



fachhochschule
stralsund
university of
applied
sciences

—
fachbereich school of
elektrotechnik electrical engineering +
+ informatik computer science

Richtlinienkonformes Softwaredesign eines Medikamentendispensers für den HomeCare-Bereich

Abschlussarbeit für die Erlangung

des akademischen Grades

Master of Science (M.Sc.)

im Studiengang Medizininformatik

Vorgelegt von Torsten Philipp

aus Stralsund

2011

Erstgutachter und Betreuer: Prof. Dr. med. MPH Stefan Klatt

Zweitgutachter: Prof. Dr. sc. hum. Dipl.-Ing. (FH) Holger Specht

Verfasser: Torsten Philipp

Kontakt: Torsten-Philipp@web.de

Abgabe der Arbeit am

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen wurden, sind als solche kenntlich gemacht. Alle Abbildungen in dieser Arbeit sind von mir selbst erstellt oder mit einem entsprechenden Quellennachweis versehen worden.

Torsten Philipp

Stralsund, den

Inhaltsverzeichnis

| | | |
|-------|---|----|
| 1 | Einleitung | 1 |
| 1.1 | Systemidee des HomeCare Medidisp | 1 |
| 1.2 | Ausgangsspezifikation | 2 |
| 1.3 | Aufgabenstellung | 3 |
| 2 | Grundlagen | 4 |
| 2.1 | Rechtliche Rahmenbedingungen | 4 |
| 2.2 | Grundlegende gesetzliche Vorgaben seitens MPG | 5 |
| 2.2.1 | Bestimmung als Medizinprodukt | 6 |
| 2.2.2 | Relevante Vorgaben aus dem MPG | 11 |
| 2.3 | Konkrete Anforderungen an das Design (Harmonisierte Normen) | 17 |
| 2.3.1 | Sicherheitsvorgaben aus DIN EN 60601-1 | 18 |
| 2.3.2 | Vorgaben aus DIN EN 60601-1-4 und DIN EN 62304 | 19 |
| 2.3.3 | Vorgaben an das Risikomanagement aus DIN EN ISO 14971 | 25 |
| 2.3.4 | Aspekte aus der DIN EN 62366 | 31 |
| 2.3.5 | Übernommene Prinzipien aus den Normen | 35 |
| 3 | Richtlinienkonformes Design | 36 |
| 3.1 | Entwicklungs-Lebenszyklus-Modell nach DIN EN 60601-1-4 | 37 |
| 3.2 | Phase 1 - Anforderungsanalyse | 42 |
| 3.3 | Phase 2 - Entwurf des Systems | 45 |
| 3.4 | Phase 3 - 6 - Software-Entwicklungsverfahren | 49 |
| 3.4.1 | Das V-Modell XT | 49 |
| 3.4.2 | Agile Softwareentwicklung | 52 |
| 3.4.3 | V-Modell oder agile Softwareentwicklung? | 55 |
| 3.5 | Phase 7 - 8 - Abschluss der Entwicklung | 57 |

| | | |
|-------|---|-----|
| 4 | Software des Medikamentendispensers im HomeCare-Bereich | 58 |
| 4.1 | Anforderungsanalyse des Gesamtsystems | 58 |
| 4.1.1 | Ermittlung von Anforderungen aus einer Prozessanalyse..... | 61 |
| 4.1.2 | Festlegung des USB-Protokolls | 62 |
| 4.2 | Erster Entwurf des Gesamtsystems | 67 |
| 4.2.1 | Architektspezifikation PEMS (Gesamtsystem) | 67 |
| 4.2.2 | Spezifikation der Anwendungen..... | 68 |
| 4.2.3 | Ermittlung der häufig genutzten Funktionen | 70 |
| 4.2.4 | Festlegung der Hauptbedienfunktionen..... | 70 |
| 4.2.5 | Erste Risikomaßnahmen | 71 |
| 4.3 | Entwicklung nach Scrum | 77 |
| 4.3.1 | Vorbetrachtungen..... | 77 |
| 4.3.2 | Der erste Sprint..... | 79 |
| 4.3.3 | Nachbearbeitung des ersten Sprints | 94 |
| 5 | Umsetzungsaspekte | 99 |
| 5.1 | Doppelte Sicherheit..... | 99 |
| 5.2 | Defensives Design | 101 |
| 5.3 | Usability des Systems | 103 |
| 6 | Diskussion | 104 |
| 7 | Zusammenfassung | 107 |
| 8 | Literaturverzeichnis | 108 |
| 8.1 | Normen- und Richtlinienregister | 108 |
| 8.2 | Internetquellen | 108 |
| 8.3 | Literatur..... | 111 |

