äußere Drehzahl des Rollerdrive EC310 in Abhängigkeit der angelegten Spannung. Abbildung 47 zeigt die Geschwindigkeit in Abhängigkeit der angelegten Spannung.

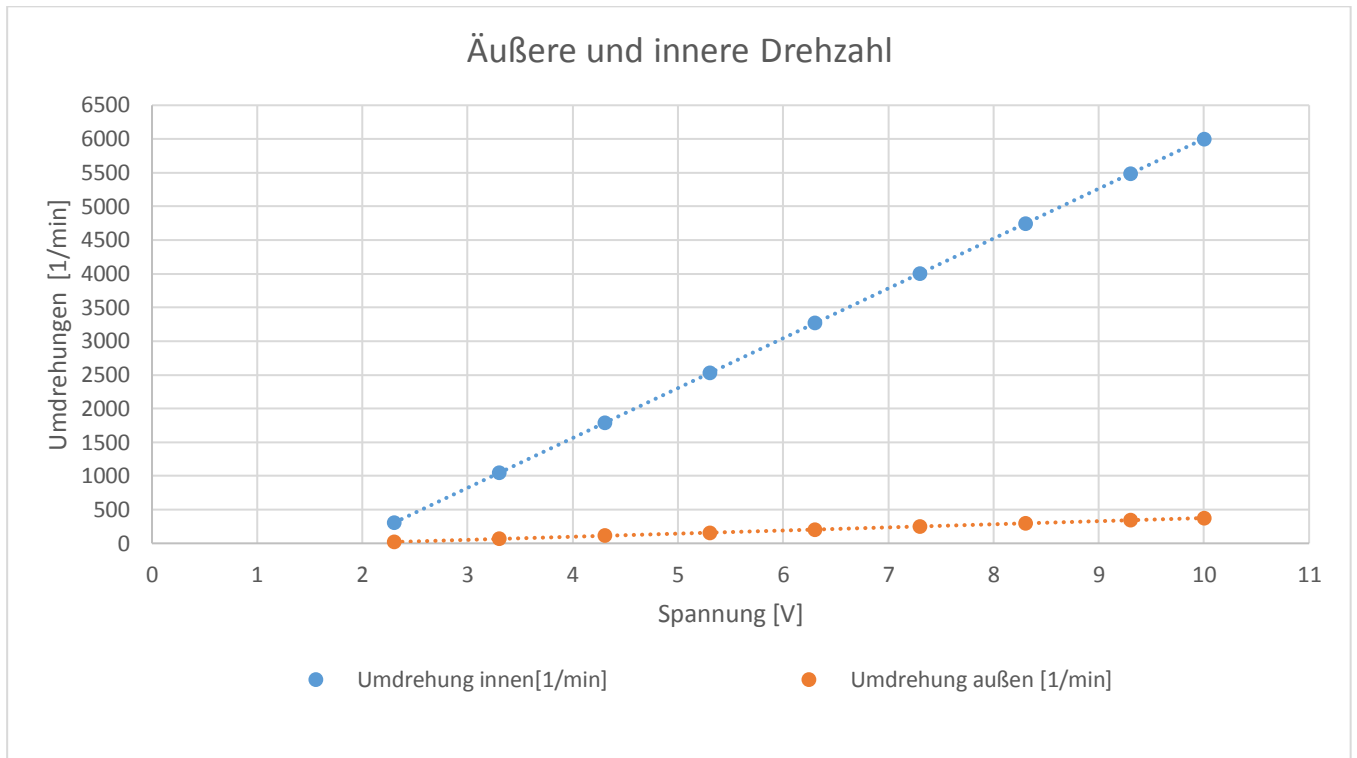


Abbildung 46: Rollerdrive EC310 Drehzahl in Abhängigkeit der Spannung

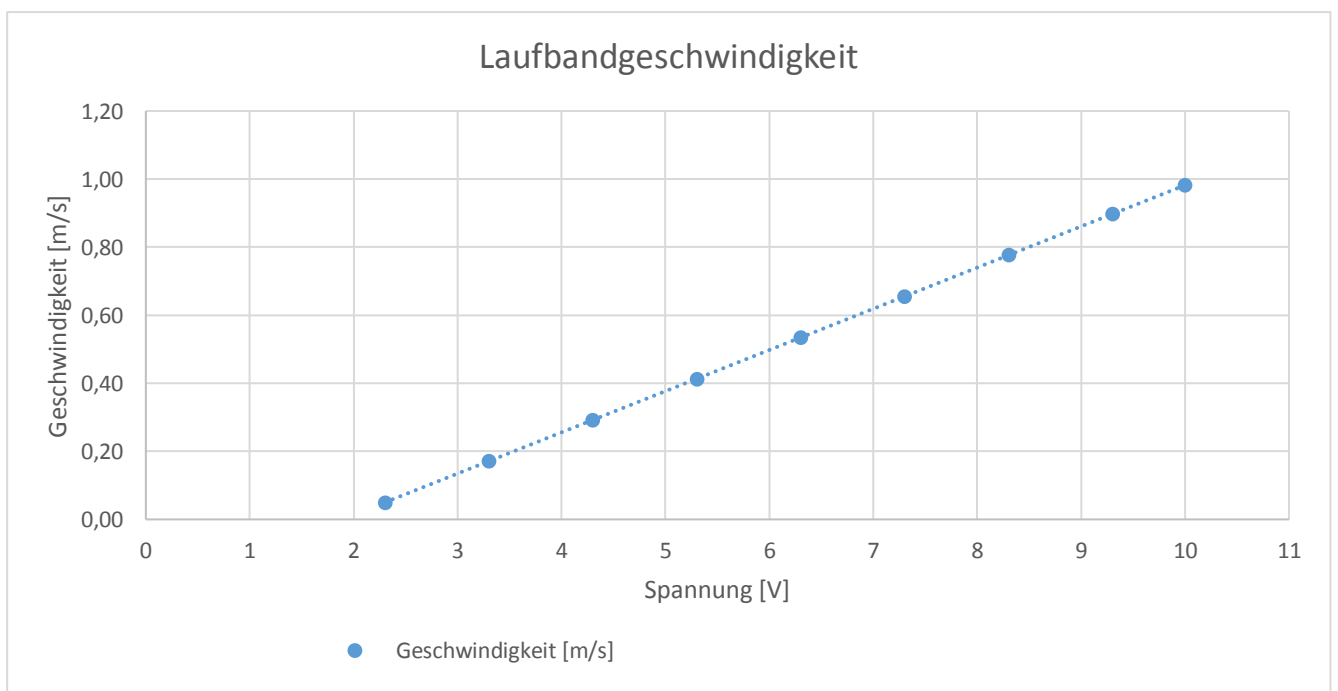


Abbildung 47: Rollerdrive EC310 Geschwindigkeit in Abhängigkeit der Spannung

### 3.12.3 Schaltung und Steuerung mit USB-DAQ-Device

Zur Steuerung des Rollerdrives kommt erneut das bereits vorhandene USB-DAQ-Device 6001 von National Instruments zum Einsatz, siehe Abbildung 48. Die analogen Ausgänge können  $\pm 10$  Volt bei einem maximalen Strom von 5 mA liefern. Der Spannungsbereich für das analoge Geschwindigkeitssignal des Rollerdrives beträgt 0 bis 10 Volt. Das DAQ-Device ist somit ohne vorgelagerte Verstärkerschaltung in der Lage, die Drehzahl des Motors zu steuern. Der analoge Ausgang Nr. 0 (ao0) wird mit Pin 5 des Motors verbunden und dient der Steuerung der Drehzahl. Der verbleibende freie analoge Ausgang Nr. 1 (ao1) des DAQ-Devices kann zur Drehrichtungsvorgabe an Pin 2 des Rollenmotors genutzt werden. Pin 3 des Motors stellt die gemeinsame Masse für die Spannungsversorgung und das Signal dar und wird mit dem gemeinsamen GND-Anschluss der analogen Ausgänge (ao GND) verbunden.

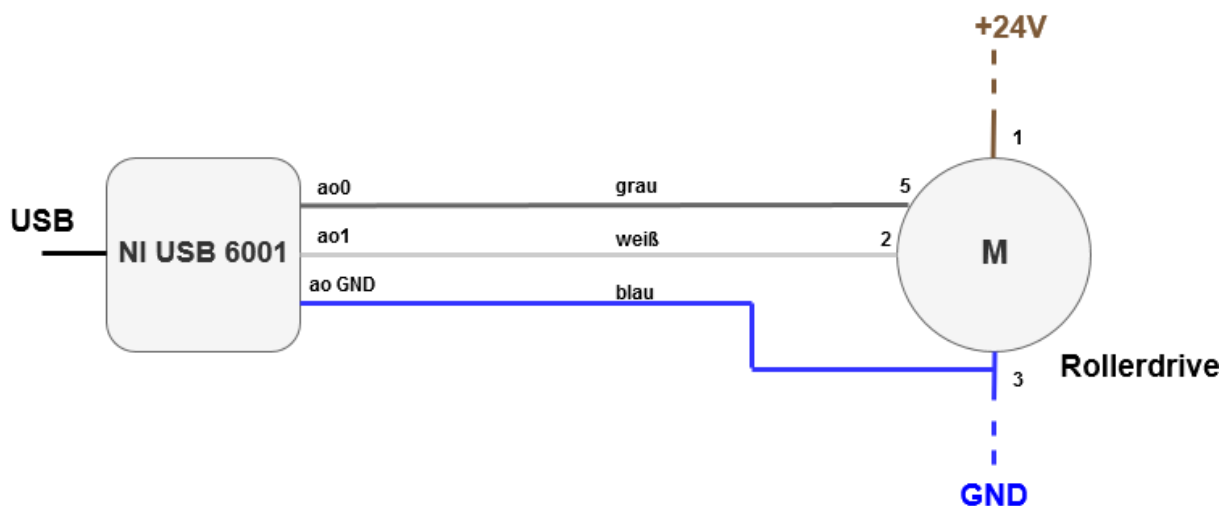


Abbildung 48: Rollerdrive Steuerung mit USB-DAQ-Device

Um das Rollerdrive mit LabView anzusteuern, musste die mathematische Relation zwischen der eingestellten Kadenz (Schritte pro Zeiteinheit) und der zu applizierenden Spannung ermittelt werden. Aus dem gegebenen Zusammenhang (siehe Abbildung 46) lässt sich die innere Drehzahl der Rolle ( $n_1$ ) durch folgende Gleichung beschreiben:

$$n_1 = (U - 2,3) \cdot 740 + 305 \left[ \frac{1}{min} \right] \quad (35)$$

Die äußere Drehzahl ( $n_2$ ) ist um das 16-fache geringer als die innere, aufgrund der gewählten Übersetzung:

$$n_2 = \frac{(U - 2,3) \cdot 740 + 305}{16} \left[ \frac{1}{min} \right] \quad (36)$$

Wird die äußere Drehzahl mit dem Umfang des Rollenantriebs (Durchmesser  $d = 0,05 \text{ m}$ ) multipliziert, ergibt sich die Fördergeschwindigkeit ( $v$ ):

$$v = \frac{(U - 2,3) \cdot 740 + 305}{16} \cdot 0,05 \cdot \pi \left[ \frac{\text{m}}{\text{min}} \right] \quad (37)$$

Umgeformt auf die zu applizierende Spannung ( $U$ ) ergibt sich der gesuchte Zusammenhang mit der Fördergeschwindigkeit [ $\text{m}/\text{min}$ ] zu:

$$U = \frac{16 \cdot v}{0,05 \cdot \pi} - \frac{305}{740} + 2,3 \text{ [V]} \quad (38)$$

Da die Fördergeschwindigkeit des Laufbands der Gehgeschwindigkeit des Patienten entsprechen muss, wird jene mit dem bereits bekannten Zusammenhang mit der Kadenz und der Schrittlänge ( $l_s$ ) substituiert. Der endgültige Ausdruck ist somit von der Schrittlänge [ $\text{m}$ ] und der Kadenz [ $1/\text{min}$ ] abhängig:

$$U = \frac{16 \cdot l_s \cdot \text{Kadenz}}{0,05 \cdot \pi} - \frac{305}{740} + 2,3 \text{ [V]} \quad (39)$$

Eine weitere programmatische Voraussetzung war, dass sich die Geschwindigkeit des Laufbands bei sprunghafter Änderung der Soll-Kadenz nur begrenzt schnell ändern darf. Diese Situation tritt sowohl beim Anfahren als auch Abbremsen und bei Änderung der Geschwindigkeit während der Therapie auf. Implementiert wurde hierfür eine Rampenfunktion, welche die maximale Änderung der Kadenz pro Zeiteinheit limitiert. Die durchgeführte Beschleunigung und Verzögerung musste jener der Orthesen angepasst werden. Käme es zu einer nicht mehr vernachlässigbaren Asynchronität zwischen den sich bewegenden Orthesengerüsten und dem Laufband, könnte dies zu Muskel- oder Knochenverletzungen des Patienten führen. Einen Sonderfall stellt das Anfahren aus dem Stillstand dar. In der Software kann eine Ziel-Kadenz von 20 bis 120 Schritten pro Minute vorgegeben werden. Bei Therapiestart beginnen sich die Orthesen und das Laufband direkt mit einer Startkadenz von 20 Schritten/min zu bewegen. Anschließend inkrementiert die implementierte Rampenfunktion den Wert so lange, bis die eingestellte Ziel-Kadenz erreicht wird. Sollten in zukünftigen Umsetzungen der Lokomotionstherapie-Software Kadenzen kleiner als 20 Schritte/min realisiert werden, muss der limitierende Faktor des Kadenzminimums beachtet werden. Dieses Kadenzminimum ist bedingt durch Formel (39) und der Startspannung des Motors von 2,3 Volt gegeben. Umgeformt auf die Kadenz und für  $U = 2,3 \text{ V}$  substituiert, ergibt sich:

$$Kadenz_{min} = \frac{305 \cdot 0,05 \cdot \pi}{16 \cdot l_s} \left[ \frac{\text{Schritte}}{\text{min}} \right] \quad (40)$$

Grafisch dargestellt ist dieser Zusammenhang in Abbildung 49.

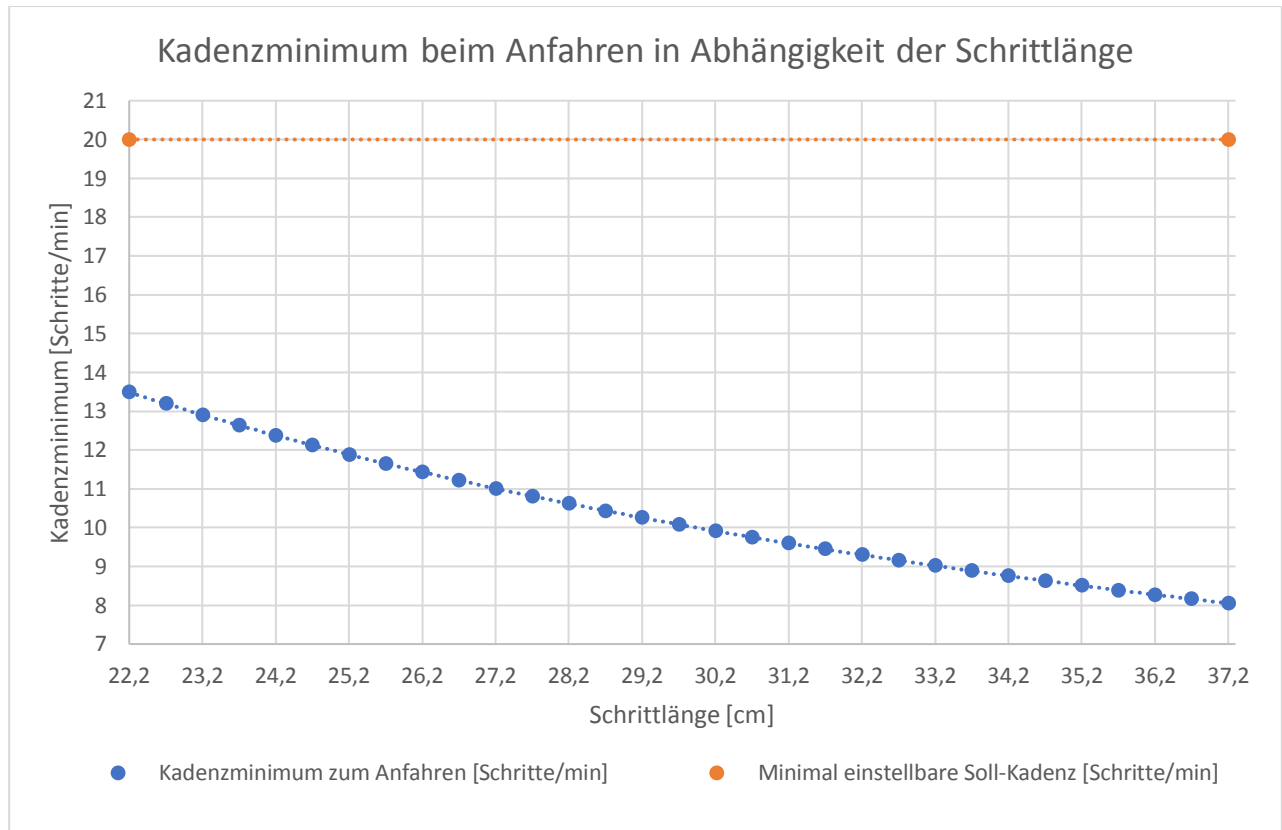


Abbildung 49: Kadenzminimum beim Anfahren in Abhängigkeit der Schrittlänge

Ausgehend von der minimalen Schrittlänge von 22,2 cm verringert sich das Kadenzminimum, welches zum Anfahren des Rollerdrives benötigt wird, bis zur maximalen Schrittlänge von 37,3 cm. Beispielhaft sei das Kadenzminimum für die minimale Schrittlänge betrachtet. Die Rampenfunktion müsste in diesem Fall mit 14 Schritten/min initialisiert werden, um das Laufband zu Therapiestart in Gang zu setzen. Würde die Rampenfunktion mit einer fest vorgegebenen Kadenz, unabhängig von der Schrittlänge und kleiner als das jeweilige Kadenzminimum starten, so würde das Laufband zu Beginn stillstehen. Dies wäre so lange der Fall, bis die Rampenfunktion das jeweilige Kadenzminimum überschreitet.

Abschließend sei noch auf die Problematik des Spannvorgangs des Laufbands hingewiesen. Um einen Geradeauslauf des Laufbands zu garantieren und damit es bei Fußkontakt nicht durchhängt, muss dieses entsprechend gespannt werden. Die angetriebene Rolle sitzt am hinteren Ende, die Spannvorrichtung befindet sich vorne und der Mittelteil wird durch freilaufende Rollen unterstützt. Während sich eine Grund-Zentrierung durch die konischen

Rollen ergibt, muss eine Feineinstellung durch das Spannen auf den jeweiligen Seiten vorgenommen werden. Ein Ungleichgewicht bedingt, dass das Laufband nach links oder rechts abdriftet. Die Stellschrauben mussten justiert werden und nach einer längeren Therapieeinheit als Testphase, korrigiert werden. Dieser Vorgang wurde so lange wiederholt, bis keine sichtbare Abweichung mehr auftrat.

Zusätzlich variiert die Spannung, welche der Motor zum Anfahren benötigt mit der Straffheit des Laufbands. Somit musste ein Kompromiss zwischen festem Laufband und den bereits bekannten 2,3 Volt zum Anfahren eingestellt werden. Da das Rollerdrive nur gesteuert und nicht geregelt werden kann, musste die Geschwindigkeit in Abhängigkeit der Kadenz angepasst werden. Die Abweichung der Geschwindigkeit ist unter anderem darauf zurückzuführen, dass der Motor auch die Umlenkrolle und die Rollen im Zwischenraum mittels des Laufbandes antreiben muss. Ohne durchgeführte Korrekturmaßnahmen trat eine maximale Abweichung der Geschwindigkeit des Laufbands von 24 %, bei einer Kadenz von 20 Schritten pro Minute, auf, siehe Abbildung 50. Positive Werte der Ordinate stehen für ein zu langsames Laufband, während negative Werte ein zu schnelles bedeuten.

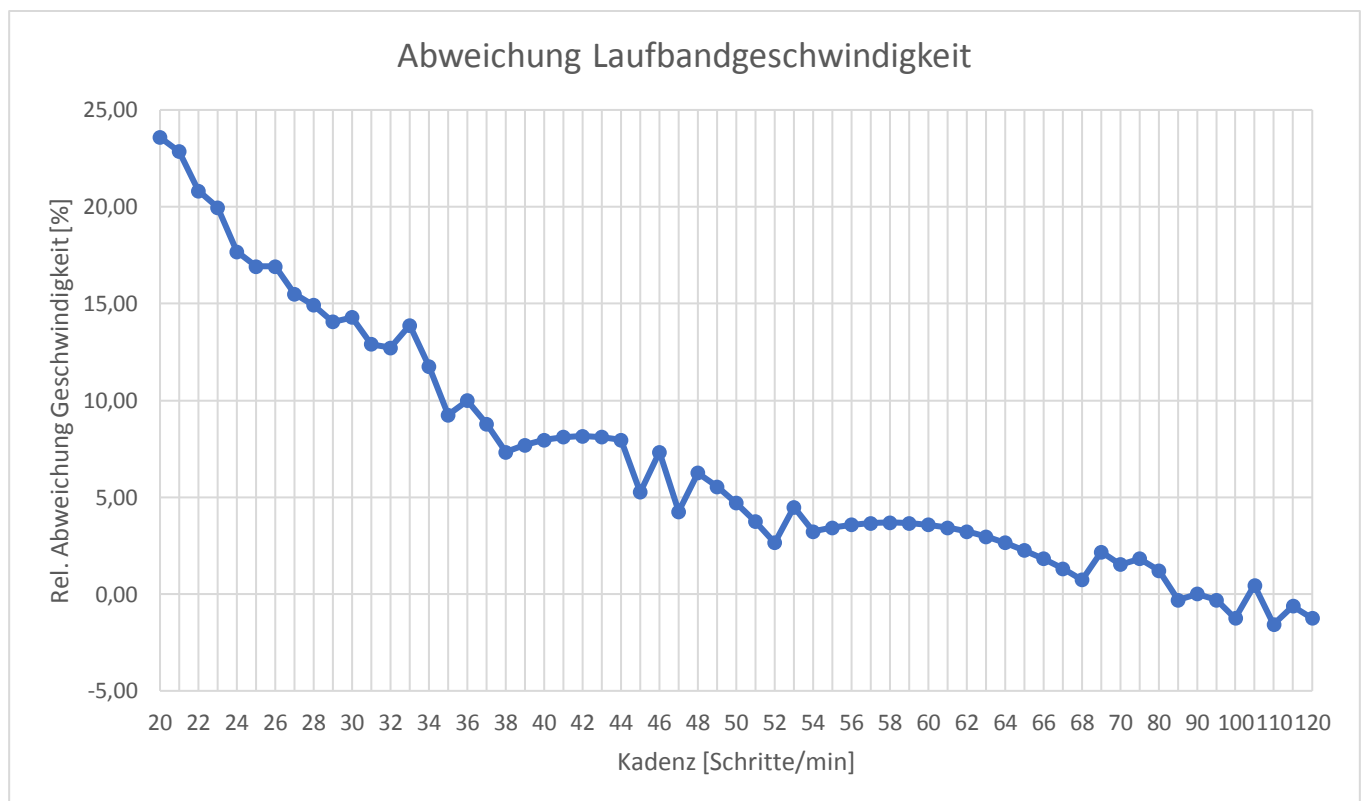


Abbildung 50: Rel. Abweichung der Laufbandgeschwindigkeit in Abhängigkeit der Kadenz

Innerhalb des Bereichs von 20 bis 70 Schritten/min wurde der Fehler für jede einstellbare Kadenz aufgezeichnet, um eine möglichst exakte Kompensation durchführen zu können. Ab 65 bis zum Maximum von 120 Schritten/min ist der Fehler kleiner als 2 %. In diesem Bereich ist die zufällige Messabweichung, bedingt durch das manuelle Zeitstoppen, größer als die zu ermittelnde Messgröße und deshalb nicht mehr aussagekräftig.