| | | |
|--------|--------------------------|----|
| 9 | Anhänge..... | 1 |
| 9.1 | Anhang 1 | 1 |
| 9.2 | Anhang 2 | 3 |
| 9.3 | Anhang 3 | 4 |
| 9.3.1 | USB-Kommunikation..... | 4 |
| 9.3.2 | USB-Deskriptoren | 5 |
| 9.3.3 | USB-Transfermodi..... | 9 |
| 9.4 | Anhang 4 | 10 |
| 9.5 | Anhang 5 | 12 |
| 9.6 | Anhang 6 | 18 |
| 9.6.1 | Änderungshistorie..... | 18 |
| 9.6.2 | Backlog-Übersicht | 18 |
| 9.6.3 | Inhalt User Stories..... | 19 |
| 9.6.4 | User Story 1..... | 19 |
| 9.6.5 | User Story 2..... | 20 |
| 9.6.6 | User Story 3..... | 20 |
| 9.6.7 | User Story 4..... | 21 |
| 9.6.8 | User Story 5..... | 22 |
| 9.6.9 | User Story 6..... | 23 |
| 9.6.10 | User Story 7..... | 23 |
| 9.6.11 | User Story 8..... | 24 |
| 9.7 | Anhang 7 | 25 |
| 9.8 | Anhang 8 | 26 |
| 9.9 | Anhang 9 | 27 |
| 9.9.1 | Szenario 1..... | 27 |
| 9.9.2 | Szenario 2..... | 31 |
| 9.9.3 | Szenario 3..... | 34 |
| 9.9.4 | Zusammenfassung | 38 |
| 9.10 | Anhang 10 | 39 |
| 9.11 | Anhang 11 | 41 |
| 9.12 | Anhang 12 | 42 |

Abbildungsverzeichnis

| | |
|--|----|
| Abbildung 1: Beispiel Medikamentendispenser | 1 |
| Abbildung 2: Systemidee elektronischer Medikamentendispenser (selbst erstellt) .. | 1 |
| Abbildung 3: Rechtsübersicht für Standort Deutschland (selbst erstellt) | 4 |
| Abbildung 4: Rechtlich konforme Gliederung einer Zweckbestimmung..... | 10 |
| Abbildung 5: Ablauf Konformitätsbewertungsverfahren | 15 |
| Abbildung 6: Beweislast bei Verwendung harmonisierter Normen..... | 17 |
| Abbildung 7: Wo setzen die Normen an? Nach Prof. Dr. J. Stettin [IQ09], S. 2 | 19 |
| Abbildung 8: PEMS/PESS Systemvorschlag (selbst erstellt) | 20 |
| Abbildung 9: Entwicklungs-Lebenszyklus-Modell nach DIN EN 60601-1-4..... | 22 |
| Abbildung 10: Komplexität versus Testabdeckung, Dr. C. Lindner [IQ12], S. 23 | 24 |
| Abbildung 11: Entwicklungs-Lebenszyklus-Modell nach DIN EN 62304 | 25 |
| Abbildung 12: Definition Gebrauchstauglichkeit C. Backhaus [BACK2009], S. 17.... | 31 |
| Abbildung 13: Verzahnung zwischen DIN EN 62366 und DIN EN ISO 14971 | 33 |
| Abbildung 14: Benutzer Taxonomie nach DIN EN 62366, S. 33..... | 34 |
| Abbildung 15: Modifiziertes E-L-Modell nach DIN EN 60601-1-4 (selbst erstellt) | 38 |
| Abbildung 16: Bewertung der Gebrauchstauglichkeit von Medizinprodukten..... | 40 |
| Abbildung 17: Beispiel Evaluation einer neuen Infusionstherapiebaureihe | 41 |
| Abbildung 18: Beispiel TOPICS [BACK2009], S. 115 | 44 |
| Abbildung 19: Typisches vorgehen nach V-Modell (selbst erstellt) | 50 |
| Abbildung 20: iterativ inkrementelles Vorgehen im V-Modell (Wikipedia) | 50 |
| Abbildung 21: Iterativer Zyklus in der agilen Softwareentwicklung [IQ16] | 53 |
| Abbildung 22: Übersicht Scrum [IQ17]..... | 54 |
| Abbildung 23: Beispiel für iterative Vorgehensweise im V-Modell (selbst erstellt).. | 55 |
| Abbildung 24: Prozessanalyse nach C. Backhaus (selbst erstellt) | 61 |
| Abbildung 25: Umfeld USB-PHD-Klasse [PHDC2007], S. 9 | 64 |
| Abbildung 26: Descriptor HID Beispiel (selbst erstellt)..... | 65 |
| Abbildung 27: PEMS/PESS Struktur für den HC Medidisp | 67 |
| Abbildung 28: Ablauf sicherer USB-Kommunikation (selbst erstellt)..... | 75 |
| Abbildung 29: Planungsprozess im Scrum [GLOG2008] | 77 |
| Abbildung 30: Use-Case-Diagramm HcMd-S (selbst erstellt) | 78 |
| Abbildung 31: Domänenmodell der Fachebene (selbst erstellt)..... | 78 |
| Abbildung 32: Architektur des HcMd-S mit Simulationsbetrieb (selbst erstellt) | 79 |
| Abbildung 33: Design Hardwaresimulation (selbst erstellt) | 80 |
| Abbildung 34: GUI-Entwurf 1 für den Simulationsbetrieb (selbst erstellt) | 81 |
| Abbildung 35: GUI-Entwurf 2 für den Simulationsbetrieb (selbst erstellt) | 81 |
| Abbildung 36: GUI für Simulationsbetrieb Final (selbst erstellt)..... | 81 |
| Abbildung 37: State Chart <i>Medikamentenausgabe</i> (selbst erstellt) | 82 |
| Abbildung 38: State Chart <i>Pharmakologische Profile Bearbeiten</i> (selbst erstellt).... | 82 |

| | |
|--|-----|
| Abbildung 39: State Chart <i>Verordnungsdatensätze verwalten</i> (selbst erstellt) | 83 |
| Abbildung 40: State Chart <i>Medikamentenlagerung verwalten</i> (selbst erstellt) | 83 |
| Abbildung 41: State Chart <i>Protokolle ausgeben</i> (selbst erstellt)..... | 83 |
| Abbildung 42: Erste GUI-Idee Standby-Monitor (selbst erstellt) | 84 |
| Abbildung 43: Standby-Monitor, erste finale Version (selbst erstellt)..... | 85 |
| Abbildung 44: Sonderfall-Monitor, erste finale Version (selbst erstellt) | 85 |
| Abbildung 45: Klassendiagramm des USB-Moduls (selbst erstellt)..... | 87 |
| Abbildung 46: Klassendiagramm des XML-Moduls (selbst erstellt) | 89 |
| Abbildung 47: Auszug XSD-Schema Patient (selbst erstellt) | 90 |
| Abbildung 48: Auszug XSD-Schema Verordnung (selbst erstellt)..... | 90 |
| Abbildung 49: Auszug XSD-Schema Vertragsarzt (selbst erstellt) | 91 |
| Abbildung 50: Auszug XSD-Schema Präparat (selbst erstellt) | 91 |
| Abbildung 51: Architektur HcMd-CS (selbst erstellt)..... | 93 |
| Abbildung 52: Klassenmodell Kontrollsoftware Schnittstelle (selbst erstellt) | 94 |
| Abbildung 53: Standby-Monitor (überarbeitet) (selbst erstellt) | 96 |
| Abbildung 54: Medikamentenausgabe-Monitor Popup-Beispiel (selbst erstellt).... | 97 |
| Abbildung 55: Abfragealgorithmus Medikamentendosis 1 (selbst erstellt)..... | 100 |
| Abbildung 56: Abfragealgorithmus Medikamentendosis 2 (selbst erstellt)..... | 100 |

Tabellenverzeichnis

| | |
|--|----|
| Tabelle 1: Quantitative Wahrscheinlichkeitsgrade..... | 30 |
| Tabelle 2: Qualitative Schweregrade | 30 |
| Tabelle 3: Ermittlung Risiko mit 5x5-Matrix | 30 |
| Tabelle 4: Prozessunterstützung und Prozessrelevanz [BACK2009], S. 85 | 44 |
| Tabelle 5: Vorgehensbausteine in Abhängigkeit des Projekttyps | 51 |
| Tabelle 6: Gewünschte/Ungewünschte Funktionen/Eigenschaften (selbst erstellt) 60 | 60 |
| Tabelle 7: Prozessanforderungen aus der Prozessanalyse (selbst erstellt)..... | 62 |
| Tabelle 8: Übertragungsmodi PHDC | 64 |
| Tabelle 9: Erste Risikobewertung für Schweregrad <i>Katastrophal</i> (selbst erstellt).... | 74 |

1 Einleitung

1.1 Systemidee des HomeCare Medidisp

Medikamentendispenser sind in der häuslichen (HomeCare) Medikamententherapie nicht mehr wegzudenken.