Für die Aufnahme der Abweichung wurde das Laufband mit einer optischen Markierung versehen und die Zeit ( $t$ ) für eine festgelegte Rundenzahl gestoppt. Dadurch, dass die Länge des Laufbands ( $s = 1,0071\text{m}$ ) zuvor bestimmt wurde, konnte die tatsächliche Geschwindigkeit ( $v_{ist}$ ) mit folgendem Zusammenhang errechnet werden:

$$v_{ist} = \frac{s}{t} \left[ \frac{m}{min} \right] \quad (41)$$

Die Soll-Geschwindigkeit ( $v_{soll}$ ) ergibt sich aus der Multiplikation der Schrittlänge und der Kadenz:

$$v_{soll} = \text{Kadenz} \cdot \text{Step Length} \left[ \frac{m}{min} \right] \quad (42)$$

Während der Aufnahme muss die Schrittlänge konstant gehalten werden, indem immer derselbe Patient für den Therapiemodus gewählt wird. Die absolute Geschwindigkeits-Abweichung ( $\Delta v_{abs}$ ) ergibt sich aus Subtraktion der Ist-Geschwindigkeit ( $v_{ist}$ ) von der Soll-Geschwindigkeit ( $v_{soll}$ ):

$$\Delta v_{abs} = v_{soll} - v_{ist} \left[ \frac{m}{min} \right] \quad (43)$$

Für die relative Abweichung der Geschwindigkeit in Prozent ( $\Delta v_{rel}$ ) gilt:

$$\Delta v_{rel} = \frac{v_{soll} - v_{ist}}{v_{soll}} \cdot 100 [\%] \quad (44)$$

Zur Korrektur der Geschwindigkeit des Laufbands wurde die relative Abweichung der jeweiligen Kadenz in einem Array gespeichert. Bei der Bewegungsgenerierung werden die Werte in Abhängigkeit der eingestellten Kadenz ausgelesen und anschließend die Geschwindigkeit um den jeweiligen Fehlbetrag erhöht. Nach erfolgter Korrektur variiert die Geschwindigkeit für alle Kadenzen nurmehr im Bereich von  $\pm 2\%$ , siehe 4.7.2 Verifizierung der Geschwindigkeit des Laufbands.



Abbildung 51 zeigt die finale Laufbandkonstruktion in das Re-Design des Lokomotionsgeräts integriert.

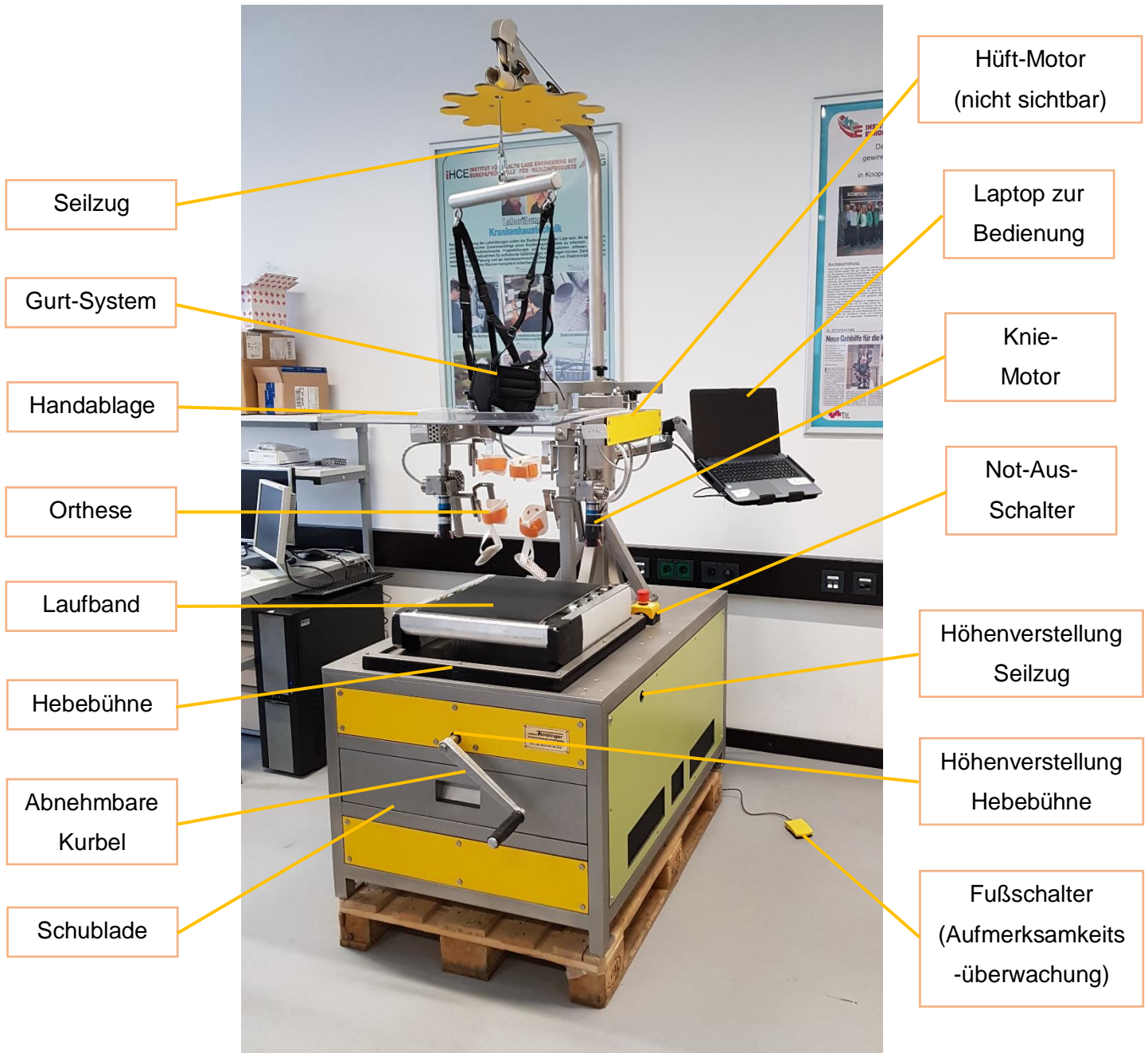


Abbildung 51: Finales Re-Design des Lokomotionsgeräts

## 4 Ergebnisse

Im folgenden Kapitel werden die Ergebnisse des Re-Designs der Lokomotionsgeräte-Software präsentiert.

### 4.1 Software-Sicherheitsklassifizierung

Im Folgenden werden die Risikomatrix der ursprünglichen Software und die neue Risikomatrix abgebildet. Fehlfunktionen durch die Software wurden mit einer Wahrscheinlichkeit von 100% angenommen. Risikokontrollmaßnahmen umgesetzt durch die Software durften, im Gegensatz zu externen, nicht berücksichtigt werden. Bezüglich der vollständigen Risikoanalyse wird auf das Dokument „RA-V1\_Re-Design.docx“ [40] im digitalen Anhang verwiesen.

Tabelle 14: Ursprüngliche Risikomatrix für die Software-Sicherheitsklassifizierung [12]

<b>Eintrittswahrscheinlichkeit</b>	6	<b>häufig</b>	0	0	0	0	<b>Risikostufe</b>
	5	<b>manchmal</b>	0	0	0	0	
	4	<b>gelegentlich</b>	0	3	0	0	
	3	<b>selten</b>	1	4	0	0	
	2	<b>unwahrscheinlich</b>	0	6	0	0	
	1	<b>unglaublich</b>	0	0	1	0	
			<b>gering</b>	<b>mittel</b>	<b>schwer</b>	<b>katastrophal</b>	
			1	2	3	4	
			<b>Schadensfolge</b>				

Tabelle 15: Aktuelle Risikomatrix für die Software-Sicherheitsklassifizierung

<b>Eintrittswahrscheinlichkeit</b>	6	<b>häufig</b>	0	0	0	0	<b>Risikostufe</b>
	5	<b>manchmal</b>	0	0	0	0	
	4	<b>gelegentlich</b>	0	3	0	0	
	3	<b>selten</b>	1	4	0	0	
	2	<b>unwahrscheinlich</b>	2	8	0	0	
	1	<b>unglaublich</b>	0	0	1	0	
			<b>gering</b>	<b>mittel</b>	<b>schwer</b>	<b>katastrophal</b>	
			1	2	3	4	
			<b>Schadensfolge</b>				

Tabelle 15 zeigt die Zusammenfassung der Risikoanalyse für das Re-Design der Software in Form einer Risikomatrix. Der Unterschied zur Risikoanalyse der letzten Version (Tabelle 14) liegt in der zusätzlich verbauten Hardware (Laufband und Waage) begründet. Die Anzahl der Risiken ist, zurückzuführen auf die zusätzlichen Funktionalitäten, insgesamt leicht gestiegen. Im Vergleich zu den vormals 15 Risiken wurden nun in Summe 19 Risiken zur Bewertung der Software-Sicherheitsklasse identifiziert.

Demnach entfallen von den vier neu hinzugekommenen Risikosituationen eine auf den Betrieb des Laufbands und drei Risiken auf die Gewichtsmessung mit der Waage, siehe Abbildung 52.

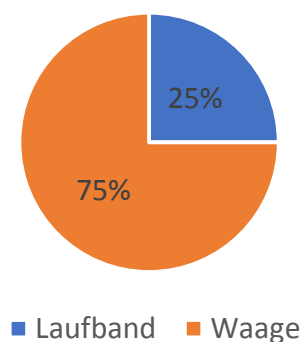


Abbildung 52: Neue relevante Risiken für die Software-Sicherheitsklassifizierung durch das Re-Design

#### 4.1.1 Risiko des Laufbands

Das auf den Betrieb des Laufbands zurückzuführende Risiko besteht in einer möglichen Asynchronität zu den Orthesen. Dabei werden sowohl die Ereignisse, dass das Laufband sich zu schnell, als auch zu langsam bewegt, inklusive dessen Stillstand, berücksichtigt. Alle genannten Fälle könnten Quetschungen, Muskelverletzungen oder Knochenfrakturen beim Patienten herbeiführen. Externe Risikobeherrschungsmaßnahmen stellen hierfür der Not-Aus sowie die Gebrauchsanweisung dar. Letztere gibt vor, den Patienten erst bei Therapiestart auf Laufband-Niveau abzusenken, um eine direkte Blockade beim Start und eine daraus resultierende Verletzungsgefahr zu vermeiden.

#### 4.1.2 Risiken in Verbindung mit der Gewichtsmessung

Die Messung eines fehlerhaften Patientengewichts kann drei Auswirkungen hervorrufen. Wird ein zu geringes Patientengewicht gemessen, könnte das den Therapeuten dazu veranlassen, ein geringeres Kraftlevel (Drehmomentgrenze) einzustellen. Die Therapie wäre dementsprechend weniger wirkungsvoll, was keine Verletzungen nach sich ziehen würde. Wird ein höheres Patientengewicht gemessen, als der Patient tatsächlich aufweist, könnte es

den Therapeuten dazu veranlassen, ein größeres Drehmoment als vorgesehen zuzulassen. Der Patient könnte dadurch Muskelverletzungen erleiden. Ist im dritten Fall das Messergebnis der Waage dahingehend inkorrekt, dass das höchst zulässige Gewicht zur Durchführung der Lokomotionstherapie überschritten wird, ist der jeweilige patientenbezogene Datensatz nicht erstellbar- bzw. abänderbar. Somit wäre eine Therapie erst gar nicht durchführbar. Für diese drei Auswirkungen einer möglichen Fehlmessung existieren keine geeigneten konstruktiven oder mittelbaren Sicherheitsmaßnahmen. Lediglich die Gebrauchsanweisung schreibt die regelmäßige zusätzliche Kontrolle des Patientengewichts mit einer geeichten Personenwaage vor.

#### **4.1.3 Risikobewertung zur Software-Sicherheitsklassifizierung**

Die in Tabelle 15 in Summe dokumentierten 19 Risiken verteilen sich auf die Risikostufen III und IV. Risiken der Stufe I und II wären vorab ohnedies als nicht tolerierbar eingestuft worden bzw. hätten bei Auftreten zusätzliche Risiko-Beherrschungsmaßnahmen impliziert. Die Tabelle weist 11 Risiken der Stufe IV, welche als vernachlässigbar betrachtet werden, aus. Die übrigen 8 Risiken fallen in die Risikostufe III. Das ALARP-Prinzip schreibt vor, dass diese nur tolerierbar sind, wenn eine Vermeidung nicht praktikabel ist. Die ursprüngliche und die aktuelle Risikomatrix weisen eine höchstmögliche Schadensfolge, die im Einzelfall eintreten kann auf, welche einem schweren Schaden entspricht. Gleichzusetzen ist diese Bewertung mit einer schweren Verletzung, ein letaler Ausgang kann in diesem Fall jedoch ausgeschlossen werden. Der Fall, dass die Aufsichtspflicht gegenüber dem Patienten vernachlässigt wird und die Software gleichzeitig eine Fehlsituation auslöst, könnte theoretisch eine schwere Verletzung bewirken. Dieses Ereignis wurde jedoch mit einer Eintrittswahrscheinlichkeit der Stufe 1 (unglaublich) beurteilt. Grundsätzlich ist die höchste Eintrittswahrscheinlichkeit, mit der eine Schädigung auftreten kann, mit Stufe 4 (gelegentlich) gegeben.

#### **4.1.4 Ergebnis der Software-Sicherheitsklassifizierung**

Insgesamt besteht die Möglichkeit, dass das Software-System zu einer Gefährdungssituation beitragen kann. Die externen Risiko-Beherrschungsmaßnahmen können ein unvertretbares Risiko nicht verhindern, jedoch führt der Schaden im Regelfall zu keiner schweren Verletzung. Unter Einbeziehung, dass kein Risiko jemals einen letalen Ausgang haben kann und darüber hinaus keine schweren oder katastrophalen Schäden eintreten können, entspricht das Software-System der **Software-Sicherheitsklasse B**.

## 4.2 Software-Risikomanagement

Im Zuge des Software-Risikomanagements wurden alle bereits erkannten vorhandenen und neu hinzugekommenen Risiken bewertet. Die bereits durchgeführten Risikokontrollmaßnahmen wurden dementsprechend um neue ergänzt. Da die Restrisiken vorab bereits ein dermaßen niedriges Niveau einnahmen, konnte das Re-Design im Allgemeinen keine weitere Minderung der Risiken mehr erwirken. Begründet ist dies unter anderem, dass trotz neuer Risikokontrollmaßnahmen die Risiken teilweise noch immer gleich zu bewerten waren. Beispielhaft angeführt wird an dieser Stelle die Aufmerksamkeitsüberwachung durch den Therapeuten. Es wurde zusätzlich zur Überwachung der Mausbewegung ein Fußschalter adaptiert, dennoch ist das Restrisiko mit derselben Eintrittswahrscheinlichkeit „unwahrscheinlich“ zu bewerten. Die Wahrscheinlichkeit „unglaublich“ ist nicht zu vergeben, da die Verantwortung nach wie vor beim Therapeuten liegt und der Fußschalter nur ein Hilfsmittel darstellt. Einen weiteren Grund, dass die Anzahl der Risiken nach dem Re-Design nahezu gleichgeblieben ist, stellen behobene Fehler dar, welche anschließend definitiv nicht mehr auftreten können (Eintrittswahrscheinlichkeit = 0 %) und deshalb im Risikomanagement keine Berücksichtigung mehr erfahren. Als Beispiel genannt sei hier die Änderung eines Passworts bzw. die Eingabe des Passworts bei Erstellung eines neuen Anwenders (Therapeuten). Vor dem Re-Design führte Drücken der Eingabetaste im Passwortfeld zu einem Zeilenumbruch. Blieb dieser womöglich unbemerkt, konnte sich der Therapeut nicht mehr im System anmelden. Durch das Re-Design wird einerseits die Eingabe des Passworts auf eine Zeile limitiert, zusätzlich dient das Drücken der Eingabetaste nun zur Bestätigung der Eingabe. Der Großteil der Änderungen der Software zielte hauptsächlich auf deren Gebrauchstauglichkeit ab oder waren Korrekturen von Fehlern, welche keine Auswirkungen auf die Risiken hatten.

Die identifizierten und durch Abhilfemaßnahmen verminderten Risiken sind in Tabelle 16 ersichtlich.

Tabelle 16: Risikomatrix des Software-Risikomanagements nach Durchführung der Abhilfemaßnahmen

Eintrittswahrscheinlichkeit	6	häufig	0	0	0	0	Risikostufe
	5	manchmal	0	0	0	0	
	4	gelegentlich	0	0	0	0	
	3	selten	0	1	0	0	
	2	unwahrscheinlich	10	31	2	0	
	1	unglaublich	9	27	5	0	
			gering	mittel	schwer	katastrophal	
			1	2	3	4	
			Schadensfolge				

Durch das Re-Design kamen acht neue Risiken begründet durch die Funktionalität hinzu, wodurch sich die Gesamtanzahl der Risiken auf 85 summiert. Die auslösenden Risiko-Ursachen, siehe Tabelle 17, wurden einer Fehlermöglichkeits- und Einflussanalyse unterzogen, um die verschiedenen Folgen von Ereignissen zu den Gefährdungssituationen zu eruieren.

Tabelle 17: Risiko-Ursachen und angewandte Kontrollmaßnahmen des Re-Designs

Risiko-Ursache	Angewandte Risikokontrollmaßnahme
Fehlerhafte Gewichtsmessung	RA-V2-35
Ändern der Zielkadenz durch den Anwender	RA-V2-36
Verlust der Synchronität des Laufbands zu den Orthesen	RA-V2-37

Die ursprünglich 34 definierten Risikokontrollmaßnahmen wurden um weitere drei ergänzt, siehe Tabelle 18.

Tabelle 18: Zusätzliche Risikokontrollmaßnahmen des Re-Designs

Identifikation	Risikokontrollmaßnahme
RA-V2-35	Vorschrift Gebrauchsanweisung: Verwendung einer geeigneten medizinischen Waage zur Kontrolle des Gewichts
RA-V2-36	Beschränkung der Geschwindigkeitsänderung bei Zielkadenzänderung (Laufband)
RA-V2-37	Betätigung des Not-Aus-Schalters

### 4.2.1 Gesamtbewertung

Zur Gesamtbewertung des Risikos müssen eventuelle Abhängigkeiten zwischen den einzelnen Risiken, als auch die Möglichkeit eines zeitgleichen Eintritts mehrerer Einzelrisiken berücksichtigt werden. Es konnten keine Risiken bzw. Kombinationen von Risiken identifiziert werden, die zu einem nicht akzeptablen Gesamtrisiko hätten führen können.

Die aktualisierte Risikoanalyse „RA-V2\_Re-Design.docx“ [41] sowie die aktualisierten Risikokontrollmaßnahmen „RA-V2- Risikobeherrschungsmaßnahmen\_Re-Design .docx“ [42] sind im digitalen Anhang beinhaltet.

### 4.3 Software-Anforderungen

Die ursprüngliche Lokomotionsgerätesoftware wurde durch insgesamt 86 Software-Anforderungen beschrieben. Daraus haben sich 39 Anforderungen aus System-Anforderungen, 46 aus Risiko-Beherrschungsmaßnahmen und eine weitere aus der Umsetzung der Maschinendirektive abgeleitet. [12]

Um dem Re-Design der Software gerecht zu werden, wurden zu Beginn die System-Anforderungen (SysA) ergänzt. Die Aufgabenstellung wurde durch insgesamt 5 neu eingeführte System- Anforderungen wie folgt umgesetzt:

- SysA 23: Möglichkeit zur Bestimmung des Patientengewichts mit zuvor eingebauten Sensoren
- SysA 24: Alarmierung bei aktivem Not-Aus Schalter in der Software
- SysA 25: Ansteuerung des automatisierten Laufbands
- SysA 26: Einbinden eines Fußschalters zur Aufmerksamkeitsüberwachung
- SysA 27: Exportmöglichkeit der Patientendaten (Patientenakte)

Die Usability-Ergebnisse [13], welche neue Softwarefunktionen vorsahen, sind bereits in den System-Anforderungen integriert. Die übrigen Usability-Änderungen betrafen vorhandene Software-Komponenten und konnten durch Problem- und Änderungsanträge umgesetzt werden, siehe Kapitel 4.8.

Aus den neuen System Anforderungen ließen sich die Software -Anforderungen (SWA) ableiten. Exemplarisch wird ein Usability-Ergebnis, der Wunsch nach der Möglichkeit des Ausdrucks der Therapiedaten, genannt. Die entsprechende System-Anforderung lautet: „Exportmöglichkeit der Patientendaten (Patientenakte)“. Eine Ebene tiefer lauten die abgeleiteten Software-Anforderungen: „Möglichkeit zum Speichern der Daten“ und „Möglichkeit zum Drucken der Daten“.

Die aus den 5 SysA resultierenden 9 SWA können den folgenden Software-Hauptkomponenten zugeordnet werden:

- Anwenderverwaltung: 17 Software-Anforderungen
- Patientenverwaltung: 23 (+5) Software-Anforderungen
- Therapiemodus: 45 (+4) Software-Anforderungen
- Fehlerbehandlung: 1 Software-Anforderung

Sämtliche aktualisierten System- und Software-Anforderungen sowie das Traceability Dokument zur Rückverfolgbarkeit sind im digitalen Anhang unter „System-Anforderungen-V-1.docx“ [43], „Software-Anforderungen-V-0.3.docx“ [44] beziehungsweise „Traceability-V-0.3.docx“ [45] dokumentiert.