Abbildung 1: Beispiel Medikamentendispenser¹

Diese Arbeit beschäftigt sich mit der Idee eines elektronischen Medikamentendispensers und mit dem richtlinienkonformen Design dessen Softwarekomponente. Der prinzipielle Aufbau des Systems ist in Abbildung 2 zu sehen.

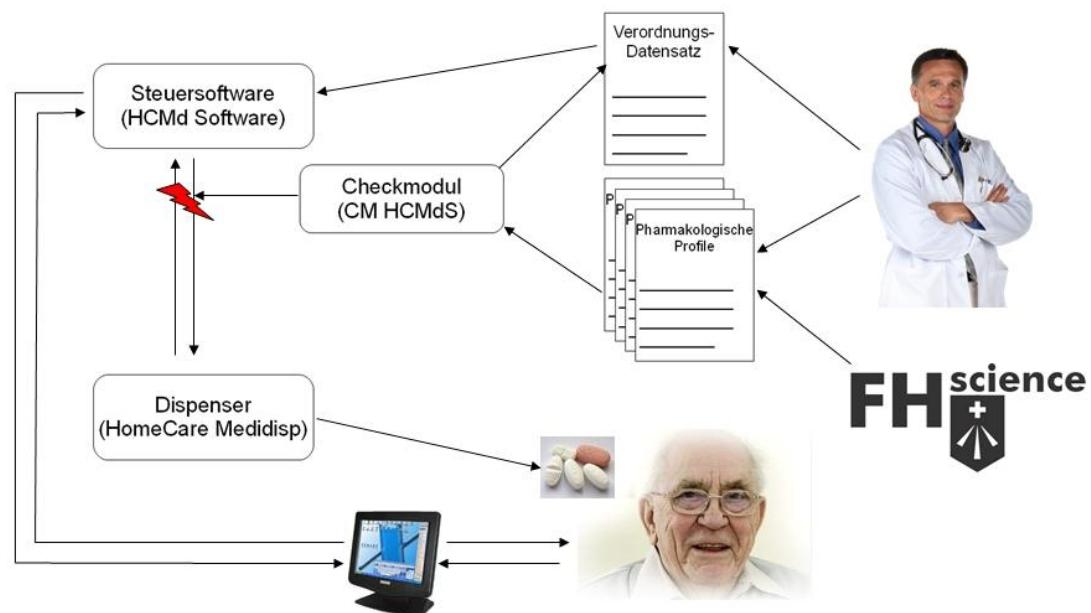


Abbildung 2: Systemidee elektronischer Medikamentendispenser (selbst erstellt)

¹ Abbildung von: <http://eco-medizintechnik.eshop.t-online.de> (BIGONE Medikamentendispenser), 11.10.2011

Das System des elektronischen Medikamentendispensers wurde HomeCare Medidisp getauft. Der Patient erhält über die Hardwarekomponente seine Medikamente in einen für das System geschaffenen Becher. Die Becher spiegeln anhand der Farbe die Tageszeiten² wieder. Das System erkennt den untergestellten Becher und reagiert darauf. Über die Softwarekomponente des Systems wird der Patient an Termine erinnert und die Ausgabe der Medikamente über die Becher gesteuert und protokolliert. Der Arzt kann das System über die Arztsicht der Software beliebig konfigurieren, sodass übliche Verordnungen und die Ausgabe von Bedarfsmedikamenten möglich sind. Über die Becher können somit auch unterschiedliche Patientenprofile angesprochen werden - beispielsweise bei einem älteren Ehepaar. Über sogenannte pharmakologische Profile wird die Sicherheit der Verordnungen gewährt. Jedes Präparat muss ein pharmakologisches Profil besitzen. Pharmakologische Profile enthalten Dosisangaben und unerlaubte Kombinationen mit anderen Medikamenten.

1.2 Ausgangsspezifikation

Basierend auf der Systemidee wurden folgende Anforderungen als Ausgangsspezifikation erstellt.

Anforderungen an die Hardware:

- Elektronischer Medikamentendispenser mit 10 Lagerschächten
- Medikamentenbehältnisse in den Lagerschächten austauschbar
- Identität der Medikamente werden über Barcode (PZN) ermittelt
- Füllstand der Lagerbehältnisse über Lichtschranken ermittelbar
- Füllstand des Ausgabebechers über Lichtschranken ermittelbar
- Medizinrechtlich konforme Schnittstelle zwischen Software und Hardware

Anforderungen an die Software:

- Steuerbefehle an Hardware senden
- Statusmeldungen von Hardware empfangen und verarbeiten können
- Verordnungsdatensätze/pharmak. Profile verwalten und anwenden
- Dialoge für häusliche Medikamententherapie (Medikamentenausgabe/Terminerinnerung)
- Dialoge angepasst an die Patientenzielgruppe
- Angepasstes Interface für Touchscreen-Monitor

² Morgens (gelb), Mittags (grün), Abends (blau)

1.3 Aufgabenstellung

Der Kandidat soll unter Zugrundlegung der vorab festgelegten Rahmenbedingungen eine Softwarelösung entwerfen, die den praktischen Erfordernissen entspricht und dabei (nachvollziehbar) prinzipiell zulassungsfähig nach den Kriterien des MPG und anderer Normen/Standards ist.

Dies betrifft Aspekte der Schnittstellengestaltung, Logik, Bedienbarkeit und Architektur.

Der Schwerpunkt der Arbeit liegt hierbei in der Usability und dem Zusammenspiel mit dem Dispenser.

Folgende Teilaufgaben sollen bearbeitet werden:

- Bestandsaufnahme der rechtlichen Voraussetzungen.
- Funktionell und rechtlich konformes Design und Umsetzung einer Schnittstelle, inklusive grundlegender Dispensersimulation.
- Auswahl und Beschreibung einer geeigneten Methodik zum situations-/ patientenzentrierten Design des HomeCare Medidisp.
- Anwendung und Evaluation der Methodik anhand ausgewählter Anwendungsfälle.
- Prototypische Umsetzung der ausgewählten Anwendungsfälle auf der Grundlage der Dispensersimulation.

2 Grundlagen

2.1 Rechtliche Rahmenbedingungen

Um ein richtlinienkonformes Softwaredesign zu ermöglichen, müssen bestimmte rechtliche Grundlagen eingehalten werden. Deutschland zählt zu dem europäischen Wirtschaftsraum (EWR). Aufgrund dessen gelten die europäischen gesetzlichen Rahmenbedingungen für Deutschland. Die europaweiten Richtlinien regeln sämtliche Bedingungen für den europäischen Raum. Das nationale Medizinproduktegesetz (MPG)³, welches sich aus den Richtlinien bildet (Abbildung 3), regelt spezifisch nur für Deutschland die Entwicklung und Zulassung von Medizinprodukten.

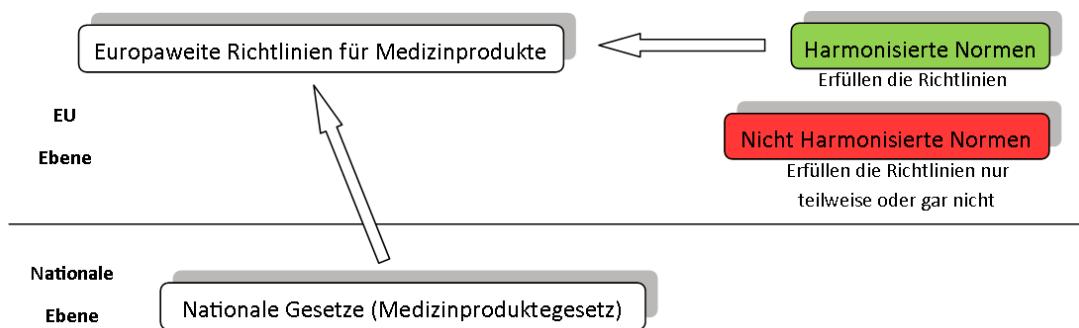


Abbildung 3: Rechtsübersicht für Standort Deutschland (selbst erstellt)

Für Medizinprodukte gelten im Allgemeinen die Richtlinien 93/42/EWG⁴, 98/79/EG, 90/385/EWG, 2000/70/EG, 2001/104/EG, 2003/32/EG, 2006/42/EG und 2007/47/EG. Neben den rechtlichen Vorlagen existieren zusätzlich sogenannte Normen. Diese sind entweder harmonisiert oder nicht harmonisiert. Bei Einhaltung einer harmonisierten Norm wird von einer rechtlich konformen Durchführung nach dazugehöriger EU Richtlinie ausgegangen. Welche Norm als harmonisiert gilt, wird im Amtsblatt der EU veröffentlicht. Für die Erstellung der Normen sind auf europäischer Ebene das Komitee für Normierung (CEN) und das Komitee für elektronische Normung (CENELEC) zuständig. Erstellte Normen dieser Gremien erhalten die Kennung EN und sind in der Regel mit einer EU Richtlinie harmonisiert. Für den deutschen Raum ist die DIN (Deutsches Institut für Normung) und VDE (Prüf- und Zertifizierungsinstitut) zuständig. Zugelassene europäische Normen und erstellte Normen dieser Institute erhalten die Kennung DIN und sind für den Standort Deutschland zulässig.