#### 4.4 Software–Architektur

Die grundlegende Architektur der Software wurde durch das Re-Design nicht geändert. Die Lokomotionstherapiesoftware ist in vier Hauptsoftware-Komponenten unterteilt. Die Anwenderverwaltung, die Patientenverwaltung, den Therapiemodus und die Fehlerbehandlung, siehe Abbildung 53.

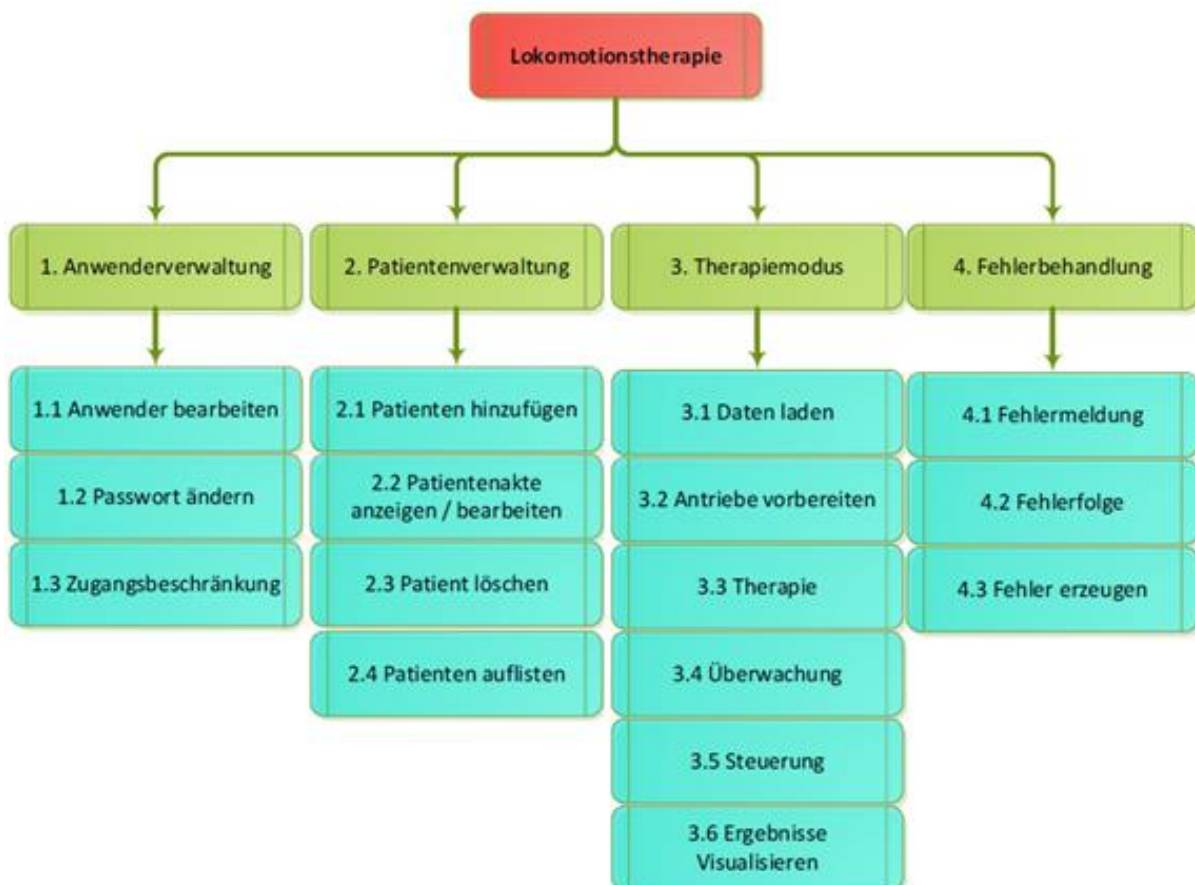


Abbildung 53: Vereinfachtes Hierarchiediagramm der Lokomotionstherapie-Software [12]



Die Architektur wird im Dokument „Software-Architektur-Hauptdokument-V-0.1.docx“ [46] beschrieben und ist dem digitalen Anhang beigefügt. Es verweist auf die entsprechenden Dokumente bezüglich der Struktur, dem Ablauf und der Darstellungen der einzelnen Software-Hauptkomponenten. Zusätzliche Dokumente für die Definition der Variablen sowie die Spezifikationen der SOUPs ergänzen die Architektur. Ein ausführliches Hierarchiediagramm, welches die Ein- und Ausgangsvariablen und die jeweiligen Hardwareschnittstellen der Software aufzeigt, befindet sich im Anhang in Abbildung 64.

Der Ablauf der Software wird in den nachfolgenden Kapiteln erläutert.

#### **4.4.1 Ablauf der Lokomotionstherapie-Software Version 0.3**

Zu Beginn verlangt die Software die Authentifizierung des Anwenders. Der Therapeut wird aufgefordert, sich mittels Eingabe von Benutzernamen und Passwort im System anzumelden. Sind die Daten korrekt, gelangt man ins Hauptmenü, andernfalls erfolgt die erneute Abfrage. Wird das Anmeldefenster vom Benutzer geschlossen, beendet dies das Programm automatisch.

Im Hauptmenü werden in der Statusleiste der eingeloggte Therapeut, sowie das aktuelle Datum und die Uhrzeit angezeigt. Durch Drücken der entsprechenden Buttons hat der Anwender folgende Auswahlmöglichkeiten:

- Therapiemodus: Startet den Therapiemodus
- Patienten verwalten: Öffnet die Patientenverwaltung
- Passwort ändern: Änderung des eigenen Passworts
- Therapeut wechseln: Ein Anwenderwechsel ist möglich
- Therapeut bearbeiten: Andere Anwender bearbeiten (Administratorrechte vorausgesetzt)
- Beenden: Schließt die Anwendung

#### **Therapiemodus**

Durch Aufruf des Therapiemodus erscheint eine Liste aller im System angelegten Patienten. Ein noch nicht erfasster Patient muss vorab in der Patientendatenbank angelegt werden. Die Auswahl bestätigt, werden die Patientendaten inklusive Foto in den Therapiemodus übernommen. Im Therapiemenü ist das Kraftlevel, die Kadenz und die Therapiedauer festzulegen.

Im nächsten Schritt werden die Antriebe vorbereitet, indem die folgenden Vorgänge sukzessiv durchgeführt werden:

- Initialisierung der USB2CAN Schnittstelle
- Initialisierung der Motion-Controller
- Konfiguration der Motion-Controller
- Referenzieren der Antriebe

Der Patient wird in den dafür vorgesehenen Halterungen fixiert und die Beine mit den Orthesen verbunden. Die erfolgreiche Vorbereitung muss in einer Abfrage bestätigt werden. Anschließend werden die Antriebe aktiviert, die Startposition angefahren und mit der Therapiedurchführung begonnen.

Details für eine Auswertung werden nach erfolgter Therapie in einem zusätzlichen Fenster angezeigt. Durch Beenden des Therapiemodus erscheint erneut das Hauptmenü.

### **Patienten verwalten:**

Bei Aufruf werden alle Patienten in der Datenbank in einer Liste angezeigt. Es ist möglich Patienten zu löschen oder neue hinzuzufügen. Notwendige Änderungen von Patientendaten können durch Anzeige der Patientenakte durchgeführt werden. Die Patientenakte kann gedruckt bzw. exportiert werden. Bei Änderungen werden diese sofort in die Patientenliste übertragen und aktuell angezeigt.

### **Passwort ändern**

In dieser Ansicht kann der Anwender sein eigenes Passwort ändern.

### **Therapeut wechseln**

Das Programm führt wieder zur Authentifizierung des Anwenders, wie zu Beginn.

### **Therapeut bearbeiten**

Der folgende Menüpunkt ist nur für Administratoren freigeschaltet. Bei Aufruf werden alle Anwender aus der Datenbank geladen und übersichtlich in einer Liste dargestellt. Es ist möglich, Anwender zu löschen, zu bearbeiten oder neue hinzuzufügen. Die Bearbeiten-Funktion ermöglicht auch die Vergabe von Administratorrechten. Wurden Änderungen durchgeführt, werden diese umgehend in der Anwenderliste übernommen und angezeigt.

### **Beenden**

Die Anwendung wird geschlossen.

Auf nachfolgender Seite wird in Abbildung 54 der beschriebene Programmablauf grafisch dargestellt. Eine bebilderte Bedienungsanleitung [47], welche im Zuge dieser Arbeit erstellt wurde, dokumentiert die exakte Anwendung der Lokomotionsgerätesoftware.

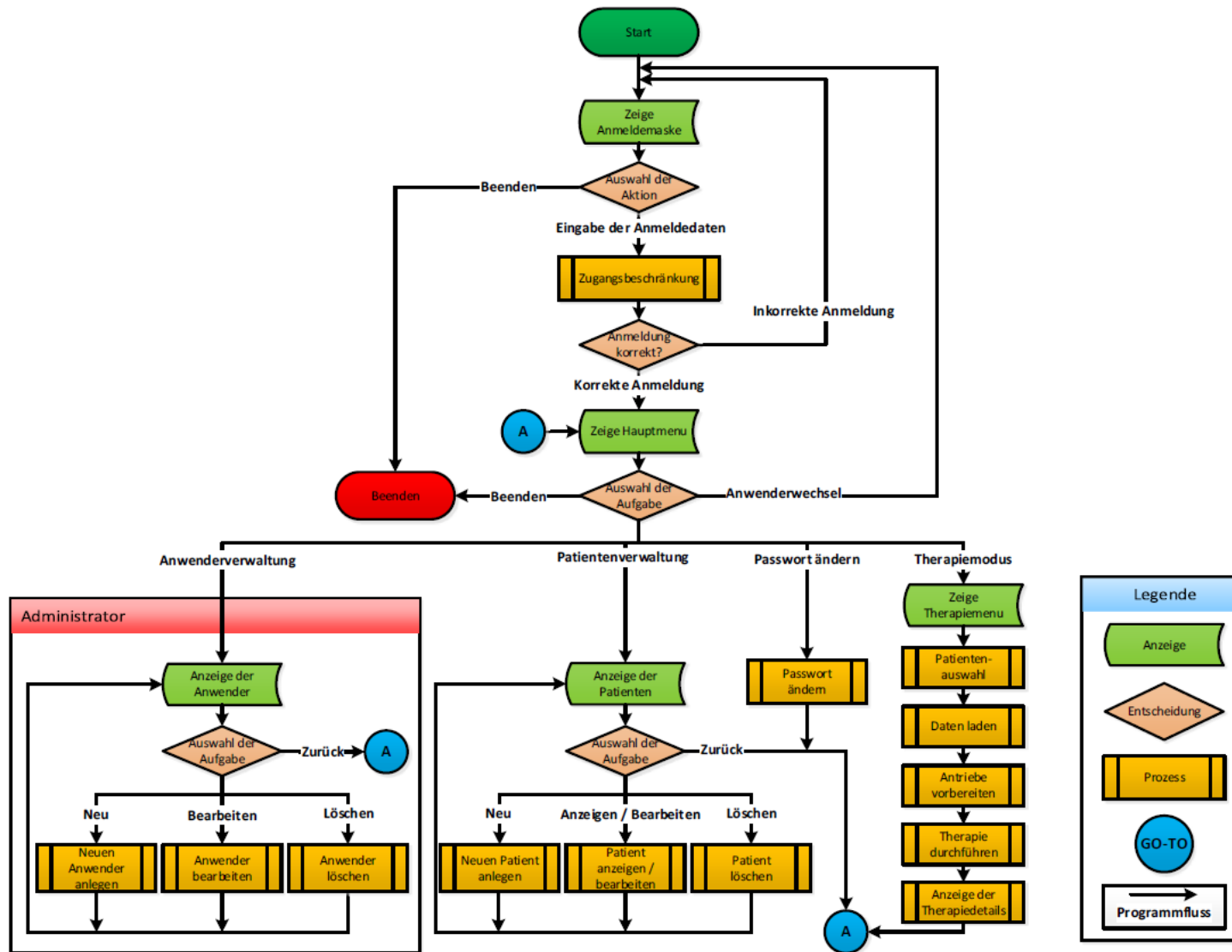


Abbildung 54: Ablauf der Lokomotionstherapie-Software

## 4.5 Software-Detail-Design

Das Software-System umfasst nach dem Re-Design 87 Software-Komponenten, welche in ihren Design-Dokumenten ausführlich beschrieben sind. Übersichtlich dargestellt bildet Tabelle 19 die Gesamtheit der Software-Komponenten ab. Acht Software-Komponenten sind durch das Re-Design neu hinzugekommen, ihre einzelnen Bezeichnungen sind in der Tabelle grün markiert. Geändert hingegen wurden 31 der bereits vorhandenen 79 Komponenten, deren Bezeichnungen in der Tabelle orange markiert sind. Weiterführende Informationen werden im „Detail-Design-Hauptdokument-V-0.3.docx“ [48] dargestellt, welches den Ausgangspunkt für die zugehörigen Design-Dokumente im digitalen Anhang bildet. Sämtliche Änderungen an der Lokomotionsgeräte-Software sind im Anhang in Kapitel D aufgelistet.

Tabelle 19: Übersicht aller Software-Komponenten

0. Lokomotionstherapie	3.0.9 Timer
0.1 Prüfsumme	3.0.10 Byte2Nmbr
0.3 Sort 2D Array	3.1 Daten laden
0.4 Select Printer	3.1.1 xml-String laden
0.5 Not-Aus Überwachung	3.1.2 Patient wählen
0.6 Gewicht messen	3.2 Antriebe vorbereiten
1. Anwenderverwaltung	3.2.1 Init USB2CAN
1.0.1 Anwenderdaten laden	3.2.1.1 Reset USB-CAN
1.0.2 Anwenderdaten schreiben	3.2.2 Init Endstufe
1.1 Anwender bearbeiten	3.2.3 Referenzierung
1.1.1 Neuen Anwender erzeugen	3.2.3.1 Automatisch Referenzieren
1.1.2 Anwender ändern	3.2.3.2 Manuell Referenzieren
1.1.3 Anwender löschen	3.2.3.0.1 Nullpunkt verschieben
1.1.4 Anwender darstellen	3.2.4 Setting Antriebe
1.1.5 Anwender sortieren	3.3 Therapie
1.2 Passwort ändern	3.3.1 Startposition anfahren
1.3 Zugangsbeschränkung	3.3.2 zyklische Bewegung
2. Patientenverwaltung	3.3.2.1 Schrittlänge berechnen
2.0.1 Alter berechnen	3.4 Überwachung
2.0.2 Patientenakte schreiben	3.4.1 Leseschleife
2.0.3 Patientenfoto kopieren	3.4.1.1 CAN-Nachrichten Verwalten
2.0.4 Patientenakte laden	3.4.1.1.1 Abfrage Verwaltung
2.0.5 Foto laden	3.4.1.1.2 Antriebsstatus Verwalten

2.1 Patient hinzufügen	3.4.1.1.3 Weitergabe undefinierte Nachrichten
2.2 Patientenakte anzeigen bearbeiten	3.4.1.1.4 Node Guarding Verwaltung
2.2.1 Patientendaten ändern	3.4.2 Zustandsüberwachung
2.2.2 Therapiedetail anzeigen	3.4.3 Drehmomentüberwachung
2.2.3 Übersicht der Therapien darstellen	3.4.4 Missing Code Überwachung
2.2.4 Therapieinfos2Grafik	3.4.5 Node Guarding
2.2.5 Aktualisierungsdatum prüfen	3.4.6 Abfrage Schleife
2.2.6 Patient identifizieren	3.4.6.1 Abfrage-Konfiguration
2.2.7 Patientenakte drucken	3.4.6.2 Abfrage anfordern
2.2.8 Patientenakte exportieren	3.4.7 Aufmerksamkeitsüberwachung
2.3 Patient löschen	3.5 Steuerung
2.4 Patienten auflisten	3.5.1 Orthesen einstellen
3. Therapiemodus	3.5.2 Therapieinfos live darstellen
3.0.1 Antriebe starten	3.5.3 Online Fehlerbehandlung
3.0.2 Kraftlevel einstellen	3.6 Ergebnisse visualisieren
3.0.3 NodeID2Antrieb	3.6.1 Speichern der Therapieinformationen
3.0.4 Bewegung vorbereiten	4. Fehlerbehandlung
3.0.5 Bewegung starten	4.1 Fehlermeldung
3.0.6 Antriebe abschalten	4.2 Fehlerfolge
3.0.7 Grad zu Inkremente	4.3 Fehler erzeugen
3.0.8 Inkremente zu Grad	

## 4.6 Implementieren-Verifizieren-Integrieren

Ausgehend vom Aspekt eines Re-Designs der Software wurden die erstellten und vorhandenen Änderungsanträge in einer gemeinsamen IVI-Phase durchgeführt. Werden die vorangegangenen Iterationen, die für die Erstellung der Software „Lokomotionstherapie-V-0.2“ nötig waren, miteingeschlossen, handelt es sich hier bereits um die vierte Iteration. Einen vollständigen Überblick zu den neuen bzw. geänderten Softwarekomponenten und den zugehörigen Testcode zur Verifizierung gibt das Dokument „Implementieren-Verifizieren-Integrieren-V-0.3.docx“ [49].

Alle 39 Software-Komponenten, die geändert bzw. erstellt werden mussten, um das Software-System **Lokomotionstherapie-V-0.3** zu entwickeln, wurden implementiert und erfolgreich getestet. Der aktualisierte „Implementieren-Verifizieren-Integrieren-Plan“, die zugehörigen 39 neuen Testprotokolle als auch die ursprünglichen Testprotokolle sind im digitalen Anhang beinhaltet.

## 4.7 Verifizierung des Software-Systems

Eine abschließende Software-System Verifizierung wurde mit der **Lokomotionstherapie-Software Version 0.3** durchgeführt. Nach dem Re-Design wurden 95 Software-Anforderungen durch einen umfangreichen Funktionstest geprüft.

Vier Software-Anforderungen konnten nicht durch die Software-System Verifizierung überprüft werden, diese wurden jedoch bereits in der IVI-Phase verifiziert.

Die Überprüfung aller Software-System Anforderungen kann als positiv bewertet werden und das Software-System wurde somit erfolgreich verifiziert. Der vollständige Bericht „SoftwareSystem-Verifizierung-V-0.1.docx“ [50] ist im digitalen Anhang ausgewiesen.

### 4.7.1 Verifizierung der Bewegung der Orthesen

Die Erfüllung der Forderung eines physiologischen Gangs soll in diesem Abschnitt verdeutlicht werden. Um die Bewegungsmuster zu verifizieren, wurden diese aufgezeichnet und mit den Soll-Kurven überlagert. Der Datensatz zur Verifizierung, welcher auch zur ursprünglichen Bewegungserstellung diente, stammt von der Klinik Judendorf-Straßengel [51]. Dabei wurden die Verläufe der Knie- und Hüftwinkel von 12 Kindern aufgezeichnet, gemittelt und als Datensatz mit 101 Werten (0-100) zur Verfügung gestellt.

Zur Auswertung wird eine Studie, welche Referenzdaten für den Hüft- und Kniewinkel während des Gangs liefert, verwendet. Die Studie führt zum Ergebnis, dass geringfügige Änderungen in Abhängigkeit des Patientenalters auftreten. Sie zeigte, dass keine Seitenunterschiede, jedoch signifikante Geschlechtsunterschiede als auch Variationen mit der Gehgeschwindigkeit vorkommen. Die jeweils geringsten Werte für die Standardabweichung, ungeachtet des Geschlechts, wurden aus der Studie extrahiert und in Tabelle 20 dargestellt. [52]

*Tabelle 20: Standardabweichung der Knie- und Hüftwinkel in Abhängigkeit der Ganggeschwindigkeit*

Langsamer Gang		Normaler Gang		Schneller Gang	
$\sigma$ [°]		$\sigma$ [°]		$\sigma$ [°]	
Knie	Hüfte	Knie	Hüfte	Knie	Hüfte
4,6	5,4	4,8	5,3	5,1	6,2

Die nachfolgenden Abbildungen 55 bis 60 zeigen den Soll- sowie Ist-Verlauf der Winkel von Knie und Hüfte und den resultierenden Winkelfehler. Der jeweilige Fehler ist als 10-facher Betrag dargestellt, um die Unterschiede ersichtlicher zu machen. Während der Winkelfehler bei einer Kadenz von 20 bzw. 60 Schritten /min dieselbe Größenordnung aufweist, tritt der Höchstwert des Fehlers bei der maximalen Kadenz von 120 Schritten /min auf. Mit circa 2,5

Grad für das Knie und 2 Grad für die Hüfte sind die Grenzwerte deutlich unterschritten und die Bewegung kann demnach als physiologisch angesehen werden.