³ BfArM (Bundesinstitut für Arzneimittel und Medizinprodukte): „Medizinproduktegesetz“. [\[IQ01\]](#) (Siehe im Literaturverzeichnis unter Internetquellen, S. 108)

⁴ EU Recht: „Richtlinie 93/42/EWG“. [\[IQ02\]](#)

Zwei Beispiele dafür sind:

- DIN EN 62304 - Medizingeräte-Software-Software-Lebenszyklus-Prozesse (harmonisiert)
- DIN EN 61508-3 - Funktionale Sicherheit sicherheitsbezogener elektrischer/ elektronischer/programmierbar elektronischer Systeme (nicht harmonisiert)

Für die Umsetzung dieser Arbeit wurden folgende harmonisierte Normen verwendet:

- DIN EN 60601-1⁵
- DIN EN 60601-1-4⁵
- DIN EN 62304⁵
- DIN EN 62366⁵
- DIN EN ISO 14971⁵

Als zentrale Richtlinie für Medizinprodukte gilt die 93/42/EWG. Sie wurde am 14. Juni 1993 verabschiedet und ist bis heute über weitere Richtlinien mehrmals ergänzt und verändert worden. Die meisten der angesprochenen Richtlinien nutzen die 93/42/EWG als Grundlage. Im Wesentlichen legt die Richtlinie 93/42/EWG (MDD - Medical Device Directive) die Rahmenbedingungen für das Zertifizierungsverfahren nach CE fest. Jedes im Europäischen Wirtschaftsraum zugelassene Medizinprodukt muss ein CE- Zertifikat besitzen.

2.2 Grundlegende gesetzliche Vorgaben seitens MPG

Das MPG gilt speziell für Deutschland und regelt - gleich wie die 93/42/EWG - das Zulassungsverfahren in Deutschland. Hauptgrundlage des MPG ist die EU Richtlinie 93/42/EWG. Zusammen mit der MPBetreibV⁶ und MPV⁷ stellt das MPG den gültigen Rahmen für das Inverkehrbringen, Betreiben und Anwenden von Medizinprodukten in Deutschland dar. Damit das MPG greift, muss ein Produkt über das MPG als Medizinprodukt identifiziert werden. Als Grundlage für die Identifikation dient die Zweckbestimmung des Produktes. Ein Produkt gilt als Medizinprodukt, wenn es eine der Bedingungen aus §3, Abs. 1 des MPG erfüllt:

⁵ Für vollständigen Titel, sowie Ausgabedatum der Norm siehe im Literaturverzeichnis unter Normen- und Richtlinienregister (S. 108).

⁶ Medizinprodukte - Betreiberverordnung

⁷ Medizinprodukte - Verordnung

„Medizinprodukte sind alle einzeln oder miteinander verbunden verwendeten Instrumente, Apparate, Vorrichtungen, Software, Stoffe und Zubereitungen aus Stoffen oder andere Gegenstände einschließlich der vom Hersteller speziell zur Anwendung für diagnostische oder therapeutische Zwecke bestimmten und für ein einwandfreies Funktionieren des Medizinproduktes eingesetzten Software, die vom Hersteller zur Anwendung für Menschen mittels ihrer Funktionen zum Zwecke

- a) der Erkennung, Verhütung, Überwachung, Behandlung oder Linderung von Krankheiten,*
- b) der Erkennung, Überwachung, Behandlung, Linderung oder Kom pensierung von Verletzungen oder Behinderungen,*
- c) der Untersuchung, der Ersetzung oder der Veränderung des anatomischen Aufbaus oder eines physiologischen Vorgangs oder*
- d) der Empfängnisregelung*

zu dienen bestimmt sind und deren bestimmungsgemäße Hauptwirkung im oder am menschlichen Körper weder durch pharmakologisch oder immunologisch wirkende Mittel noch durch Metabolismus erreicht wird, deren Wirkungsweise aber durch solche Mittel unterstützt werden kann.“

Erst seit kurzem wurde auch Software als eigenständiges Medizinprodukt durch die 2007/47/EG definiert. Zuvor galt Software nur dann als Medizinprodukt, wenn es in Verbindung mit klassischen Medizinprodukten stand (z.B. Steuersoftware CT). Der §3, Abs. 1 aus dem MPG wurde dementsprechend angepasst, sodass diese Regelung auch für Deutschland gilt.