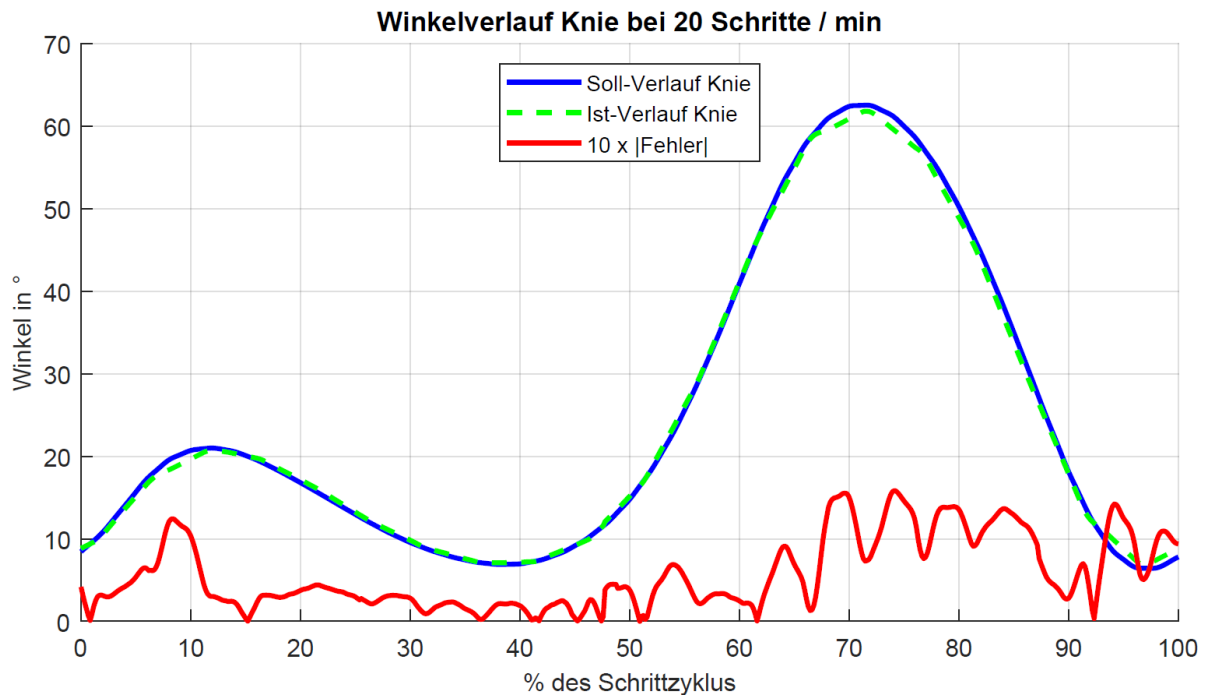


Abbildung 55: Soll- Ist-Vergleich Kniewinkel 20 Schritte/min

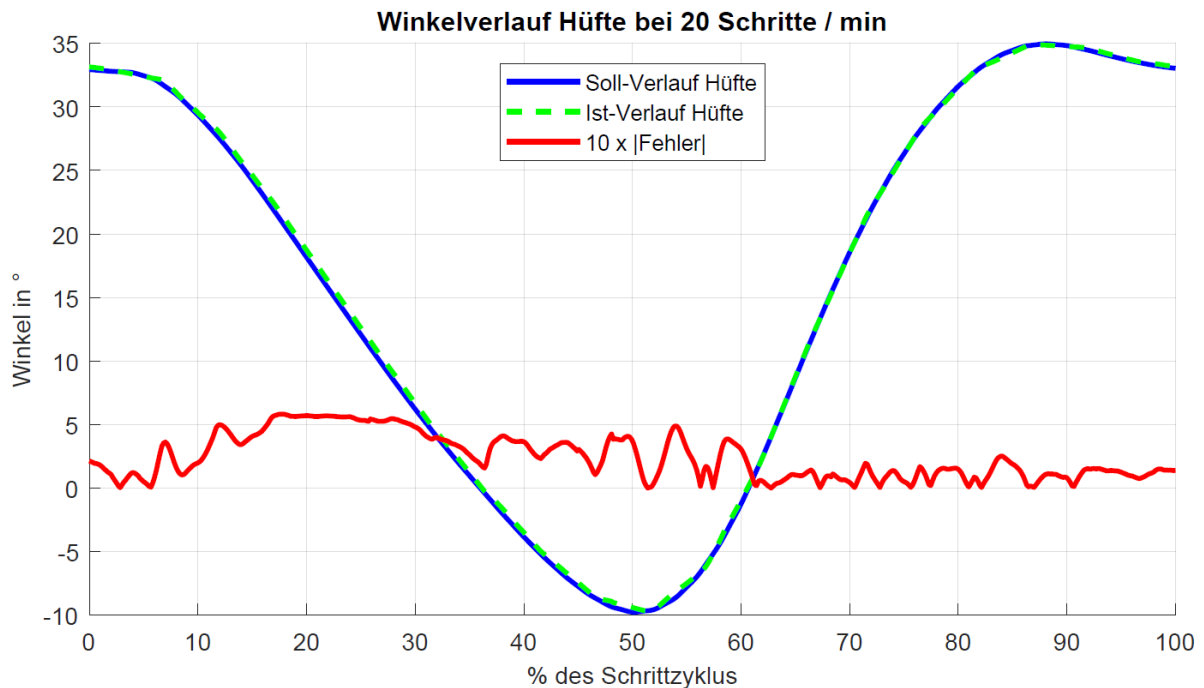


Abbildung 56: Soll- Ist-Vergleich Hüftwinkel 20 Schritte/min

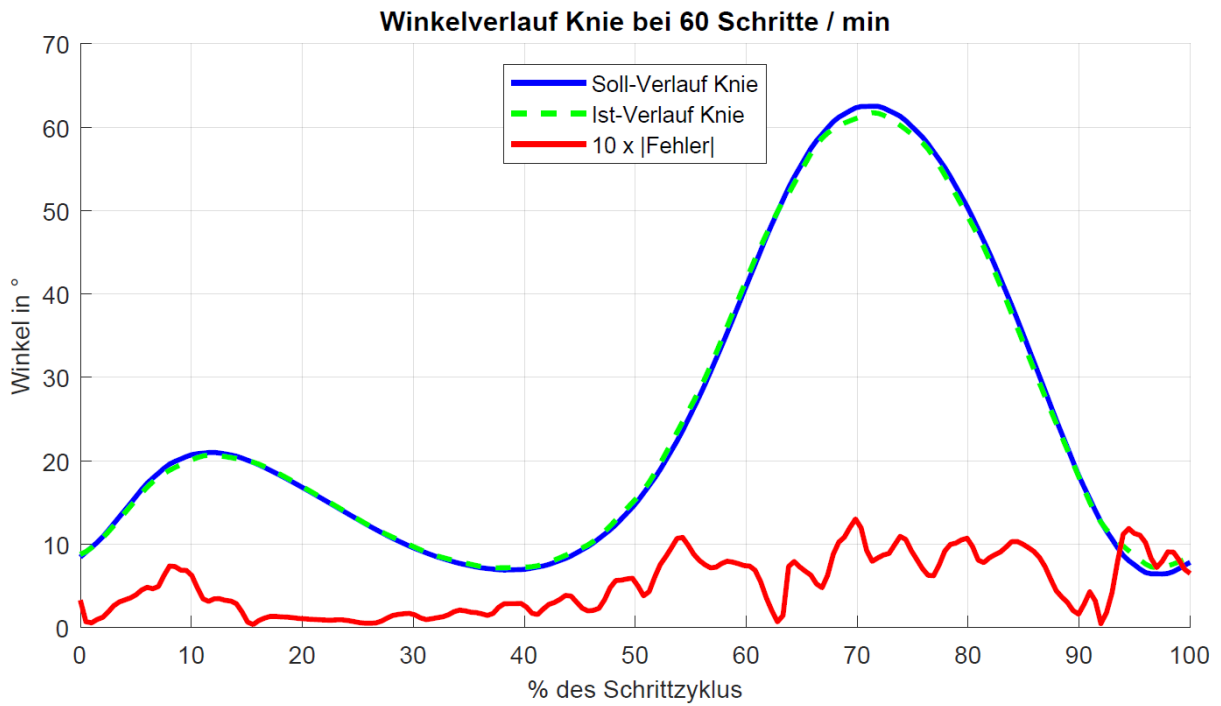


Abbildung 57: Soll- Ist-Vergleich Kniewinkel 60 Schritte/min

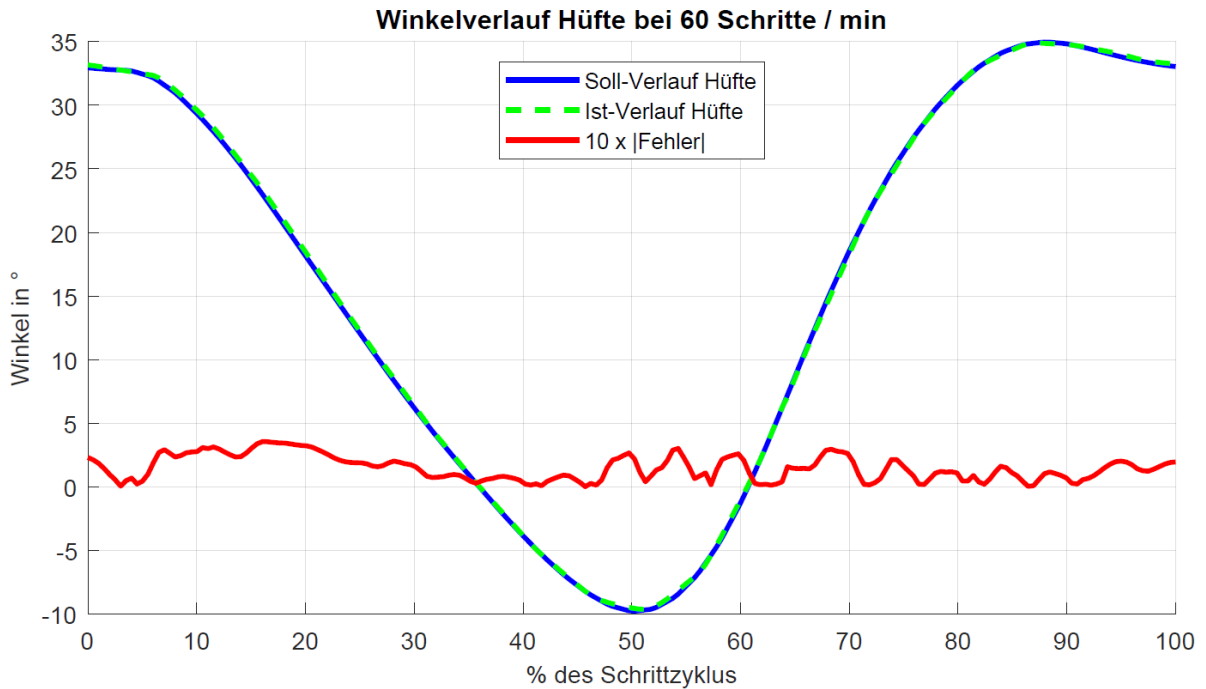


Abbildung 58: Soll- Ist-Vergleich Hüftwinkel 60 Schritte/min



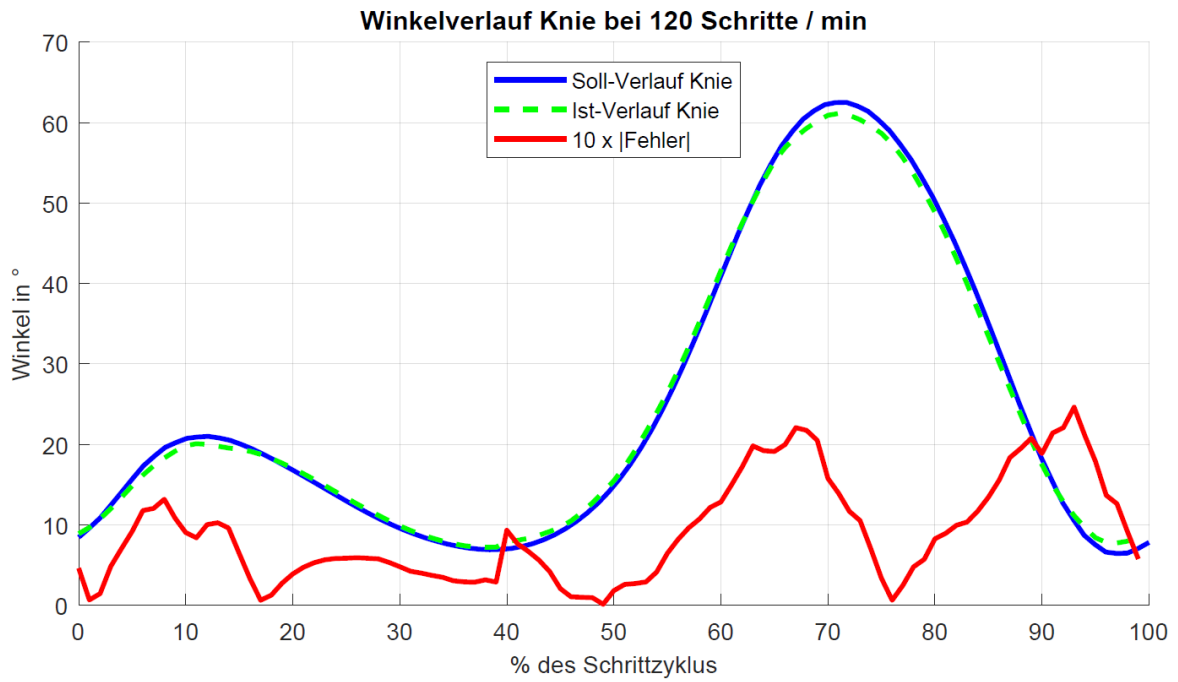


Abbildung 59: Soll- Ist-Vergleich Kniewinkel 120 Schritte/min

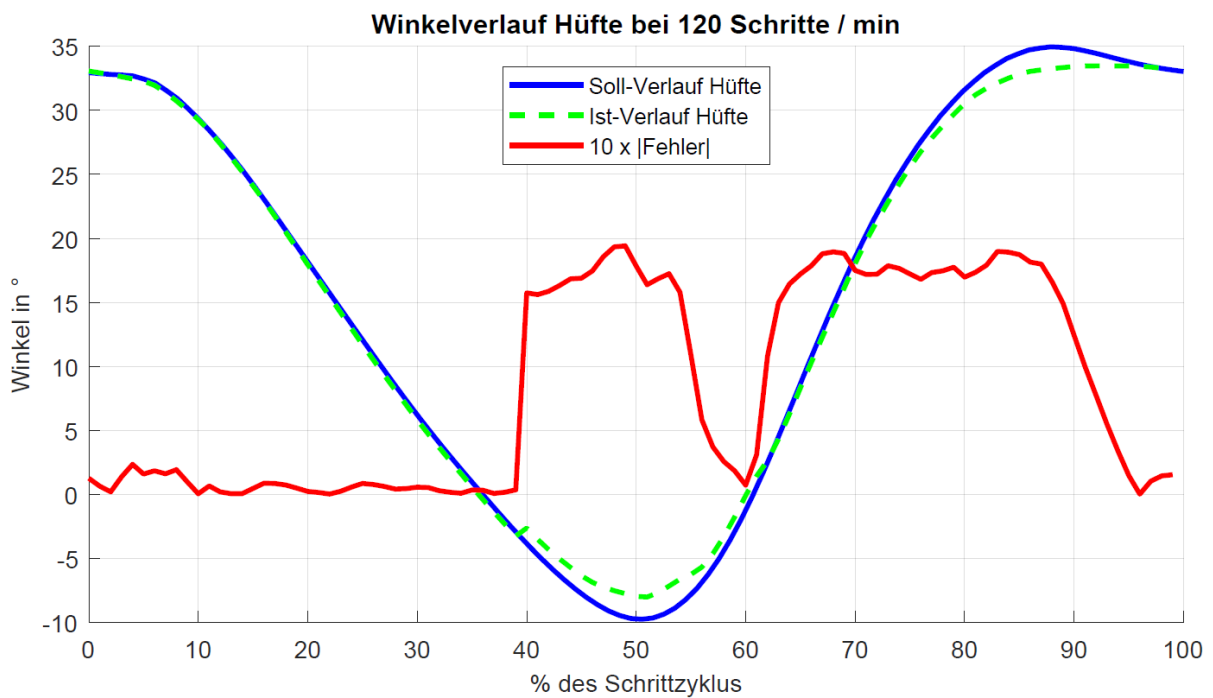


Abbildung 60: Soll- Ist-Vergleich Hüftwinkel 120 Schritte/min

### 4.7.2 Verifizierung der Geschwindigkeit des Laufbands

Für die Verifizierung der Laufbandgeschwindigkeit wurden die Soll-Geschwindigkeiten mit den Ist-Geschwindigkeiten verglichen. Die Vorgehensweise entspricht exakt der Aufnahme der Abweichung aus Kapitel 3.12.3.

Für die Verifizierung der Geschwindigkeit wurde das Laufband erneut mit einer optischen Markierung versehen und die Zeit für eine festgelegte Rundenzahl gemessen. Durch die gemessene Länge des Laufbands ( $s = 1,0071\text{m}$ ) konnte die tatsächliche Geschwindigkeit bestimmt werden.

Für die relative Abweichung der Geschwindigkeit in Prozent ( $\Delta v_{rel}$ ) gilt:

$$\Delta v_{rel} = \frac{v_{soll} - v_{ist}}{v_{soll}} \cdot 100[\%] \quad (45)$$

Die relativen Abweichungen der Geschwindigkeit wurden dokumentiert und sind in Abbildung 61 dargestellt. Positive Werte der Ordinate stehen für ein zu langsames Laufband, negative Werte hingegen stehen für ein zu schnelles Laufband im Vergleich zur vorgegebenen Geschwindigkeit durch die Software.

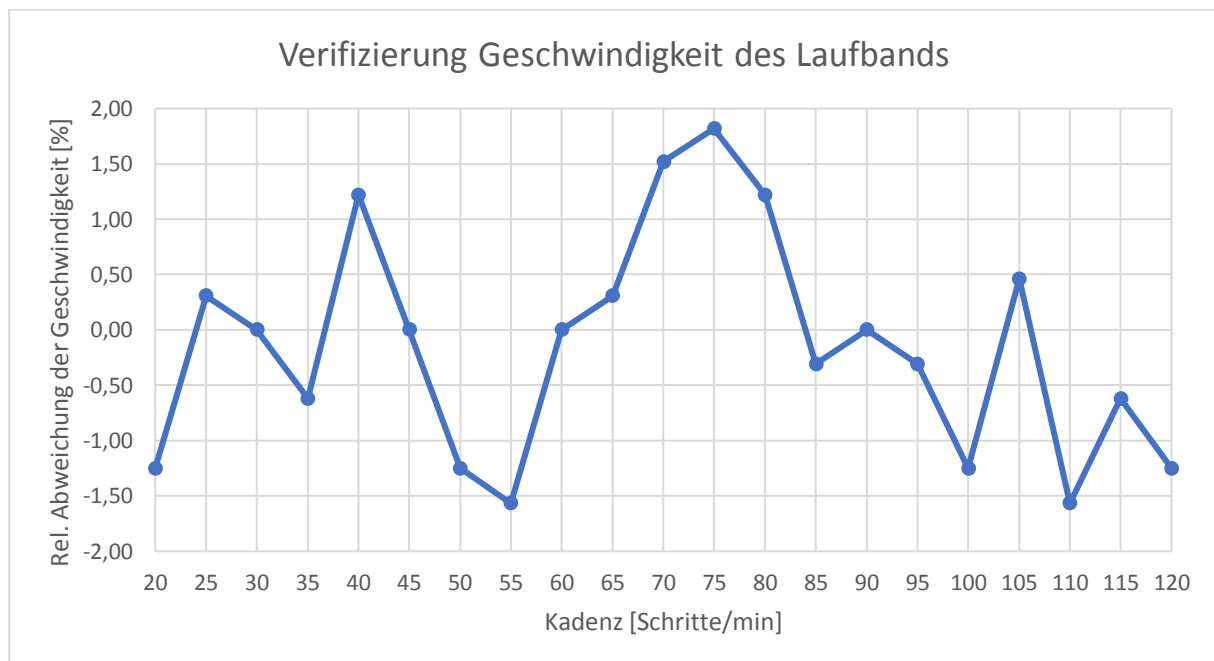


Abbildung 61: Verifizierung der Laufbandgeschwindigkeit

Die Verifizierung beweist, dass die Geschwindigkeit für alle Kadenzen nur noch im Bereich von  $\pm 2\%$  variiert. In diesem Bereich ist die zufällige Messabweichung, bedingt durch die manuelle Zeitmessung größer als die zu ermittelnde Messgröße, daraus folgt, dass die Ansteuerung des Laufbands durch die Software als exakt zu betrachten ist.

## 4.8 Problemlberichte und Änderungsanträge

Bei Übernahme der Software waren sowohl offene Problemlberichte, für welche noch keine Änderungsanträge erstellt wurden, als auch bereits erstellte aber nicht geschlossene Änderungsanträge vorhanden. Die Usability-Änderungsvorschläge von S. Langthaler [13] mussten zum Teil noch in konkrete Berichte überführt werden. Bei der bereits vorhandenen Dokumentation sowie den ausstehenden Usability-Änderungsvorschlägen wurde nach dem vorhandenen Schema vorgegangen. Manche Problemstellungen konnten durch Berücksichtigung in der Gebrauchsanweisung gelöst werden. Um jedoch ein entsprechendes Re-Design durchführen zu können, wurde in weiterer Folge vom existenten Schema abgewichen. Begründet wird dies insofern, dass bereits für die Änderung der Bezeichnung „Anwender“ zu „Benutzer“ auf einer Oberfläche ein Problemlbericht und ein Änderungsantrag erstellt werden mussten und für andere Vorkommnisse zusätzlich mögliche Testcases wiederholt werden mussten. Hinsichtlich eines im Rahmen dieses Re-Design überblickbaren Arbeitsaufwandes gelangte auch M. Kalkgruber nach der Erstellung der Erstversion zu folgender Erkenntnis:

*„Bei der Planung der Software-Entwicklung wurde von einem überschaubaren Entwicklungsaufwand ausgegangen. Auf Basis dieser Einschätzung wurde somit eine manuelle Verwaltung der Versionen gewählt. Mit der fortschreitenden Entwicklung wuchs jedoch der Dokumentationsaufwand stark an. Bei der Umsetzung der Revisionen stellte sich heraus, dass aufgrund der hierarchischen Software-Struktur, bereits bei kleinen Änderungen der Software mit einem beträchtlichen Aufwand zu rechnen ist. Dies betrifft einerseits die Software selbst, aber auch die Änderungen der Dokumentation.“*

Zitiert aus [12]

Somit wurden fortan sämtliche Änderungen innerhalb eines Revisionszyklus einer einzelnen Software-Einheit zusammengefasst. Mit jeweils einem Problem- sowie Änderungsantrag pro Einheit muss nicht jede einzelne Komplikation, dazu zählen auch Änderungswünsche, als solche behandelt werden. Die Forderungen der Norm EN62304 werden jedoch eingehalten bzw. umgesetzt.

## 4.9 Software-Freigabe

Die Aktivitäten des Software-Entwicklungsplans inklusive der Verifizierung der Software **Lokomotionstherapie-V-0.3** konnten erfolgreich abgeschlossen werden. Verbleibende Anomalien sind in Tabelle 21 aufgelistet und hinsichtlich ihres Risikos bewertet. Da von diesen keine Gefährdungen ausgehen, wurden die beschriebenen Abweichungen als vorübergehend akzeptierbar bewertet. Die freizugebende Software-Version wurde umfassend dokumentiert, einschließlich ihrer Generierung sowie die Umgebung, in der sie entwickelt wurde. Alle Bestandteile der Dokumentation und Software wurden archiviert.

Tabelle 21: Restanomalien der Lokomotionstherapie-Software V 0.3

Nr.	Beschreibung	Auswirkung auf die Sicherheit
1	Im gegebenen Entwicklungsstand ist keine Verstellung des Spitzen- und Dauerstroms der Motion Controller durch die Software möglich. Dadurch kann in Bezug auf die Grenzen nicht zwischen Referenzierung und Therapieausführung unterschieden werden. Somit mussten die höheren Stromgrenzen, welche bei Therapieausführung gelten, jeweils für Hüfte und Knie, festgelegt werden.	<p><b>Keine Auswirkung auf die Sicherheit</b></p> <p>Wird bei der Referenzierung der rasche Stromanstieg nicht erkannt, verhindern die Grenzen, welche bei Therapieausführung gültig und in den Motion Controllern gespeichert sind, eine Beschädigung des Lokomotionsgeräts.</p>

Die Software **Lokomotionstherapie-V-0.3** wird freigegeben und kann mit dem Re-Design des Lokomotionsgeräts für Kleinkinder ausgeliefert werden.