2.2.1 Bestimmung als Medizinprodukt

Nach §3, Abs. 1 des MPG ist sichtbar, dass unser Produkt über Punkt a) (Erkennung, Verhütung, Überwachung, Behandlung oder Linderung von Krankheiten) als Medizinprodukt eingestuft werden könnte. Da unser Produkt die häusliche Medikamententherapie unterstützt, dient es zur Behandlung oder Linderung von Krankheiten. Nach weiteren Überlegungen könnte das Produkt auch in die Sparte der Infusionspumpen fallen, die abstrakt gesehen eine ähnliche Aufgabe erfüllen. §3, Abs. 3 des MPG besagt:

„Medizinprodukte sind auch Produkte nach Nummer 1, die als Bestandteil einen Stoff enthalten, der gesondert verwendet als Bestandteil eines Arzneimittels oder Arzneimittel aus menschlichem Blut oder Blutplasma im Sinne des Artikels 1 der Richtlinie 2001/83/EG des Europäischen Parlaments und des Rates vom 6. November 2001 zur Schaffung eines Gemeinschaftskodexes für Humanarzneimittel (ABl. L 311

vom 28.11.2001, S. 67), die zuletzt durch die Verordnung (EG) Nr. 1394/2007 (ABl. L 324 vom 10.12.2007, S. 121) geändert worden ist, betrachtet werden und in Ergänzung zu dem Produkt eine Wirkung auf den menschlichen Körper entfalten kann.“

Laut J. Uhlemann et al. fallen Infusionspumpen mit Hilfe 93/42/EWG Anhang IX in die Risikoklassen IIa/IIb⁸. Da unser Produkt auch Arzneimittel nach 2001/83/EG und §2 Abs. 1 des Arzneimittelgesetzes an den Patienten verabreicht, kann es über diesen Absatz als Medizinprodukt gelten.

Einen weiteren Gedanken zeigt die Betrachtung über die Zulassung von PDMS⁹ von A. Gärtner¹⁰ auf. Hier wird darauf eingegangen, dass auch Standalone Software, wie zum Beispiel KIS Systeme oder PDMS, seit dem 21.03.2010 als Medizinprodukt gelten.

Können wir unsere Steuersoftware auch als Standalone Software betrachten?

Laut A. Gärtner kann Standalone Software als Medizinprodukt gelten, wenn der Hersteller der Software eine medizinische Zweckbestimmung gemäß §3, Abs. 10 des MPG definiert. Das MPG definiert die Zweckbestimmung folgendermaßen:

„Zweckbestimmung ist die Verwendung, für die das Medizinprodukt in der Kennzeichnung, der Gebrauchsanweisung oder den Werbematerialien nach den Angaben des in Nummer 15 genannten Personenkreises bestimmt ist.“

Damit eine Standalone Software nach §3 MPG, Abs. 1 als Medizinprodukt eingestuft werden kann, muss sie beispielsweise eine Therapie unterstützen. A. Gärtner weist auf Seite 8 den PDMS die Eigenschaften *Unterstützung von Therapie und Diagnose* zu. Die PDMS sollen laut Gärtner damit einen medizinischen Zweck erfüllen und als Medizinprodukt gelten.

„Die Frage, ob bereits der Begriff Unterstützung die Definition eines PDMS als Medizinprodukt rechtfertigt, lässt sich dahingehend beantworten, dass Unterstützung ein übergeordneter Begriff ist, der vielfältige Formen beinhaltet. Die Unterstützung des Klinikers reicht von der reinen Präsentation von Daten bis hin zur automatisierte Be- und Verrechnung klinischer Daten mit assoziativen Trends, Entscheidungshilfen bis hin zur Planung von Therapiemaßnahmen anhand von hinterlegten Behandlungsleitlinien. Unterstützung bedeutet also, dass eine IT-Applikation wie ein PDMS in vielfältiger Weise dem Kliniker helfen kann,

⁸ J. Uhlemann et. al.: „Sicherheitsrelevante Aspekte der internationalen Normen für Infusionspumpen“. [\[UHLE2010\]](#) (Siehe im Literaturverzeichnis unter Literatur, S. 111)

⁹ Patientendatenmanagementsystem

¹⁰ A. Gärtner: „Patientendatenmanagementsysteme als Softwaremedizinprodukt? Eine regulatorische Betrachtung“. [\[IQ03\]](#)

diagnostische und therapeutische Entscheidungen zu treffen und umzusetzen. Die Studie der Herstellerunterlagen zeigt sehr deutlich, wie vielfältig PDM-Systeme diese Unterstützung im Sinne des klinischen Workflow durch zahlreiche Funktionalitäten durchführen bzw. erst ermöglichen.“

So gesehen unterstützt unser System ebenfalls die häusliche Medikamententherapie von Patienten und würde somit auch als Medizinprodukt gelten. Unser System würde laut A. Gärtner sogar noch als Medizinprodukt gelten, wenn nur die Software als Standalone-Erinnerungssystem erscheinen würde.