## 5 Diskussion

Die dieser Arbeit zugrundeliegende Aufgabenstellung ist die Entwicklung eines Re-Designs einer Ansteuerungssoftware eines Lokomotionsgeräts für Kleinkinder. Ein Gebrauchstauglichkeits-Test der ursprünglichen Software hat gezeigt, dass hierfür noch diverse Adaptierungen nötig waren. Zusätzlich dazu wurden im Zuge des Re-Designs auch neue Funktionen im Gesamtsystem implementiert.

### 5.1 Portierung und Debugging

Bevor mit dem eigentlichen Re-Design begonnen werden konnte, mussten mehrere Änderungen an der Software „Lokomotionstherapie-V-0.2“ durchgeführt werden, um diese auf der neuen Lokomotionstherapie-Hardware gängig zu machen. Aufgrund der entstandenen Komplexität, durch parallel geführte Code-Abwicklung, dem Programmfluss zu folgen, erwies sich diese Portierung als besonders zeitintensiv. Erschwerend zum Debuggen kam das sog. Node Guarding hinzu. Dabei wird die Kommunikation des Masters (Laptop) mit den Slaves (Motion Controller) überwacht. Ein Kommunikationsausfall würde zwangsläufig zum Verlust weiterer Sicherheitsmechanismen führen. Deshalb wird ein Zeitfenster definiert, welches bei Überschreitung einen Fehler generiert, welcher den Therapievorgang stoppt. Während das Node Guarding ein Sicherheitsfeature bei Therapieausführung darstellt, erschwert es das Debuggen ungemein. LabView bietet die Möglichkeit, den Code künstlich zu verlangsamen und den Datenfluss zu beobachten. Da es keinen definierten Ein/Aus-Schalter für das Node Guarding gibt, konnte diese Möglichkeit, ohne größere Umgestaltung des Programms, nur begrenzt eingesetzt werden.

### 5.2 Dokumentation und Verifizierung

Der Aufwand der Dokumentation der Software-Entwicklung hat ein beträchtliches Ausmaß angenommen. Bereits kleinste Änderungen führen zu neuen Versionen zahlreicher Dokumente. Ebenso waren der Testcode und die zugehörigen Testprotokolle für das Verifizieren zu aktualisieren. Um diesen Vorgang zu beschleunigen, könnte der Einsatz eines automatisierten Test-Codes für die einzelnen Software-Komponenten in Betracht gezogen werden, welcher autonom die einzelnen Testschritte durchführt und das Resultat präsentiert. Denkbar wäre auch, direkt ein Testprotokoll-Template mit den Ergebnissen zu befüllen und dieses anschließend mit korrekter Version im Dateinamen zu archivieren. Ein zusätzliches potentiell Fehlerpotential ist durch die zahlreichen Querverweise zwischen den Dokumenten gegeben. Um die Dokumentation übersichtlicher zu gestalten, könnte nachträglich ein Versionsverwaltungssystem, wie es im Zuge des Re-Designs für die Software konzipiert wurde, eingeführt werden. Der große Vorteil bestünde darin, dass zwischen den einzelnen Dokument-Versionen gewechselt werden könnte, ohne dabei

redundante Dokumente mit sich unterscheidenden Dateinamen im Projekt-Ordner aufbewahren zu müssen.

### **5.3 Versionsverwaltung und Backup**

Wie zuvor bereits ausgeführt, wurde der Quellcode der Software im Zuge des Re-Designs mit einem Versionsverwaltungssystem versehen. Verwendet wurde TortoiseSVN, ein freier Client für den Versionsverwaltungs-Dienst Subversion. Durch die Versionierung können unerwünschte Änderungen rasch und praktikabel rückgängig gemacht werden. Während der Entwicklung stellte sich heraus, dass das Projekt mit der 64-Bit Version anstatt der ursprünglichen 32-Bit Version von LabView geöffnet wurde. Wurden nun VIs gespeichert, so hat LabView selbstständig eine Konvertierung derselben vorgenommen. Es kam zu einem Fehler, der zu einem nicht wieder verwendbaren Frontpanel der Software führte. Ohne die Versionsverwaltung hätte eine gänzliche Neuerstellung der betroffenen VIs durchgeführt werden müssen. Einen zusätzlichen Vorteil der Datensicherung würde die Auslagerung des Repositorys (das digitale Archiv zur Versionsverwaltung) auf einem externen Datenträger oder Server bieten.

### **5.4 Gewichtsmessung**

Eine Gewichtsmessung des Patienten konnte erfolgreich entwickelt werden. Der Fokus lag dabei auf einer möglichst exakten Messung innerhalb des für das Lokomotionsgerät zulässigen Gewichtsbereichs. Dennoch wird nochmals darauf hingewiesen, dass die Waage keine Kalibrierung im herkömmlichen Sinne aufweist und deshalb nur in Kombination mit einer medizinischen Waage zum Einsatz kommen sollte.

### **5.5 Automatisiertes Laufband**

Ein automatisiertes Laufband, welches den physiologischen Gang zusätzlich unterstützt, wurde im Zuge der Arbeit ebenfalls implementiert. Größte Herausforderung diesbezüglich war, das Laufband mit den bewegten Orthesen zu synchronisieren. Vorab mussten Berechnungen erstellt werden, um einerseits den richtigen Antrieb zu wählen und in weiterer Folge von der Kadenz auf die Fördergeschwindigkeit des Laufbands schließen zu können. Während im eingeschwungenen Zustand beide Systeme problemlos zu synchronisieren waren, mussten die Anfahr- und Abbremsphase genauer analysiert werden. Hierfür wurde mittels Rampenfunktion die Betriebsspannung des Motors so variiert, dass die Komponenten einen möglichst gleichmäßigen Lauf aufweisen. Eine weitere Variable bei der Abstimmung stellte die Schrittlänge dar. Bei gleichbleibender Kadenz muss sich das Laufband kontinuierlich schneller bewegen, je länger die Beine des Patienten sind. Diesbezüglich wird in der Betriebsanleitung auf das höhenverstellbare Laufband hingewiesen. So sollte der Fußkontakt erst direkt nach Beginn der Therapie hergestellt werden, um eine Blockade zu

Therapiestart oder eine inkorrekte Geschwindigkeit aufgrund falscher Körpermaße zu verhindern.

## **5.6 Risikomanagement**

Die identifizierten Risiken des Re-Designs konnten durch Anwendung geeigneter Risiko-Kontrollmaßnahmen auf ein akzeptables Niveau minimiert werden. Da die Risiken der ursprünglichen Lokomotionstherapie-Software vorab bereits ein dermaßen niedriges Niveau einnahmen, konnte das Re-Design im Allgemeinen keine weitere Minderung der vorhandenen Risiken mehr erwirken. Der Großteil der Änderungen der Software zielte hauptsächlich auf deren Gebrauchstauglichkeit ab oder waren Korrekturen von Fehlern, welche keine Auswirkungen auf die Risiken hatten. Einen weiteren Grund, dass die Anzahl der Risiken nach dem Re-Design nahezu gleichgeblieben ist, stellen behobene Fehler dar, welche definitiv nicht mehr auftreten können (Eintrittswahrscheinlichkeit = 0 %) und demnach im Risikomanagement keine Berücksichtigung mehr erfahren. Da keine unvermeidbaren Risiken jemals eintreten können bzw. die Restrisiken als akzeptabel bewertet wurden, stellt das entwickelte Lokomotionsgerät inklusive dem Re-Design der Software eine sichere Therapiemöglichkeit für Patienten mit infantiler Zerebralparese dar.

## **5.7 Verbesserungspotential**

Problematisch zu betrachten ist, dass der Entwickler der Software laut EN 62304 gleichzeitig auch der Tester jener sein darf. Eine Erweiterung des Entwicklungspersonals oder zumindest eine Trennung zwischen Entwickler und Tester würde das Fehlerpotential maßgeblich einschränken. Eigene Fehler werden einfach leichter übersehen. Die Software-Entwicklung der Lokomotionstherapie V 0.3 ist erfolgreich beendet, jedoch sollte eine Weiterentwicklung des gesamten Projekts ins Auge gefasst werden. Zusätzlich zu den bereits für die nächste Softwareversion erkannten Verbesserungspotentialen, siehe Anhang Unterpunkt E, erscheint ein erneuter Gebrauchstauglichkeitstest als empfehlenswert.

## 6 Schlussfolgerung

Die vorhandene Software zur Steuerung eines Lokomotionsgerätes für Kleinkinder konnte durch die vorliegende Arbeit verbessert und sicherer gemacht werden. Die Forderungen der Medizingeräte-Software Norm EN 62304 wurden, mit Ausnahme des Qualitätsmanagement-Systems und dem Software-Wartungsprozess erfolgreich umgesetzt. Dazu wurde die Software umfassend dokumentiert bzw. verifiziert.

Das Lokomotionsgerät bzw. dessen Software wurden um gewünschte Funktionen, unter anderem einer Gewichtsmessung, sowie eines automatisierten Laufbands erweitert. Die Software bietet zusätzlich neue Möglichkeiten, wie zum Beispiel das Exportieren der Patientenakte. Ebenfalls sind die Ergebnisse des Usability-Tests in den Entwicklungsprozess eingeflossen. Somit konnte insgesamt eine Vielzahl an Fehlern korrigiert werden.

Die erneute Durchführung einer Risikoanalyse und die Ableitung von geeigneten Abhilfemaßnahmen reduzierten die neuen Risiken des Re-Designs erheblich, während die vorhandenen Risiken bereits vorab auf einem derart niedrigen Niveau waren, sodass das Re-Design an dieser Stelle keine Minderung mehr erwirken konnte. Der medizinische Nutzen überwiegt dem Risiko in hohem Maße. Die Restrisiken sind als akzeptabel zu bewerten und führten somit zur Freigabe der Lokomotionstherapie-Software (Version 0.3) entsprechend der Norm EN 62304.

Das Re-Design einer Ansteuerungssoftware eines Lokomotionsgerätes für Kleinkinder konnte erfolgreich abgeschlossen werden. Aus Sicht der Software-Entwicklung und des Risikomanagements kann das Gerät in der Klinik, im Sinne des Nutzens für die Patienten, eingesetzt werden.



## 7 Literatur

- [1] L. Döderlein, *Infantile Zerebralparese*, Berlin Heidelberg: Springer, 2015.
- [2] J. Perry, *Gait Analysis: Normal and Pathological Function*, Thorofare, New Jersey: SLACK, 1992.
- [3] K. Götz-Neumann, *Gehen verstehen*, Stuttgart - New York: Georg Thieme Verlag, 2006.
- [4] D. Beckers und J. Deckers, *Ganganalyse und Gangschulung: Therapeutische Strategien für die Praxis*, Hoensbroek: Springer, 1997.
- [5] M. Oskoui, F. Coutinho, J. Dykeman, N. Jetté und T. Pringsheim, „An update on the prevalence of cerebral palsy: a systematic review and meta-analysis,“ *Dev Med Child Neurol* 55, p. 509–519, 2013.
- [6] L. Mutch, E. Alberman, B. Hagberg, K. Kodama und M. Velickovic Perat, „Cerebral Palsy Epidemiology: Where are We Now and Where are We Going?,“ *Dev Med Child Neurol* 34, p. 547–551, 1992.
- [7] R. Palisano, P. Rosenbaum, S. Walter, D. Russell, E. Wood und B. Galuppi, „Development and reliability of a system to classify gross motor function in children with cerebral palsy,“ *Dev Med Child Neurol* 39, p. 214–223, 1997.
- [8] A. Kumar, *Effektivität und Sicherheit der Roboterunterstützten Gangorthese Lokomat bei Kindern und Jugendlichen mit zentralen Gangstörungen*, Ludwig-Maximilians-Universität zu München: Dissertation, 2009.
- [9] G. Colombo, *Entwicklung und Einsatz einer automatisierten Lokomotionstherapie für Querschnittgelähmte*, ETH Zürich: Dissertation, 2001.
- [10] *EN 62304:2006 + A1:2015 Medizingeräte-Software - Software-Lebenszyklus-Prozesse*, 2015.
- [11] *Medizinprodukterichtlinie 93/42/EWG*, 1993.
- [12] M. Kalkgruber, *Entwicklung einer Software für ein Lokomotionsgerät*, Technische Universität Graz: Masterarbeit, 2016.

- [13] S. Langthaler, *Beurteilung der Gebrauchstauglichkeit eines Lokomotionsgerätes für Kleinkinder*, Technische Universität Graz: Masterarbeit, 2016.
- [14] ÖNORM EN ISO 14971:2013-03 *Medizinprodukte - Anwendung des Risikomanagements auf Medizinprodukte*, 2013.
- [15] N. Leitgeb, *Sicherheit von Medizingeräten: Recht-Risiko-Chancen*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2015.
- [16] „RICHTLINIE 2006/42/EG DES EUROPÄISCHEN PARLAMENTS UND DES RATES vom 17. Mai 2006 über Maschinen und zur Änderung der Richtlinie 95/16/EG (Neufassung),“ 2006. [Online]. Available: <https://eur-lex.europa.eu/legal-content/DE/TXT/PDF/?uri=CELEX:32006L0042&from=DE>. [Zugriff am 15 10 2018].
- [17] A. Tilp, *Entwicklung eines Lokomotionsgerätes für Kleinkinder*, Technische Universität Graz: Masterarbeit, 2013.
- [18] S. Steinecker, *Re-Design eines Lokomotionsgerätes für Kleinkinder*, Technische Universität Graz: Masterarbeit, 2018.
- [19] M. Prim, *Konfigurationsmanagement\_Re-Design.docx*, 2018.
- [20] „Faulhaber Motion Controller MCDC 3006 S CF,“ [Online]. Available: [https://www.faulhaber.com/fileadmin/Import/Media/DE\\_MCDC3006S\\_V2-5\\_DFF.pdf](https://www.faulhaber.com/fileadmin/Import/Media/DE_MCDC3006S_V2-5_DFF.pdf). [Zugriff am 24 08 2018].
- [21] „Faulhaber Inkrementalencoder HEDS 5500 A,“ [Online]. Available: [https://www.faulhaber.com/fileadmin/Import/Media/DE\\_HEDS-5500\\_DFF.pdf](https://www.faulhaber.com/fileadmin/Import/Media/DE_HEDS-5500_DFF.pdf). [Zugriff am 07 09 2018].
- [22] „Faulhaber DC-Kleinstmotor 3863H024CR,“ [Online]. Available: [https://www.faulhaber.com/fileadmin/Import/Media/DE\\_3863\\_CR\\_DFF.pdf](https://www.faulhaber.com/fileadmin/Import/Media/DE_3863_CR_DFF.pdf). [Zugriff am 01 10 2018].
- [23] „Faulhaber Planetengetriebe 38A,“ [Online]. Available: [https://www.faulhaber.com/fileadmin/Import/Media/DE\\_38A\\_DFF.pdf](https://www.faulhaber.com/fileadmin/Import/Media/DE_38A_DFF.pdf). [Zugriff am 01 10 2018].
- [24] „Faulhaber DC-kleinstmotor 3863H024C,“ [Online]. Available: [https://www.micromo.com/media/pdfs/3863\\_C\\_DFF.pdf](https://www.micromo.com/media/pdfs/3863_C_DFF.pdf). [Zugriff am 01 10 2018].

- [25] „Faulhaber Planetengetriebe 38/2S,“ [Online]. Available: [https://www.faulhaber.com/fileadmin/Import/Media/DE\\_38-2S\\_FMM.pdf](https://www.faulhaber.com/fileadmin/Import/Media/DE_38-2S_FMM.pdf). [Zugriff am 01 10 2018].
- [26] „ELRA Schneckengetriebe SR25,“ [Online]. Available: <https://www.elra.at/images/produkte/getriebe/datenblaetter/SR25.pdf>. [Zugriff am 01 10 2018].
- [27] „IXXAT USB-to-CAN V2 compact,“ [Online]. Available: <https://www.ixxat.com/de/produkte/industrie-produkte/pc-interfaces/pc-can-interfaces/pc-can-interfaces-details/usb-to-can-v2>. [Zugriff am 24 08 2018].
- [28] W. Georgi und E. Metin, Einführung in LabVIEW, München: Carl Hanser Verlag, 2009.
- [29] B. Mütterlein, Handbuch für die Programmierung mit LabVIEW, Heidelberg: Spektrum Akademischer Verlag, 2009.
- [30] National Instruments, „SPECIFICATIONS NI USB-6001,“ [Online]. Available: <http://www.ni.com/pdf/manuals/374369a.pdf>. [Zugriff am 31 05 2018].
- [31] J. Hoffman, Taschenbuch der Messtechnik, Leipzig: Fachbuchverlag, 2015.
- [32] R. Parthier, Messtechnik: Grundlagen und Anwendungen der elektrischen Messtechnik, Mittweida, Deutschland: Springer Vieweg, 2016.
- [33] Texas Instruments, „INA 125 Instrumentation Amplifier,“ [Online]. Available: <http://www.ti.com/lit/ds/symlink/ina125.pdf>. [Zugriff am 04 06 2018].
- [34] *EN 60601-1:2014 : Medizinische elektrische Geräte - Teil 1: Allgemeine Festlegungen für die Sicherheit einschließlich der wesentlichen Leistungsmerkmale*, 2014.
- [35] „Schrack Installationsschutz 25A, 3S+1Ö, 230VAC 2TE,“ [Online]. Available: <https://www.schrack.at/shop/steuer-regeltechnik/schuetze-thermische-ueberlastrelais-motorschutzschalter/installationsschuetze-ac-1/installationsschuetze-r-ac-1-ac-spule/installationsschuetz-25a-3-schliesser-1-oeffner-230v-ac-bz326463.html>. [Zugriff am 23 08 2018].
- [36] „f-pro Fußschalter,“ [Online]. Available: <https://www.f-pro.de/usb-fussschalter>; <http://www.cleware-shop.de>. [Zugriff am 05 07 2018].

- [37] S. T. Moore, H. G. MacDougall, J.-M. Gracies, H. S. Cohen und W. G. Ondo, „Long-term monitoring of gait in Parkinson’s disease,“ *Gait & Posture* 26, pp. 200-207, 2007.
- [38] „Interroll EC310,“ [Online]. Available:  
[https://www.interroll.de/fileadmin/products/de/Resources\\_pdf\\_568810251.pdf](https://www.interroll.de/fileadmin/products/de/Resources_pdf_568810251.pdf).  
[Zugriff am 10 09 2018].
- [39] „Interroll DriveControl 20-54,“ [Online]. Available:  
[https://www.interroll.com/fileadmin/user\\_upload/Downloads\\_\\_PDF\\_/User\\_Manuals/Controls/DriveControl/DriveControl\\_20-54\\_2014\\_DE\\_2.pdf](https://www.interroll.com/fileadmin/user_upload/Downloads__PDF_/User_Manuals/Controls/DriveControl/DriveControl_20-54_2014_DE_2.pdf). [Zugriff am 12 10 2018].
- [40] M. Prim, *RA-V1\_Re-Design*, 2018.
- [41] M. Prim, *RA-V2\_Re-Design.docx*, 2018.
- [42] M. Prim, *RA-V2- Risikobeherrschungsmaßnahmen\_Re-Design.docx*, 2018.
- [43] M. Prim, *System-Anforderungen-V-1.docx*, 2018.
- [44] M. Prim, *Software-Anforderungen-V-0.3*, 2018.
- [45] M. Prim, *Traceability-V-0.3*, 2018.
- [46] M. Prim, *Software-Architektur-Hauptdokument-V-0.1.docx*, 2018.
- [47] M. Prim, *Bedienungsanleitung Software-V-0.3.docx*, 2018.
- [48] M. Prim, *Detail-Design-Hauptdokument-V-0.3.docx*, 2018.
- [49] M. Prim, *Implementieren-Verifizieren-Integrieren-V-0.3.docx*, 2018.
- [50] M. Prim, *SoftwareSystem-Verifizierung-V-0.1.docx*, 2019.
- [51] M. Svehlik, Ganganalysedaten- Ganglabor Graz.
- [52] T. Öberg, A. Karsznia und K. Öberg, „Joint angle parameters in gait : Reference data for normal subjects, 10-79 years of age,“ *Journal of Rehabilitation Research and Development Vol . 31*, p. 199—213, 1994.

# 8 Anhang

## A. Weitere Abbildungen

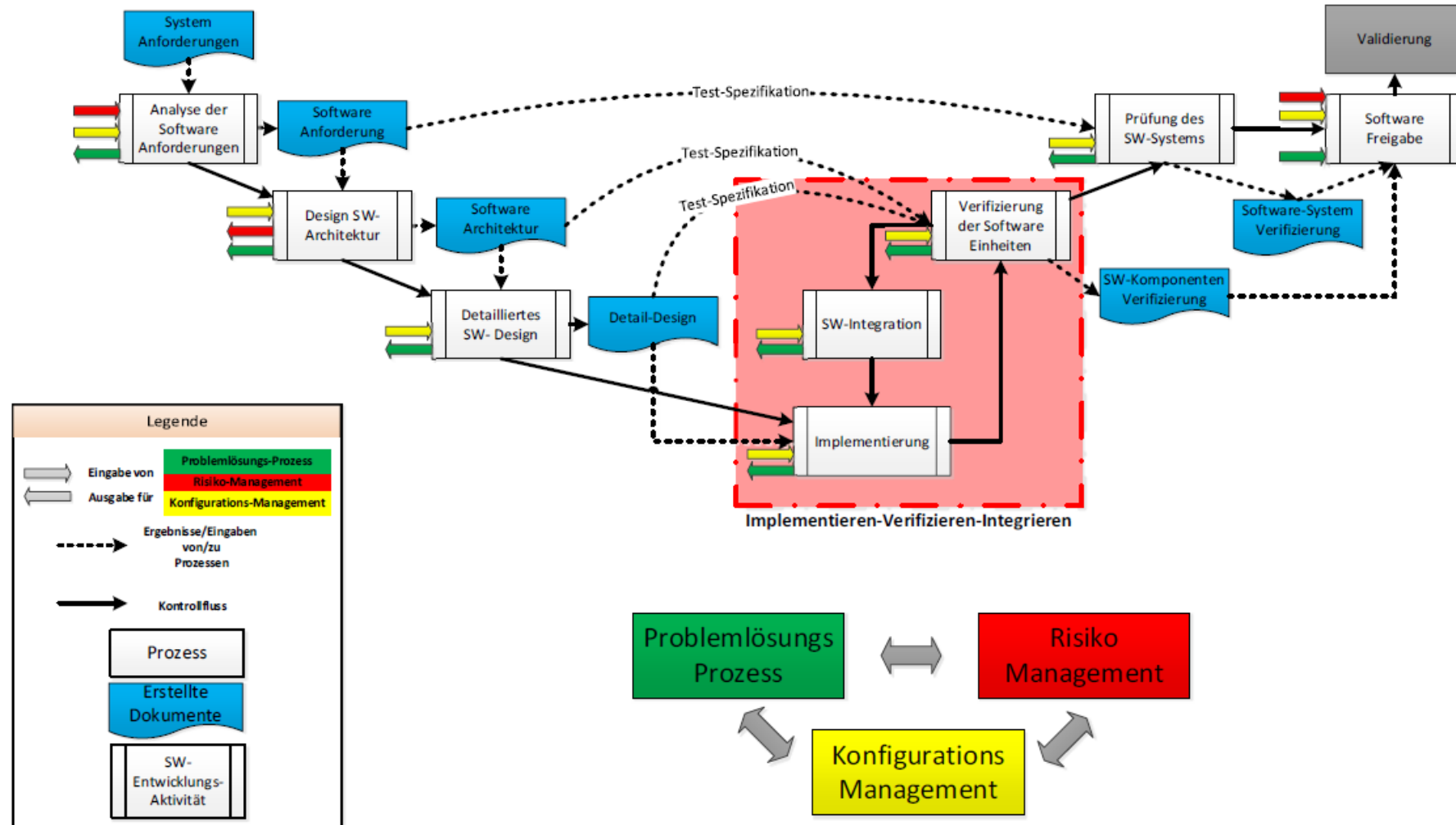


Abbildung 62: V-Modell des Software-Entwicklungs-Prozesses [12]

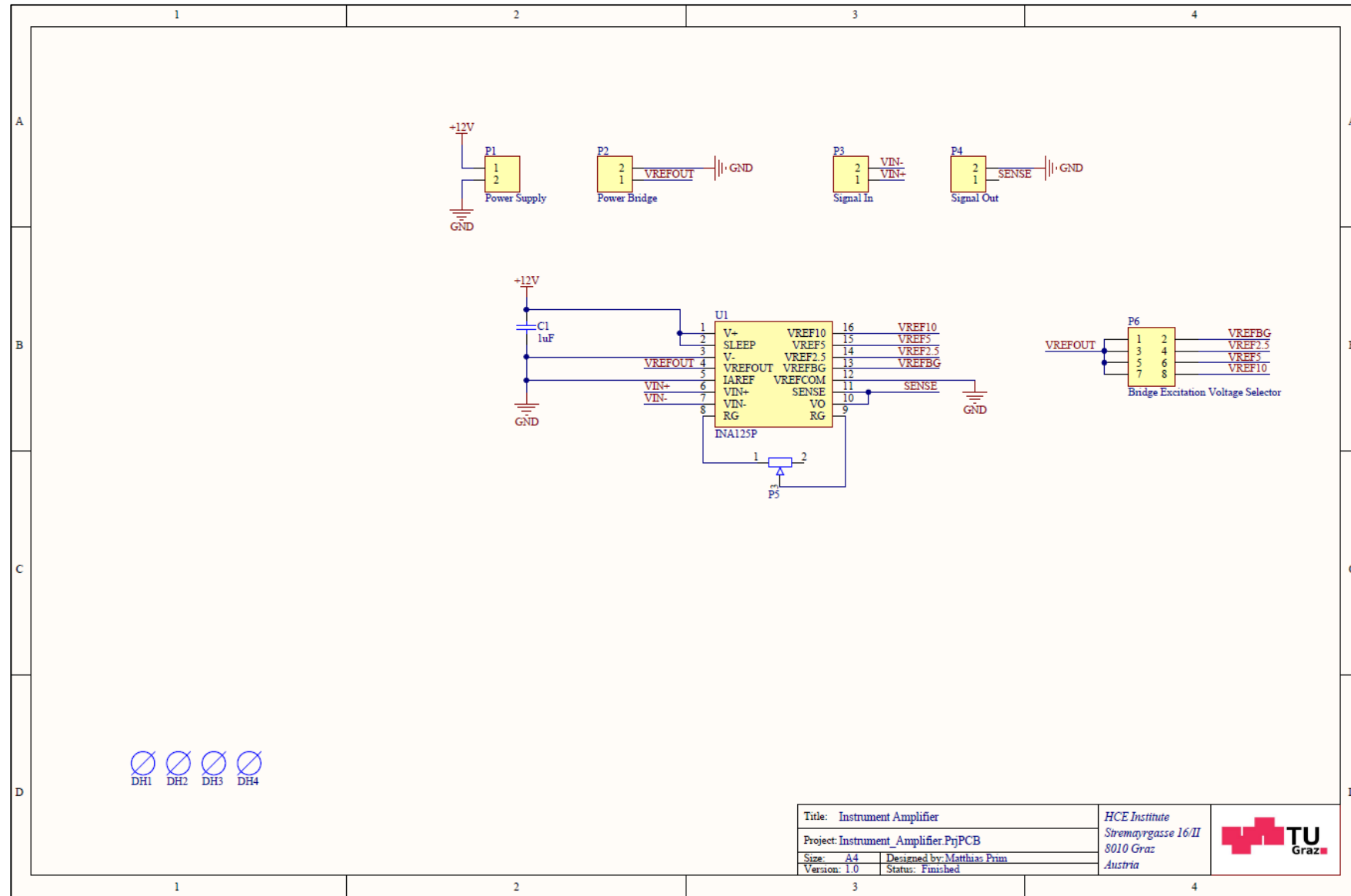


Abbildung 63: Konstruktionsplan Instrument Amplifier Board

A-Struktur-V-0.1.vsd

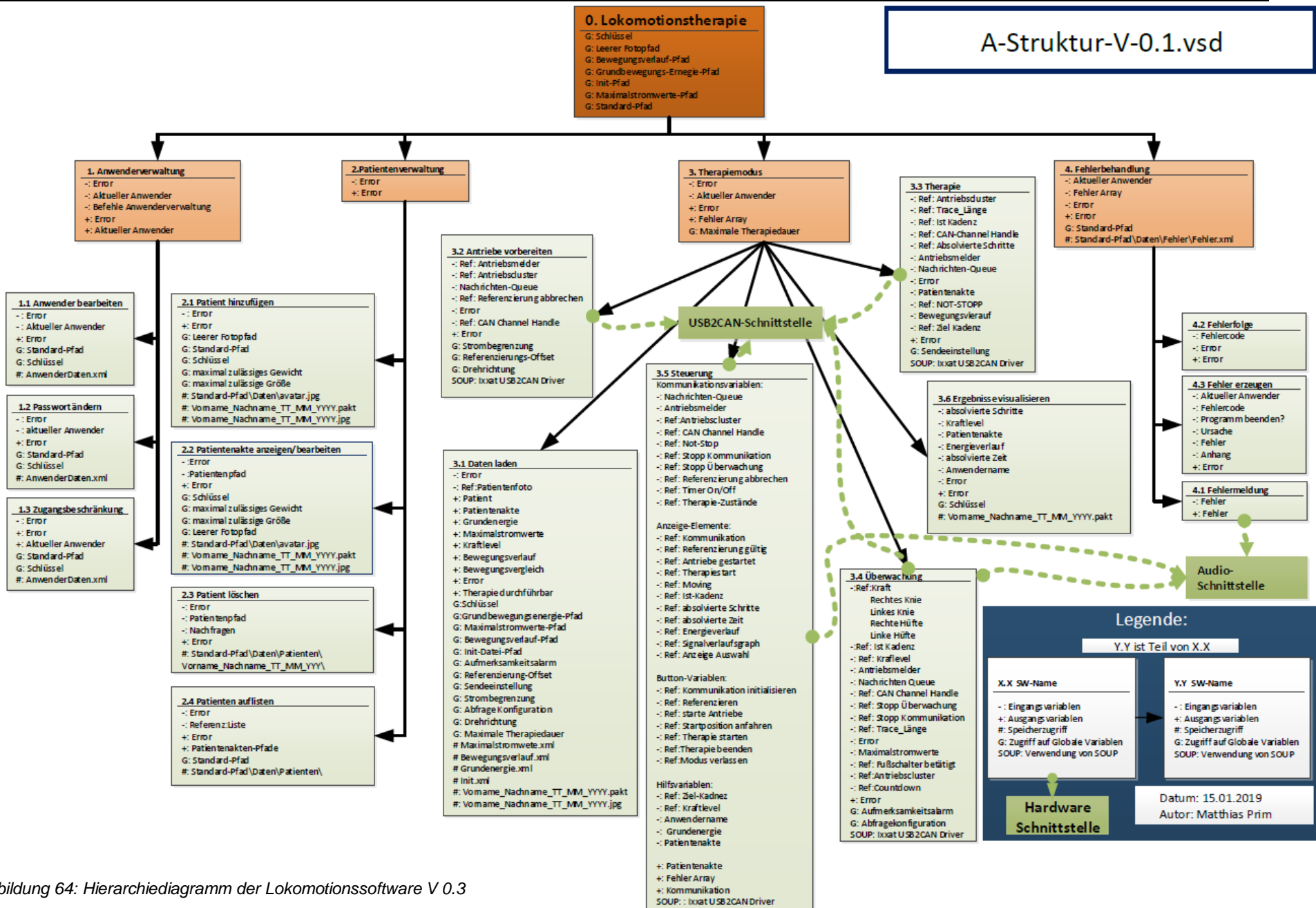


Abbildung 64: Hierarchiediagramm der Lokomotionssoftware V 0.3

## B. Berechnung der Rollerdrive Fördergeschwindigkeit

### B.1 Minimale Fördergeschwindigkeit

Da der Rollerdrive einen Durchmesser von  $d = 50 \text{ mm}$  aufweist, errechnet sich der benötigte Umfang  $U$  zu:

$$U = d \cdot \pi \text{ [m]} \quad (1)$$

$$U = 0,05 \cdot \pi \quad (2)$$

$$U \approx 0,157 \text{ m} \quad (3)$$

Ab 2,3 Volt beginnt sich der Motor im Inneren der Rollerdrive mit 305 U/min zu drehen. Um die äußere Drehzahl zu bestimmen, muss die Getriebeuntersetzung eingerechnet werden. Das Untersetzungsverhältnis ( $i$ ) ist das Verhältnis der inneren Drehzahl ( $n_1$ ) zur äußeren ( $n_2$ ) und wurde für den benötigten Geschwindigkeitsbereich mit 16:1 gewählt:

$$i = \frac{n_1}{n_2} \text{ []} \quad (4)$$

$$i = \frac{16}{1} \quad (5)$$

Um die äußere Drehzahl des Rollerdrives ( $n_2$ ) zu berechnen, muss die Motordrehzahl im Inneren ( $n_1$ ) durch die Untersetzung ( $i$ ) dividiert werden:

$$n_2 = \frac{n_1}{i} \left[ \frac{1}{\text{min}} \right] \quad (6)$$

$$n_2 = \frac{305}{16} \frac{1}{\text{min}} \quad (7)$$

$$n_2 \approx 19,06 \frac{1}{\text{min}} \quad (8)$$

Die Fördergeschwindigkeit ( $v$ ) ergibt sich aus der Multiplikation der äußeren Rollendrehzahl ( $n_2$ ) mit dem Umfang ( $U$ ):

$$v = n_2 \cdot U \left[ \frac{\text{m}}{\text{s}} \right] \quad (9)$$

$$v \approx 19,06 \frac{1}{\text{min}} \cdot 0,157 \text{ m} \quad (10)$$

$$v \approx 2,99 \frac{\text{m}}{\text{min}} \quad (11)$$



In SI-Einheiten ausgedrückt ergibt sich für die Fördergeschwindigkeit ( $v$ ):

$$v \approx 2,99 \frac{m}{min} : 60 \approx 0,05 \frac{m}{s} \quad (12)$$

## B.2 Maximale Fördergeschwindigkeit

Ab 10 Volt dreht sich der Motor im inneren der Rollerdrive mit der maximalen Drehzahl von 6000 U/min. Um die äußere Drehzahl zu bestimmen, muss erneut die Getriebeuntersetzung eingerechnet werden. Das Untersetzungsverhältnis wurde zuvor mit 16:1 gewählt:

$$i = \frac{n_1}{n_2} \quad (13)$$

$$i = \frac{16}{1} \quad (14)$$

Um die äußere Drehzahl des Rollerdrives ( $n_2$ ) zu berechnen, muss die Motordrehzahl im Inneren ( $n_1$ ) durch die Untersetzung ( $i$ ) dividiert werden.

$$n_2 = \frac{n_1}{i} \left[ \frac{1}{min} \right] \quad (15)$$

$$n_2 = \frac{6000}{16} \frac{1}{min} \quad (16)$$

$$n_2 = 375 \frac{1}{min} \quad (17)$$

Die Fördergeschwindigkeit ( $v$ ) ergibt sich aus der Multiplikation der äußeren Rollendrehzahl ( $n_2$ ) mit dem Umfang ( $U$ ):

$$v = n_2 \cdot U \left[ \frac{m}{s} \right] \quad (18)$$

$$v \approx 375 \frac{1}{min} \cdot 0,157m \quad (19)$$

$$v \approx 58,88 \frac{m}{min} \quad (20)$$

In SI-Einheiten ausgedrückt ergibt sich für die Fördergeschwindigkeit ( $v$ ):

$$v \approx 58,88 \frac{m}{min} : 60 \approx 0,98 \frac{m}{s} \quad (21)$$

### B.3 Fördergeschwindigkeit in Abhängigkeit der Spannung

Ab 2,3 Volt beginnt sich der Motor im Inneren der Rollerdrive mit 305 U/min zu drehen. Die Steigungsrate (incline rate) ab 2,3 Volt beträgt 740 U/min pro Volt. Über 10V erreicht der Motor im Inneren die maximale Drehzahl von 6000 U/min.

Als Beispiel wird die Drehzahl bei  $U = 6\text{ V}$  berechnet. Ausschlaggebend für die Fördergeschwindigkeit ist nur die Spannungsdifferenz ( $\Delta U$ ) in Bezug auf die Startspannung:

$$\Delta U = U - 2,3V [V] \quad (22)$$

$$\Delta U = 6V - 2,3V \quad (23)$$

$$\Delta U = 3,7V \quad (24)$$

Die innere Drehzahl des Rollerdrives ( $n_1$ ) errechnet sich durch Multiplikation der Spannungsdifferenz ( $\Delta U$ ) mit den 740 U/minV und anschließender Addition der Startdrehzahl von 305 U/min.

$$n_1 = \Delta U * 740 \frac{1}{V \cdot \text{min}} + 305 \frac{1}{\text{min}} \left[ \frac{1}{\text{min}} \right] \quad (25)$$

$$n_1 = 3,7V * 740 \frac{1}{V \cdot \text{min}} + 305 \frac{1}{\text{min}} \quad (26)$$

$$n_1 = 2738 \frac{1}{\text{min}} + 305 \frac{1}{\text{min}} \quad (27)$$

$$n_1 = 3043 \frac{1}{\text{min}} \quad (28)$$

Um die äußere Drehzahl des Rollerdrives ( $n_2$ ) zu berechnen, muss die Motordrehzahl im Inneren ( $n_1$ ) durch das Untersetzungsverhältnis ( $i$ ) dividiert werden:

$$n_2 = \frac{n_1}{i} \left[ \frac{1}{\text{min}} \right] \quad (29)$$

$$n_2 = \frac{3043}{16} \frac{1}{\text{min}} \quad (30)$$

$$n_2 = 190 \frac{1}{\text{min}} \quad (31)$$

Die Fördergeschwindigkeit ( $v$ ) ergibt sich aus der Multiplikation der äußeren Rollendrehzahl ( $n_2$ ) mit dem Umfang:

$$v = n_2 \cdot \text{Umfang} \left[ \frac{m}{s} \right] \quad (18)$$

$$v \approx 190 \frac{1}{\text{min}} \cdot 0,157m \quad (19)$$

$$v \approx 29,83 \frac{m}{\text{min}} \quad (20)$$

In SI-Einheiten ausgedrückt ergibt sich für die Fördergeschwindigkeit ( $v$ ):

$$v \approx 29,83 \frac{m}{\text{min}} : 60 \approx 0,50 \frac{m}{s} \quad (21)$$

## C. Usability Änderungen

Tabelle 22: Usability Defizite und Abhilfemaßnahmen der Lokomotionstherapie-Software Version 0.2

Nr.	Defizit / Problem	Ursache	Abhilfemaßnahme	Durchgeführt
1	Patientendaten auslesen	Begriffswahl „Höhe“ nicht eindeutig	Umbenennung in „Höhe Orthesenhalterung“	JA
2	Patientenakte bearbeiten	Textfelder (Tabulator)	Überarbeitung der Textfelder	JA
3	Erkennen der Gewichtsüberschreitung	Warnleuchte	Warnleuchte vergrößern	JA
4	Erkennen der Gewichtsüberschreitung	Fehlender Hinweis auf Konsequenzen durch Gewichtsüberschreitung	Hinweis in Statusmeldung „Therapie nicht durchführbar“	JA
5	Patientenakte anlegen	Fehlende Eingabe von Daten	Hinweis in Fehlermeldung auf fehlende Daten	JA
6	Patientenakte anlegen	Tabulator Reihenfolge	Tabulatorreihenfolge anpassen	JA
7	Patientenakte anlegen	Patientenfoto muss zwingend eingefügt werden	Steht kein Patientenfoto zur Verfügung, soll ein Avatar eingefügt werden	JA
8	Absturz nach Abbruch über Not-Stopp	Nicht identifiziert	keine	JA
9	Erneutes Öffnen des Therapiemodus nach Beenden	Nicht identifiziert	keine	JA
10	Fehlende Funktion zum Ausdrucken der Therapiedaten	Funktion nicht vorhanden	Funktion ergänzen	JA
11	Fehlende Anzeige des Kraftlevels im Therapiemodus	Funktion nicht vorhanden	Funktion ergänzen	JA
12	Änderung des Kraftlevels während der Therapie nicht möglich	Funktion nicht vorhanden	Funktion ergänzen	NEIN <sup>13</sup>
13	Fehlende Korrekturmöglichkeit des Kraftlevels bei Falscheingabe	Funktion nicht vorhanden	Funktion ergänzen	JA
14	Anzeigen der Patienteninformation in Therapiemodus	Funktion nicht vorhanden	Funktion ergänzen	JA
15	Übernahme und/oder Anzeige von eingestellten Parametern nach Abbruch oder Pause der Therapie	Funktion nicht vorhanden	Funktion ergänzen	JA
16	Messung und Anzeige der Kraft des Patienten	Funktion nicht vorhanden	Funktion ergänzen	JA

<sup>13</sup> Anmerkung: Die Funktion wurde aus Sicherheitsgründen nicht umgesetzt. Das Kraftlevel darf je Therapieeinheit und nach Abbruch aufgrund von zu hohem Drehmoment jeweils nur um eine Stufe erhöht werden.

Nr.	Defizit / Problem	Ursache	Abhilfemaßnahme	Durchgeführt
17	Feststellung der Gewichtsbelastung	Funktion nicht vorhanden	Funktion ergänzen	JA
18	Bearbeitung der Diagramme	Icons-Symbole nicht selbsterklärend	Mouse Over Funktion	JA
19	Fehlermeldung bei fehlender Initialisierung der Kommunikation	Keine Hilfestellung zur Fehlerbehebung in Fehlermeldung	Angabe von möglichen Ursachen in Fehlermeldung	JA
20	Checkliste Begriffswahl	Begriffswahl nicht eindeutig	Änderung der Begriffsdefinition	JA
21	Checkliste Aufbau	Zu viele Teilschritte	Automatisierung von Teilschritten	JA
22	Struktur Therapiemodus	Größe der Checkliste zu klein, Diagramm zu groß	Checkliste größer ausführen, Diagramm kleiner	JA
23	Not-Stopp	Button aktiv vor Therapiestart	Funktion ausblenden	JA
24	Not-Stopp	Aktivierung vor Therapiestart führt zu Initialisierung der Kommunikation	Funktion ausblenden	JA
25	Not-Stopp	Probanden erkennen Not-Stopp nicht	Farbgebung und Kennzeichnung	JA
26	Not-Aus lösen	Fehlermeldung enthält keinen Hinweis auf Lösen des Not-Aus	Hinweis in Fehlermeldung	JA
27	Schrittgeschwindigkeit einstellen	Automatisches Rücksetzen bei frühzeitigem Einstellen	Funktion ausblenden	JA
28	Funktion „Pause/Beenden“	Probanden erkennen Funktion nicht	Farbgebung und Kennzeichnung	JA
29	Eingaben	Bestätigung durch Enter nicht möglich	Bestätigung durch Drücken von Enter implementieren	JA
30	Textfelder	Vorgaben in Textfelder werden bei Eingabe nicht automatisch gelöscht	Vorgaben bei Eintrag automatisch löschen	JA
31	Geräteeinstellungen treffen	Fehlender Hinweis auf Orthesenwechsel	Funktion ergänzen	NEIN <sup>14</sup>
32	Geräteeinstellungen treffen	Fehlende Unterstützung bei Treffen der Geräteeinstellungen	Informationen über Patientenabmessungen in Patientenakte ergänzen	JA
33	Geräteeinstellungen treffen	Fehlende Unterstützung bei Treffen der Geräteeinstellungen	Eingabe und/oder automatische Berechnung der Geräteeinstellungen anhand der Patientenabmessungen	NEIN <sup>15</sup>

<sup>14</sup> Anmerkung: Die Orthesen beim neuen Lokomotionsgerät sind mittels Klettverschlusses variabel an die Patientenabmessungen anzupassen. Sie sind starr mit der Konstruktion verbunden und ein Wechsel ist erst bei größerer Abnutzung vorgesehen.

<sup>15</sup> Anmerkung: Nach Rücksprache war damit eine Anzeige, auf welche Länge die Orthesen vom Therapeuten einzustellen sind, gemeint. Da das neue Lokomotionsgerät nur ein fixes Raster zur Längeneinstellung hat und keine exakte Skala, wäre eine Vorgabe nutzlos und deshalb wurde von der Implementierung vorerst Abstand genommen.

## D. Durchgeführte Änderungen an der Lokomotionssoftware

Die nachfolgenden Aufzählungen beinhalten sämtliche Änderungen, welche während der Entwicklung von **Lokomotionstherapie-V-0.3** durchgeführt wurden. Die Versionsnummern der VI-Namen wurden inkrementiert und entsprechen bereits der aktuellen Software.

### Allgemeine Änderungen:

Die allgemeinen Änderungen dienen rein der Optik, zur besseren Lesbarkeit des Codes bzw. zur Steigerung der Usability. Die Funktionalität wird nicht beeinflusst bzw. verändert.

- Sämtliche Frontpanele, welche eine User Interaktion benötigen, wurden mit einem neuen Design versehen, welches besonders auf eine verbesserte Usability abzielt.
- Frontpanele, welche dem User nicht angezeigt werden, wurden aufgeräumt. Häufig waren Elemente außerhalb des sichtbaren Bereichs angeordnet.
- Sämtliche Blockdiagramme wurden vereinheitlicht, indem z. B. kleine Icons verwendet wurden, die gezogenen „Wires“ begradigt wurden bzw. wenn nötig die Symbole neu angeordnet wurden. Des Weiteren wurde versucht, den Code kompakter zu gestalten, um etwaiges Scrollen zu vermeiden.
- Die „Window Appearance“ wurde so eingestellt, dass in der herkömmlichen Entwicklung alle benötigten Elemente, insbesondere die Toolbar mit dem Abort-Knopf, angezeigt werden und im kompilierten Programm der Endanwender nur die für ihn zulässigen Buttons sieht.
- Die „Window Size“ beträgt nie mehr als 1366x768 Pixel, um auf dem eingesetzten Endgerät mit derselben Display-Auflösung nicht zu groß zu erscheinen.
- Die „Window Run-Time Position“ wurde, je nach Erfordernis, entweder zentriert oder maximiert.
- LabView unterscheidet bei Variablen zwischen Label und Caption. Während das Label die eigentliche Bezeichnung zum Programmieren ist, ist das Caption eine zusätzliche Beschriftung. Um die Anzeige im Block-Diagramm und auch im Hilfe-Fenster zu vereinheitlichen, sollte nur das Label verwendet werden. Das betrifft insbesondere die Error Cluster.

### Typedefs:

- Kraftlevel.ctl: neues Design
- Orthesenlängen.ctl: neues Design; Änderung zu „strict typedef“; Der nicht eindeutige Begriff „Höhe“ wurde durch „Höhe Orthesenhalterung“ ersetzt. Orthesenlängen.ctl war ursprünglich dafür gedacht, die Einstellungen der Orthesen der vorangegangenen Therapie zu speichern und sie somit leichter am Lokomotionsgerät wiederherzustellen können. Da eine eindeutige Skala am Gerät fehlt, wurde das Control umfunktioniert um die Beinlängen anstatt den Orthesenlängen zu speichern. Das Control wurde verkleinert, sodass die Anzeige „Höhe Orthesenhalterung“ nicht mehr sichtbar ist und nur noch die Ober- und Unterschenkellänge eingestellt werden kann. Der Name des Typedefs wurde aufgrund der gleichen Eingabemöglichkeiten vorerst beibehalten.
- Therapiedetail.ctl: neues Design

**Virtual Instruments (VIs):****0.3 Sort 2D Array-V-0.vi**

- Dieses neu erstellte VI ermöglicht das spaltenweise Sortieren von beliebig großen 2D-String-Arrays. Als Eingang muss das Array, die Spaltennummer nach der sortiert werden soll und die Reihenfolge (auf- oder absteigend) angegeben werden. Ausgänge stellen das sortierte Array, die einzelne sortierte Spalte und ein Index-Nummern-Array der ehemaligen Elemente dar. Das Index-Array dient zum Sortieren weiterer Datentypen mit derselben Reihenfolge, wie es auch in „2 Patientenverwaltung“ zum Einsatz kommt.

**0.4 Select Printer-V-0.vi**

- Dieses neu erstellte VI zeigt eine GUI zur Auswahl einer der verfügbaren Drucker an. Der Drucker wird aus dem Dropdown-Menü ausgewählt. Durch Drücken von „Drucker auswählen“ wird der entsprechende Druckername in Form eines Strings retourniert. Durch Drücken von „Abbrechen“ wird „Kein Drucker gewählt“ in Form eines Strings retourniert.

**0.5 Not-Aus Überwachung-V-0.vi**

- Dieses neue VI überwacht, ob der Anwender den Not-Aus-Schalter gedrückt hat. Verwendet wird hierzu das USB-6001 DAQ-Device. Verbunden ist der 5V-Ausgang mit dem Analog In Channel Nr. 7, über einen Öffner des Schützes (Klemme 5 und 6). Wird nun der Not-Aus betätigt, kommt es zur Unterbrechung der Versorgung, welche detektiert wird. Gemessen wird ob die Spannung kleiner als 2,5 V ist, da dies der halben Versorgungsspannung entspricht. Eine Abfrage, ob die Spannung 0 Volt entspräche, wäre niemals zielführend, da bedingt durch äußere Störeinflüsse und dem Aufbau des USB- DAQ Devices immer Mikrovolt zu messen sind. Die Funktion des VIs ist rein visuell und hat keinen sicherheitstechnischen Aspekt. Die entsprechende Fehlermeldung, welche bei Betätigung generiert wird, weist auch auf das Lösen des Not-Aus-Schalters hin.

**0.6 Gewicht messen-V-0.vi**

- Dieses neue VI ermittelt das Patientengewicht. Verwendet wird hierzu das USB-6001 DAQ-Device. Verbunden ist der Analog In Channel Nr. 0 mit der Instrumentenverstärker-Platine. Eingestellt, für die Verstärkung, wurden 7,5V bei 15kg. Berechnet wird das Gewicht mit Hilfe der Geradengleichung  $y=kx + d \rightarrow x = (y-d)/k$ . Die Werte für d und k mussten zuvor mit bekannten Gewichten bestimmt werden. Retourniert wird das Gewicht in Gramm.

**0.Lokomotionstherapie-V-0.3.vi**

- Das VI „3.2.1.1 Reset USB-CAN“ wurde im Beenden-Case platziert, siehe „3.2.1.1 Reset USB-CAN-V-0.1.vi“.
- Der leere Fotopfad verweist nun auf „avatar.jpg“ anstatt auf „blank.jpg“. Dadurch ist in sämtlichen VIs, in denen das Patientenfoto angezeigt wird, deutlicher ersichtlich, wenn die aktuelle Patientenakte kein Foto aufweist.

- Die Flat Sequence Structure im Therapiemodus Case war hinfällig, da SubVIs immer zuerst auf alle Inputs warten, bis sie ausgeführt werden.
- Dadurch, dass VI „1.3 Zugangsbeschränkung“ zur Zugangsbeschränkung bei Programmstart als auch zum Wechseln des Therapeuten aus dem Hauptmenü heraus verwendet wird, musste eine Änderung durchgeführt werden. Der „Abbrechen“-Button ließ immer das Programm enden. Jetzt endet das Programm nur bei drücken von „Abbrechen“ zu Programmstart und nicht bei „Therapeut wechseln“. Es wird überprüft ob der Anwendername der retournierten Variable „Aktueller Anwender“ nicht leer ist, ansonsten endet das Programm. Eine Ausnahme stellt eine (unerlaubte) Änderung der Datei „AnwenderDaten.xml“ dar. Die Zugangsbeschränkung gibt einen Fehler und eine leere „Aktueller Anwender“-Variable zurück. In diesem Fall wird das Programm nicht sofort beendet, sondern erst nach Anzeige der Meldung, dass die Datei unsachgemäß manipuliert wurde.

### **1.1 Anwender bearbeiten-V-0.2.vi**

- Ein Doppelklick auf eine leere Zeile oder Spaltenüberschrift der Therapeutenliste öffnet „1.1.2 Anwender ändern“, ohne einen Benutzer ausgewählt zu haben. Dieses Fehlverhalten wurde durch eine Case-Structure unterbunden.
- Der Timeout-Case war leer und wurde deshalb entfernt.
- Das doppelte Build Path für den Pfad für die Datei „AnwenderDaten.xml“ war hinfällig und wurde durch ein einzelnes ersetzt.
- Die Anwenderliste stellte beim Erstaufruf die Benutzernamen unsortiert dar. Ab sofort erscheinen die User in alphabetischer Reihenfolge. Zusätzlich ist es nun möglich, auf- bzw. absteigend zu sortieren. Der neue „Benutzername“-Button hat einen Doppelpfeil bekommen, um die Möglichkeit noch deutlicher sichtbar zu machen. Sortiert werden die User mit dem neu angelegten VI „1.1.5 Anwender sortieren“.
- Die neue Eingangsvariable „Aktueller Anwender“ wird an VI „1.1.3 Anwender löschen“ weitergeleitet, um zu verhindern, dass der aktive User gelöscht werden kann.

#### **1.1.1 neuen Anwender erzeugen-V-0.2.vi**

- Die Return-Taste wurde dem „Benutzer hinzufügen“-Button für eine komfortablere Bedienung zugewiesen, die Escape-Taste dem „Abbrechen“-Button.
- Es war notwendig, wenn die Felder „Passwort“ bzw. „Passwort wiederholen“ übereinstimmten, aus dem Eingabefeld zu klicken, damit sich die Werte updaten und „Passwort korrekt“ aufleuchtet. Dies geschieht nun automatisch mit der Option „Update Value while Typing“. Außerdem konnte man auf diese Weise auch Benutzer erstellen, ohne dass das Passwort übereinstimmte.
- Der Button „Anwender hinzufügen“ wurde auf „Latch when released“ gesetzt, somit ist das Rücksetzen auf false obsolet.
- Der Button „Abbrechen“ wurde auf „Latch when released“ gesetzt, das Rücksetzen hat hier ohnehin gefehlt.
- Für die Eingabefelder „Benutzername“, „Passwort“ und „Passwort wiederholen“ wird mit der Option „Limit to Single Line“ ein ungewollter Zeilenumbruch bei der Eingabe verhindert.



### 1.1.2 Anwender ändern-V-0.2.vi

- Die Return-Taste wurde dem „Daten ändern“-Button für eine komfortablere Bedienung zugewiesen, die Escape-Taste dem „Abbrechen“-Button.
- Es war notwendig, wenn die Felder „Passwort“ bzw. „Passwort wiederholen“ übereinstimmten, aus dem Eingabefeld zu klicken, damit sich die Werte updaten und „Passwort korrekt“ aufleuchtet. Dies wird nun automatisch mit der Option „Update Value while Typing“ ausgeführt. Außerdem konnte man auf diese Weise auch Benutzer ändern, ohne dass das Passwort übereinstimmte.
- Für die Eingabefelder „Benutzername“, „Passwort“ und „Passwort wiederholen“ wird mit der Option „Limit to Single Line“ ein ungewollter Zeilenumbruch bei der Eingabe verhindert.

### 1.1.3 Anwender löschen-V-0.1.vi

- Es war möglich, alle Anwender aus „AnwenderDaten.xml“ zu löschen, somit ist man ohne Superuser ausgesperrt. Dies wurde abgeändert, sodass man den aktiven Benutzer nicht löschen kann und immer ein User-Account übrigbleibt. Meldung: „Der aktive Benutzer kann nicht gelöscht werden. Wechseln Sie zuerst auf einen anderen Benutzer, um diesen zu löschen.“

### 1.1.5 Anwender sortieren-V-0.vi

- Dieses VI wurde neu erstellt, um ein auf- und absteigendes Sortieren der Benutzernamen zu ermöglichen, siehe VI „1.1 Anwender bearbeiten“.

### 1.2 Passwort ändern-V-0.2.vi

- Beim boolschen Button „Abbrechen“ wurde der Ein-Text „ENDE“ entfernt, sodass immer nur „Abbrechen“ angezeigt wird.
- Es war möglich, durch Betätigen der Return-Taste eine neue Zeile im „neues Passwort“ bzw. „Passwort bestätigen“-Feld zu erzeugen. Dies wurde durch die Option „Limit to Single Line“ behoben.
- Es war notwendig, wenn die Felder „neues Passwort“ bzw. „Passwort bestätigen“ übereinstimmten, aus dem Eingabefeld zu klicken, damit die Werte sich updaten und ein korrektes Passwort erkannt wurde. Dies geschieht nun automatisch mit der Option „Update Value while Typing“.
- Die Return-Taste wurde dem „Ändern“-Button, für eine komfortablere Bedienung, zugewiesen, die Escape-Taste dem „Abbrechen“-Button.
- Es war möglich, ein Passwort zu speichern, obwohl die Felder „neues Passwort“ bzw. „Passwort bestätigen“ nicht übereinstimmten. Dazu wurde zuerst dieselbe Zeichenfolge in die Felder eingegeben, wodurch der „Ändern“-Button verfügbar wurde, danach wurde das neue Passwort verändert und dann direkt auf den Button „Ändern“ geklickt. Im schlimmsten Fall wäre somit ein unbekanntes Passwort gespeichert gewesen und der User-Account nicht mehr zu verwenden. Dies wurde mit der Option „Update Value While Typing“ verhindert.
- Der „Ändern“-Button wurde auf „Latch when released“ gesetzt, somit ist das Rücksetzen auf false hinfällig.
- Der „Abbrechen“-Button wurde auf „Latch when released“ gesetzt, somit ist das Rücksetzen auf false obsolet.

### 1.3 Zugangsbeschränkung-V-0.2.vi

- Die Willkommensmeldung wurde deaktiviert, da ohnehin im Hauptfenster der Lokomotionstherapie der Therapeutename bei erfolgreichem Login angezeigt wird. Stimmen Benutzername und Passwort nicht überein bzw. existiert der User nicht, wird weiterhin die entsprechende Meldung ausgegeben und das Zugangsbeschränkungsfenster bleibt weiterhin bestehen.
- Der Ein-Text des boolschen Button "Einloggen" wurde entfernt, sodass immer nur "Einloggen" angezeigt wird.
- Das doppelte Build Path für den Pfad für das File „AnwenderDaten.xml“ war hinfällig und wurde entfernt.
- Der boolsche Button "Beenden" hatte den Ein-Text "ENDE", welcher entfernt wurde, sodass immer nur "Beenden" angezeigt wird.
- Der boolsche Button "Beenden" wurde in "Abbrechen" umbenannt und beim Drücken endet die Lokomotionssoftware nicht mehr automatisch, sondern kehrt ins Hauptmenü zurück (Flat Sequence Structure war überflüssig in diesem Fall). Notwendig war dies, da das VI einerseits zu Programmstart, wie auch zum Wechsel des Therapeuten verwendet wird. Sollte der Therapeut doch nicht gewechselt werden, beendete sich allerdings die Software. Bei der Zugangsbeschränkung zu Beginn der Lokomotionssoftware wird bei drücken von „Abbrechen“ weiterhin die Software beendet.
- Die Return-Taste wurde dem „Einloggen“-Button für eine komfortablere Bedienung zugewiesen, die Escape-Taste dem „Abbrechen“-Button.
- Der „Einloggen“-Button ist erst nach Eingabe von „Benutzername“ und „Passwort“ anwählbar und nicht mehr ausgegraut.
- Für „Benutzername“ und „Passwort“ wurde "Update Value while Typing" aktiviert, um nach der Eingabe nicht aus dem jeweiligen Eingabefeld klicken zu müssen und der „Einloggen“-Button sofort anwählbar und nicht mehr ausgegraut ist.
- Für „Benutzername“ und „Passwort“ wurde „Limit to Single Line“ aktiviert, um Zeilenumbrüche bei der Eingabe zu verhindern.
- „Einloggen“ wurde auf "Latch when released" gesetzt, somit ist das Rücksetzen auf false hinfällig.
- „Abbrechen“ wurde auf "Latch when released" gesetzt, somit ist das Rücksetzen auf false nichtig.
- Die korrekte Tabulator-Reihenfolge wurde gesetzt.

### 2 Patientenverwaltung-V-0.3.vi

- Das VI erzeugte eine Liste der Patienten, welche alphabetisch nach Vornamen sortiert wurde und nicht nach anderen Spalten sortierbar war. Jetzt können die Listenelemente Vorname, Nachname und Geburtsdatum auf- bzw. absteigend sortiert werden. Die drei Buttons haben nun einen Doppelpfeil, der die neue Möglichkeit noch deutlicher sichtbar macht. Im Code wurden dazu drei neue Cases angelegt. Ein neues SubVI, welches in allen drei Cases verwendet wird, ermöglicht das Sortieren von 2D-Arrays, siehe „0.3 Sort 2D Array“.
- Ein Fehler des VIs, der das Programm zum Absturz brachte, wurde behoben. Bei Doppelklick auf die Spaltenüberschriften bzw. eine Leerzeile wird der Wert -1 retourniert. Der Doppelklick-Case wird jedoch verwendet, um das Patientenpfad-

Array auszulesen. Da das Array mit „-1“ indexiert wurde, kam es zu einem Error. Eine Abfrage des Returnwerts verhindert nun diesen Fehler.

- Die boolschen Variablen der Buttons konnten, unter Verwendung von "latch when released", für mehr Übersicht in die jeweiligen Cases verschoben werden. Das Rücksetzen ist in diesem Fall hinfällig und wurde entfernt. Eine Ausnahme bildet die Variable "Liste neu darstellen". Diese benötigt weiterhin „Switch“, konnte aber trotzdem in den Case verschoben werden.

### **2.0.3 Patientenfoto kopieren-V-0.1.vi**

- Wenn der Fotopfad leer ist, wird nun das Foto aus dem leeren Fotopfad (avatar.jpg) verwendet. Festgelegt ist das Standardfoto bzw. dessen Pfad im Lokomotionstherapie-VI.

### **2.0.5 Foto laden-V-0.1.vi**

- Wurde eine Patientenakte, welche kein Foto hatte, aufgerufen und wollte man anschließend die Therapiedetails aufrufen, erhielt man insgesamt drei Mal dieselbe Meldung, dass kein Patientenfoto vorhanden sei. Die VIs „2.2.6 Patient identifizieren“, „2.2 Patientenakte Anzeigen\_Bearbeiten“ und „2.2.2 Therapiedetail anzeigen“ versuchten immer das Foto mit VI 2.0.5 zu laden. Der Hinweis, der jedes Mal mit OK bestätigt werden musste, wurde entfernt und stattdessen ein Avatar mit dem Text „Kein Foto vorhanden“ eingefügt. Existiert ein Patientenfoto, wird es wie auch zuvor geladen und angezeigt.

### **2.1 Patient hinzufügen-V-0.2.vi**

- Der Frontpanel Text „Gewicht in“ wurde zu „Gewicht [kg]“ geändert.
- Der Frontpanel Text "Größe in cm" wurde zu "Größe [cm]" geändert.
- Der Frontpanel Text "Geburtsdatum TT/MM/YY" wurde in "Geburtsdatum" umbenannt, da ohnehin „DD/MM/YY“ im Eingabefeld steht.
- Die Meldung „Patienten anlegen erfolgreich!“ wurde in „Patient erfolgreich angelegt!“ abgeändert.
- Es konnten bereits eingegebene Daten aus den Eingabefeldern gelöscht werden, bevor man speichert und somit wurde eine leere Patientenakte, nur mit Foto und Geburtsdatum, angelegt. Möglich war dies, da lediglich auf eine einmalige Wertänderung, mit Hilfe eines boolschen Arrays, überprüft wurde. Nun wird geprüft, ob die Eingabefelder weder leer noch ungültig sind. Ausnahmen sind die Orthsenlängen, die Bemerkung und der Fotopfad, siehe nächster Punkt.
- Das Frontpanel verwendet nun bei Aufruf das Bild aus dem „leeren Fotopfad“ (avatar.jpg). Dass das Bild bei Aufruf des VIs gelöscht wird, ist somit überflüssig.
- Wird kein Patientenfoto ausgewählt, wird das Bild aus dem „leeren Fotopfad“ (avatar.jpg) für den jeweiligen Patienten in der Akte gesetzt. Dies geschieht in VI „2.0.3 Patientenfoto kopieren-V-0.1.vi“. Zuvor konnte ohne Bild keine Akte angelegt werden.
- Es konnte ein Geburtsdatum in der Zukunft eingegeben werden. Dies wird nun mit einem Vergleich des aktuellen Datums verhindert und mit der Meldung „Unzulässiges Geburtsdatum“ quittiert.

- Es war möglich einen Patienten mit einer Größe von 0 cm und einem Gewicht von 0 kg anzulegen. Dies ist nicht mehr möglich.
- Wurden ein zulässiges Gewicht bzw. eine zulässige Größe eingegeben und danach wieder 0, blieb die Anzeige, ob der jeweilige Wert zulässig ist, auf true. Dies wurde behoben.
- Die Variablen "Größe zulässig:" und "Gewicht zulässig:" wurden in den jeweiligen Case verschoben und direkt verbunden. Die Verwendung von lokalen Variablen war überflüssig.
- Werden eine unzulässige Größe oder ein unzulässiges Gewicht eingegeben erscheint eine Fehlermeldung. Die Fehlermeldung beinhaltet ab sofort auch die max. zulässige Größe bzw. das max. zulässige Gewicht, welches in der „Init Variablen.vi“ spezifiziert ist.
- Die Tabulator Reihenfolge wurde angepasst.
- Die Auswahl im Datei-öffnen-Dialog wurde auf die Dateierendungen „\*.jpg; \*.jpeg“ voreingestellt, da kein Support für andere Dateierendungen besteht.
- Die korrekte Dateierendung des Patientenfotos („\*.jpg; \*.jpeg“) wird überprüft, ansonsten erscheint die Meldung „Es werden nur Bilder im JPG-Format (\*.jpg; \*.jpeg) unterstützt!“ und der Pfad wird zurückgesetzt. Da im VI 2.0.5 die entsprechende Meldung entfernt wurde und statt Zurücksetzen des Pfads der leere Fotopfad eingefügt wird, muss die Überprüfung nun hier stattfinden.
- Die Meldung „Bitte füllen Sie alle Daten aus!“ wurde passenderweise umgeschrieben zu „Bitte prüfen Sie die eingegebenen Daten auf Vollständigkeit und Korrektheit!“.
- Der neue „Gewicht messen“-Button wiegt den Patienten mithilfe der eingebauten Wägezellen. Es erscheint eine Meldung, die das Gewicht anzeigt und es ist möglich das Gewicht direkt in den Patientendaten zu übernehmen. Dafür wurde ein neuer Case im Programm angelegt. Die eigentliche Messung führt das neu angelegte SubVI „0.6 Gewicht messen“ durch.
- Die Return-Taste wurde dem „Speichern“-Button für eine komfortablere Bedienung zugewiesen, die Escape-Taste dem „Abbrechen“-Button.
- Ein ungewollter Zeilenumbruch bei der Eingabe des Vornamens oder Nachnamens wird durch die Option „Limit to Single Line“ verhindert.

## **2.2 Patientenakte anzeigen bearbeiten-V-0.3.vi**

- Die Patientenakte wurde neu gestaltet. Um die Bildschirmdiagonale voll auszunutzen zu können, ohne überladen zu wirken, kann zwischen zwei Ansichten gewählt werden. Die Patientendetails zeigen die persönlichen Daten, die Therapiedetails zeigen therapierelevante Informationen an. Um zwischen den Ansichten hin- und her zu wechseln, gibt es einen neuen Button mit der Aufschrift „Therapiedetails anzeigen“ bzw. „Patientendetails anzeigen“, je nachdem in welcher Ansicht man sich befindet. Dafür mussten zwei neue Cases erstellt werden und eine transparente Tab Control.
- In der oberen Leiste des Frontpanels ist immer „Patientenakte: Vorname Nachname“ zu sehen.
- Der „Zurück“-Button wurde in „Ansicht schließen“ umbenannt
- Die Escape-Taste wurde dem „Ansicht schließen“-Button für eine komfortablere Bedienung zugewiesen.
- Die Funktionalität der Patientenakte wurde erweitert. Es ist möglich, die Akte zu exportieren. Der „Patientenakte speichern“-Button exportiert die Akte im Word-

Format. Der „Patientenakte drucken“-Button ermöglicht, die Akte direkt zu drucken. Dafür mussten zwei neue Cases erstellt werden, siehe VIs 2.2.7 und 2.2.8.

- Der „Anzeigen“-Knopf, zugehörig zur Übersicht der Therapien, konnte ohne gewählte Therapie gedrückt werden. Daraufhin wurde VI „2.2.2 Therapiedetail anzeigen“ ohne Daten aufgerufen. Ein neuer Case, welcher prüft ob bereits eine Therapie ausgewählt wurde, aktiviert den Button. Zuvor ist er ausgegraut und deaktiviert.

### 2.2.1 Patientendaten ändern-V-0.3.vi

- Der Frontpanel Text „Gewicht in“ wurde zu „Gewicht [kg]“ geändert.
- Der Frontpanel Text "Größe in cm" wurde zu "Größe [cm]" geändert.
- Der Frontpanel Text "Geburtsdatum TT/MM/YY" wurde in "Geburtsdatum" umbenannt, da ohnehin „DD/MM/YY“ im Eingabefeld steht.
- Es wurde nicht überprüft, ob Eingabefelder leer waren, bevor gespeichert wurde und somit konnte eine Patientenakte mit fehlenden Daten entstehen. Nun wird überprüft, ob die Eingabefelder weder leer noch ungültig sind. Wenn Daten fehlen oder nicht gültig sind, wird die Meldung "Bitte prüfen Sie die eingegebenen Daten auf Vollständigkeit und Korrektheit!" ausgegeben. Ausgenommen von der Überprüfung sind die Orthesenlängen, die Bemerkung und das Patientenfoto.
- Es konnte ein Geburtsdatum in der Zukunft eingegeben werden. Dies wird nun mit einem Vergleich mit dem aktuellen Datum verhindert und mit der Meldung „Unzulässiges Geburtsdatum“ quittiert.
- Es war möglich, einen Patienten auf eine Größe von 0 cm und ein Gewicht von 0 kg abzuändern. Dies ist nicht mehr möglich.
- Wurde ein zulässiges Gewicht bzw. eine Größe eingegeben und danach wieder 0, blieb die Anzeige, ob der jeweilige Wert zulässig ist, auf true. Dies wurde behoben.
- Die Variablen "Größe zulässig:" und "Gewicht zulässig" wurden in den jeweiligen Cases direkt verbunden. Verwendung von lokalen Variablen war hinfällig.
- Werden eine unzulässige Größe oder ein unzulässiges Gewicht eingegeben erscheint eine Fehlermeldung. Die Fehlermeldung beinhaltet ab sofort auch die max. zulässige Größe bzw. das max. zulässige Gewicht, welches in der „Init Variablen.vi“ spezifiziert ist.
- Die Anpassung der Tabulator-Reihenfolge wurde soweit möglich durchgeführt.
- Die Auswahl im Datei-öffnen-Dialog wurde auf die Dateierendungen „\*.jpg; \*.jpeg“ voreingestellt, da kein Support für andere Dateierendungen besteht.
- Die korrekte Dateierendung des Patientenfotos („\*.jpg; \*.jpeg“) wird überprüft, ansonsten erscheint die Meldung "Es werden nur Bilder im JPG-Format (\*.jpg; \*.jpeg) unterstützt!" und der Pfad wird zurückgesetzt. Da im VI 2.0.5 die entsprechende Meldung entfernt wurde und statt Zurücksetzen des Pfads der leere Fotopfad eingefügt wird, muss die Überprüfung nun hier stattfinden.
- Der neue „Gewicht messen“-Button wiegt den Patienten mithilfe der eingebauten Wägezellen. Es erscheint eine Meldung, die das Gewicht anzeigt und es ist möglich, das Gewicht direkt in den Patientendaten zu übernehmen. Dafür wurde ein neuer Case im Programm angelegt. Die eigentliche Messung führt das neu angelegte SubVI „0.6 Gewicht messen“ durch.
- Ein ungewollter Zeilenumbruch bei der Eingabe des Vornamens oder Nachnamens wird durch die Option „Limit to Single Line“ verhindert.

- Die Return-Taste wurde dem "Speichern"-Button für eine komfortablere Bedienung zugewiesen, die Escape-Taste dem „Abbrechen“-Button.

### **2.2.2 Therapiedetail anzeigen-V-0.1.vi**

- In der oberen Leiste wird nun immer "Therapiedetails: Vorname Nachname Datum Uhrzeit" angezeigt.
- Der "Zurück"-Button wurde in "Ansicht schließen" umbenannt.
- Der „Edit“-Button wurde in „Therapie Anmerkung editieren“ umbenannt.
- Um die Bildschirmdiagonale voll ausnützen zu können, ohne überladen zu wirken, kann zwischen zwei Ansichten gewählt werden. Die Patientendetails zeigen die persönlichen Daten, die Therapiedetails zeigen therapierelevante Informationen an. Um zwischen den Ansichten hin- und her zu wechseln, gibt es einen neuen Button mit der Aufschrift "Therapiedetails anzeigen" bzw. "Patientendetails anzeigen", je nachdem in welcher Ansicht man sich befindet. Dafür mussten zwei neue Cases erstellt werden und eine transparente Tab Control.
- Die Escape-Taste wurde dem "Ansicht schließen"-Button für eine komfortablere Bedienung zugewiesen.

### **2.2.6 Patient identifizieren-V-0.1.vi**

- Der Text „Bitte Überprüfen Sie die Identifizierung:“ wurde zu „Patient bestätigen:“ geändert.
- Die Return-Taste wurde dem "Korrekt Patient"-Button für eine komfortablere Bedienung zugewiesen, die Escape-Taste dem "Falscher Patient"-Button.

### **2.2.7 Patientenakte drucken-V-0.vi**

- Dieses VI wurde neu erstellt und generiert die Patientenakte zum Drucken. Aufrufbar ist es aus „2.2 Patientenakte Anzeigen\_Bearbeiten“, wenn der „Patientenakte drucken“-Button gedrückt wird.

### **2.2.8 Patientenakte exportieren-V-0.vi**

- Dieses VI wurde neu erstellt und generiert die Patientenakte und exportiert diese anschließend im Word-Format. Aufrufbar ist es aus „2.2 Patientenakte Anzeigen\_Bearbeiten“, wenn der „Patientenakte speichern“-Button gedrückt wird.

## **3. Therapiemodus-V-0.2.vi**

- Es war möglich beim Control „geplante Dauer [min]“ den Wert 0 bzw. negative Werte einzustellen, woraufhin sich die Therapie sofort beendete. Die untere Grenze wurde sinnvollerweise mit einer Minute in den Variablenoptionen festgelegt. Die obere Grenze war in den Variablenoptionen nicht gesetzt. Es erfolgte eine Überprüfung im Timeout-Case, welche so allerdings keinen Sinn ergab. Es war möglich, eine Therapiedauer über dem globalen Limit, welches vom Programm zu Beginn aus „Init.xml“ geladen wird, einzustellen. Nachdem das globale Zeitlimit allerdings erreicht war, endete die Therapie. Durch das Re-Design wird die Obergrenze der einstellbaren Therapiedauer nun direkt vom globalen Limit übernommen. Somit kann

nicht mehr als das Limit eingestellt werden und die Überprüfung im Timeout-Case konnte entfernt werden.

- Die Benutzerinteraktionsschleife hat einen 10. Case erhalten. Mit dem neuen Button „Patientendaten anzeigen“ kann zwischen den Patientendaten und dem Schrittdiagramm hin- und her gewechselt werden.
- Die Benutzerinteraktionsschleife hat einen 11. Case (Key Down) erhalten der auf Eingabe von Tasten reagiert. Der Fußschalter, welcher der Aufmerksamkeitsüberwachung dient, simuliert einen Tastendruck. Eine boolesche Variable „Fußschalter betätigt“ wird dementsprechend bei Betätigung gesetzt und leitet den Status an das SubVI 3.4.7 „Aufmerksamkeitsüberwachung“ weiter.

### **3.0.7 Grad\_zu\_Inkrement-V-0.1.vi**

- Mathscript RT ist ein LabView Addon und ermöglicht es unter anderem, text-basiert zu programmieren, allerdings ist die Lizenz kostenpflichtig und das Addon muss zusätzlich installiert werden. Das Mathscript wurde entfernt, da es ohnehin nicht benötigt wurde. Für das Endgerät ist somit keine eigenständige Lizenz mehr von Nöten.

### **3.0.8 Inkremente\_zu\_Grad-V-0.1.vi**

- Mathscript wurde entfernt, siehe VI „3.0.7 Grad\_zu\_Inkrement“.

### **3.1.2 Patient wählen-V-0.1.vi**

- Der „Wählen“-Button wurde auf "Latch when released" gesetzt, somit kann er in den zugehörigen Case platziert werden und das Zurücksetzen auf false vollzieht sich automatisch.
- Der „Wählen“-Button ist nun erst bei getroffener Auswahl anwählbar. Dafür wurde ein neuer Event case angelegt.
- Die Listenelemente Vorname, Nachname und Geburtsdatum können jetzt auf- bzw. absteigend sortiert werden. Die drei Buttons haben nun einen Doppelpfeil, der die neue Möglichkeit noch deutlicher sichtbar machen soll. Im Code wurden dazu drei neue Cases angelegt. Ein neues SubVI, welches in allen drei Cases verwendet wird, ermöglicht das Sortieren von 2D-Arrays, siehe „0.3 Sort 2D Array“.
- Wurde ein Patient durch Doppelklick in der Liste oder durch den entsprechenden Button ausgewählt, öffnete sich ein Fenster zur Bestätigung der Patientenauswahl (VI 2.2.6). Wurde im neu erschienenen Fenster „Falscher Patient“ gedrückt, hat sich auch das aufrufende VI 3.1.2 geschlossen. Die neue eingefügte Case Structure verhindert dies und man kann die Eingabe korrigieren.
- Die Return-Taste wurde dem "Wählen"-Button für eine komfortablere Bedienung zugewiesen, die Escape-Taste dem "Abbrechen"-Button.

### **3.2.1 Init USB2CAN-V-0.1.vi**

- Die Handles (CAN Channel Handle, Can Control Handle und Device Handle) werden ab sofort in gleichnamigen globalen Variablen in dem VI „Current CAN VCI Handle.vi“ gespeichert, um das SOUP „devcon.exe“ zu ersetzen. Das VI „3.2.1.1 Reset USB-CAN“ benötigt die Handles zum Schließen bzw. resettieren der Verbindung.

### 3.2.1.1 Reset USB-CAN-V-0.1.vi

- Das verwendete SOUP „devcon.exe“ dient zum Zurücksetzen von USB-Verbindungen. Allerdings ist es abhängig von der Systemarchitektur (x86 oder x64) und führt trotz richtigem System häufig zu Problemen. Auf dem Entwicklungssystem und dem Endgerät, auf dem die Lokomotionstherapie-Software zum Einsatz kommt, war devcon.exe nicht funktionstüchtig und der USB2CAN-Converter musste regelmäßig an- bzw. abgesteckt werden. Der Inhalt des VIs wurde durch die vorgesehenen VIs zum Schließen (CANChannelClose, CanControlClose und VCIDeviceClose) ersetzt. Somit kann insgesamt auf die SOUP verzichtet werden. Erneutes Initialisieren der Kommunikation und auch Beenden und Neustarten des Programms sind nun ohne Probleme möglich. Dafür wurde dasselbe VI auch im „Beenden“-Case von „Lokomotionstherapie-V-0.3.vi“ platziert. Der größte Vorteil ergibt sich in der Unabhängigkeit von der Systemarchitektur und dass ab sofort nicht mehr händisch an- bzw. abgesteckt werden muss.

### 3.2.4 Setting Antriebe-V-0.1.vi

- Zu Beginn der Therapie werden gewisse Werte in den Motion Controllern durch die Software gesetzt, unter anderem der Spitzen- und Dauerstrom. Die Übernahme jener funktioniert zwar, dies kann durch Auslesen der EEPROMs mit der Faulhaber Motion Manager Software bewiesen werden, die Antworten der Motion Controller auf diese CAN-Befehle sind allerdings fehlerhaft bzw. kommen fehlerhaft an. Die Lokomotionstherapie-Software war somit vor dem Re-Design aufgrund des Errors auf der neuen Hardware nicht lauffähig. Abhilfe schaffte das manuelle Setzen der Grenzen mit der Herstellersoftware und Deaktivierung der entsprechenden Befehle in der VI. Der Grund für den Fehler bleibt ungeklärt, vor allem da die Software auf dem Vorgänger-Lokomotionsgerät einwandfrei funktioniert und sämtliche anderen Antworten auf Befehle korrekt ankommen.

### 3.3. Therapie-V-0.1.vi

- Das VI benötigte einen neuen Eingang für die Weiterleitung der „Patientenakte“ an VI „3.3.2 zyklische Bewegung“.

### 3.3.2 zyklische Bewegung-V-0.vi

- Mathscript RT ist ein LabView Addon und ermöglicht es unter anderem, text-basiert zu programmieren, allerdings ist die Lizenz kostenpflichtig und das Addon muss installiert werden. Das in der VI enthaltene Mathscript konnte durch herkömmlichen LabView-Code ersetzt werden. Für das Endgerät ist somit keine eigenständige Lizenz mehr von Nöten.
- Das VI benötigte einen neuen Eingang für die „Patientenakte“ zur Berechnung der Schrittlänge.
- Im VI wird die Ansteuerung des automatisierten Laufbands umgesetzt. Um ein langsames Anfahren bzw. Abbremsen zu ermöglichen wurde eine Rampenfunktion implementiert. Zusätzlich wird die Geschwindigkeit des Laufbands auf die Schrittlänge des Patienten angepasst.



### 3.3.2.1 Schrittlänge berechnen-V-0.vi

- Dieses neu erstellte VI verwendet die Patientenakte, um aus der Ober- und Unterschenkelänge inklusive Fuß die Beinlänge zu berechnen. Mit Hilfe dieser kann auf die Schrittlänge, welche retourniert wird, geschlossen werden.

### 3.4 Überwachung-V-0.1.vi

- Für die Aufmerksamkeitsüberwachung benötigte es einen neuen Eingang. Die Referenz von „Fußschalter betätigt“ wird übergeben und direkt in das SubVI „3.4.7 Aufmerksamkeitsüberwachung“ weitergeleitet.

#### 3.4.1.1 CAN-Nachrichten Verwalten-V-0.1.vi

- Es existierte ein Case (Arbitration ID 128..132), welcher ein deaktiviertes SubVI enthielt, das sich im Ordner „nicht benötigt“ befand. Der Case und der Ordner inklusive SubVI wurden entfernt.

#### 3.4.7 Aufmerksamkeitsüberwachung-V-0.1.vi

- Für die Aufmerksamkeitsüberwachung benötigte es einen neuen Eingang. Die Referenz von „Fußschalter betätigt“ wird aus VI „3.4 Überwachung“ übergeben. Betätigen des Fußschalters resettet ab sofort, wie auch eine beliebige Mausbewegung, den Countdown der Aufmerksamkeitsüberwachung. Dazu wird der boolsche Wert ausgelesen und durch ein logisches „Oder“ verknüpft, mit der Abfrage, ob sich die X-Y-Koordinaten der Maus geändert haben. Trifft einer der beiden Fälle zu, wird der boolsche Wert auf false zurückgesetzt und der Countdown beginnt erneut.

### 3.5 Steuerung-V-0.2.vi

- Die Einstellung der Schrittgeschwindigkeit (Ziel Kadenz) war vom Aufruf des Therapiemodus an verstellbar und setzte sich des Weiteren nach jedem bestätigten Schritt der „Checkliste zum Therapiestart“ zurück. Dies konnte leicht zu Verwirrung der Anwender führen und deshalb ist sie jetzt erst dann verstellbar, wenn die Motoren an sind. Zuvor ist sie ausgegraut und nicht verstellbar. Nach Beenden der Therapie ist sie wieder ausgegraut, allerdings wird die Einstellung nicht sofort wieder zurückgesetzt, sondern kann noch abgelesen werden. Dazu wurde die Referenz „Ziel Kadenz“ auf „disabled and grey“, für die jeweiligen Zustände gesetzt.
- Die Anzeige der Schrittgeschwindigkeit (Ist-Kadenz) war vom Aufruf des Therapiemodus an sichtbar und Anwender versuchten teilweise mit der Anzeige, anstatt der Einstellung, die Geschwindigkeit zu setzen. Jetzt ist die Anzeige zu Beginn ausgegraut und erst ab Bewegungsbeginn der Orthesen (Zustand: Therapie läuft) sichtbar und zum Abschluss wieder ausgegraut. Dazu wurde die Referenz „Ist Kadenz“ auf „disabled and grey“, für die jeweiligen Zustände gesetzt.
- Der „Not-Stop“-Button war bereits zu Beginn des Therapiemodus anwählbar. Wurde er vor Initialisierung der Kommunikation gedrückt, führte dies unpassender Weise zur Anzeige „Kommunikation-EIN“. Nun ist der Button erst nachdem die Kommunikation initialisiert wurde anwählbar.

- Die Anzeige der Schrittgeschwindigkeit (Ist-Kadenz) wird für den Zustand "Z:Init GUI" auf 0 gesetzt. Trat ein Fehler auf und wurde das Frontpanel wieder in diesen Zustand zurückversetzt, resettierte die Anzeige nicht.
- Nach diversen Fehlern (z.B.: Node-Guarding, Statusänderungsfehler,...) wird das Frontpanel des Therapiemodus in den Zustand "Z:Init GUI" versetzt, also noch vor Therapiestart. Dennoch lief die absolvierte Zeit weiter. Deshalb musste die Variable "Timer On/Off" im Zustand "Z:Init GUI" eingefügt und auf false gesetzt werden.

### **3.5.3 Online Fehlerbehandlung-V-0.2.vi**

- Ein neuer Error Case (" -201003, -50808") wurde hinzugefügt. Dieser Case tritt auf, wenn kein USB-DAQ-Device für die NOT-AUS-Detektion, bzw. die Gewichtsmessung verbunden ist.
- Wenn „Kommunikation initialisieren“ einen Fehler auslöst (Error Code 5008), wird das SubVI „3.0.6 Antriebe abschalten“ aufgerufen, um die Antriebe zu deaktivieren. Waren die Antriebe jedoch nicht in Betrieb, kommt es just von jenem SubVI erneut zu einem Error, der dann vom Default Case abgefangen wurde und als Unbekannter Fehler dem User angezeigt wurde. Deshalb wird im Error Case 5008 der Error mit dem Code -536805369 gelöscht, da bereits Fehler 5008 für das eigentliche Problem steht.
- Der Text von Fehlermeldung 5008 wurde leicht abgeändert. Bei fehlerhafter Kommunikation wird nun auf die Abhilfemaßnahme des Aus- und Ansteckens der USB-Verbindung hingewiesen.
- Der Case 5022 wurde hinzugefügt und steht für einen aktiven Not-Aus-Schalter. Gesendet wird die Nachricht Z: Init Gui, da wieder alle Schritte (Kommunikation initialisieren, ...) nach Spannungsunterbrechung durchgeführt werden müssen.
- Bei einer Drehmomentüberschreitung der Hüfte bzw. des Knies erschien zusätzlich zur Error-Meldung eine Bestätigung mit demselben Text, die nur mit OK quittiert werden konnte. Diese redundante Meldung wurde deaktiviert.

### **3.6.1 Speichern der Therapieinformationen-V-0.1.vi**

- Wurde die Therapie mit 0 min bzw. 0 Schritten beendet, ergab sich eine Division durch 0. In VI „2.2.2 Therapiedetail anzeigen“ bzw. auch VI „2.2 Patientenakte anzeigen bearbeiten“ wurde für die durchschnittliche Geschwindigkeit bzw. den durchschnittlichen Kräften je Gelenk NaN (Not a Number) angezeigt. Jetzt wird auf NaN überprüft, und in diesem Fall die jeweilige Variable 0 gesetzt.

### **4.2 Fehlerfolge-V-0.1.vi**

- Der Case „5001..5021“ des VIs, der die Fehlercodes repräsentiert, musste auf „5001..5022“ erweitert werden, da 5022 für "Not-Aus betätigt" steht aber deswegen nicht das Programm beendet werden soll.

## E. Vorschläge für Lokomotionstherapie V-0.4

Die nachfolgenden Aufzählungen beinhalten zusätzliche Erkenntnisse, welche während der Entwicklung von Lokomotionstherapie-V-0.3 gesammelt wurden. Aufgrund des Umfangs der Arbeit wurden diese jedoch nicht mehr umgesetzt und sind für eine nachfolgende Software-Version angedacht.

1. Das Error Handling vom Therapiemodus und den SubVIs könnte vereinheitlicht und möglicherweise vereinfacht werden.
2. Sämtliche VIs mit Listen von Patienten und Therapeuten sollten, soweit möglich, analog aufgebaut werden.
3. In VI „3.5 Steuerung“ sollten die Verwendung von lokalen Variablen reduziert werden, da bei Anlage von weiteren LabView bereits einen Memory Error ausgibt.
4. Sämtliche VIs bieten die Möglichkeit der Dokumentation in den Eigenschaften. Somit könnte zu den Kommentaren im Blockdiagramm eine zusätzliche Kurzbeschreibung aller VIs realisiert werden. Wird mit geöffnetem Hilfefenster (STRG+H) über ein VI gefahren, sind sämtliche Ein- und Ausgänge und die Dokumentation ersichtlich. Siehe die neu erstellten VIs, z. B.: „0.3 Sort 2D Array-V-0.vi“.
5. Eine automatische Anpassung der Frontpanele an die Bildschirmgröße (Auflösung) wäre denkbar.
6. Die Repräsentation von Variablen (U32, I32,...) könnte überdacht und angepasst werden. Mit dieser sollte auch auf „numeric conversions“ und „coercion dots“ geachtet werden.
7. Es sollte ein boolscher Schalter (z.B.: global) integriert werden, welcher die Möglichkeit bietet, das Node Guarding temporär für ein verbessertes Debugging zu deaktivieren.
8. Der Störausgang des Rollerdrives könnte über das USB DAQ-Device ausgewertet werden.
9. Mit den Wägezellen könnte versucht werden, zusätzlich zum Gewicht auch die Kraft bzw. die Eigenständigkeit des Patienten zu bestimmen. Dies geschieht bisher indirekt über den Motorstrom der Orthesen.
10. Wird der Ordner mit der Lokomotionstherapie-Software und den bereits vorhandenen Patientendaten verschoben, kann der gespeicherte Patienten- bzw. Fotopfad im „pakt“-File Problemstellungen verursachen. Aufgrund der absoluten Pfade kann zum einen das patientenbezogene Foto nicht mehr angezeigt werden, zum anderen wird bei Bearbeitung der Patientenakte diese an ihrem ursprünglichen Ort erneut generiert. Das Problem kann durch Verwendung von relativen Pfaden behoben werden.