Ein Artikel von Dr. B. Jaeger¹¹ beschreibt diesbezüglich die Gesetzeslage auf Seite 1 folgendermaßen:

„Man hielt eine Klarstellung dahingehend für erforderlich, dass Software als solche, wenn sie spezifisch vom Hersteller für einen oder mehrere der in der Definition von Medizinprodukten genannten medizinischen Zwecke bestimmt ist, ein Medizinprodukt ist. Software für allgemeine Zwecke soll kein Medizinprodukt sein, auch wenn sie im Zusammenhang mit der Gesundheitspflege genutzt wird. Vor dem Hintergrund der zunehmenden Bedeutung von Software für Medizinprodukte – sei es als eigenständige Software oder als Bestandteil eines Medizinprodukts – soll auch die Validierung von Software in Übereinstimmung mit dem Stand der Technik zu den grundlegenden Anforderungen gehören. Eigenständige Software wird in der RL 93/42/EWG als aktives Medizinprodukt¹² klassifiziert. Allerdings wird Software, die ein Produkt steuert oder dessen Anwendung beeinflusst, automatisch derselben Klasse zugerechnet wie das Produkt selbst. Software, beispielsweise solche für computergestützte Implantologie, wird gemäß der Regel 12 von Anhang IX der Klasse I zugeordnet.“

Diese Interpretation untermauert die vorherigen Gedanken, dass unser Produkt als Medizinprodukt gilt. Da die Software außerdem als Steuersoftware einer Hardware gedacht ist, gilt für diese Software die Risikoklasse der Hardwarekomponente.

Ein weiterer Artikel von A. Gärtner¹³ greift einen weiteren wichtigen Gedanken für unser System auf. Der Artikel bestätigt die bisherigen Gedanken bei der Zulassung von Software als Medizinprodukt. Interessant ist hier aber die Hervorhebung von Betriebssystemen als Nichtmedizinprodukt. Da die Anwendung der Steuersoftware als Medizinprodukt auf einem Windows Betriebssystem gedacht ist, muss dies laut A. Gärtner explizit bei der Zweckbestimmung und der Zulassung mit beachtet werden:

¹¹ Dr. B. Jaeger. „Änderungen bei der Medizinproduktrichtlinie 93/42/EWG“. [[JAEG2008](#)]

¹² 93/42/EWG, Anhang IX, Punkt 1.4: „Medizinprodukt, dessen Betrieb von einer Stromquelle oder einer anderen Energiequelle abhängig ist.“

¹³ A. Gärtner. „IT und Medizinproduktrecht“. [[IQ04](#)]

„Durch die Interaktion von Betriebssystem als Nichtmedizinprodukt auf der Hardware-Ebene mit Software als Medizinprodukt für Diagnose und Therapie kommt ein IT-Vertriebsmitarbeiter ebenfalls automatisch mit der Medizintechnik und somit mit dem Medizinproduktegesetz in Berührung. Verkauft und installiert ein IT-Vertriebsmitarbeiter z. B. ein neues Betriebssystem für einen Rechner, auf dem ein Medizinprodukt Software läuft, so muss er wissen und den Betreiber informieren, dass damit die ursprüngliche Konformitätsbewertung des Herstellers erlischt, sofern dieser nicht grundsätzlich die Installation eines neuen Betriebssystem zugelassen hat. Dies gilt auch für andere Softwarekomponenten. Ein IT-Vertriebsberater muss also die Unterscheidungskriterien von Hardware und Software für den medizinischen und nichtmedizinischen Bereich kennen, wie er in den MEDDEV-Leitlinien 2.1/1 – April 1994 (Definition of medical devices) definiert ist.“

Zweckbestimmung des HomeCare Medidisp

Die Zweckbestimmung spielt in der Richtlinie 93/42/EWG und dem MPG eine wesentliche Rolle. Laut 93/42/EWG beschreibt die Zweckbestimmung:

„Verwendung, für die das Produkt entsprechend den Angaben des Herstellers in der Etikettierung, der Gebrauchsanweisung und/oder dem Werbematerial bestimmt ist.“

Die Zweckbestimmung dient als Grundlage für die:

- Definition der Inbetriebnahme nach 93/42/EWG¹⁴.
- Identifizierung der grundlegenden Anforderungen an das Medizinprodukt nach Anhang I¹⁵.
- Risikoklassifizierung nach Anhang IX¹⁶.

Zusätzlich kann die Zweckbestimmung in einer ersten Risikoanalyse zur Identifikation von ersten Risiken dienen. Neben der Definition des Verwendungszwecks ist die Zweckbestimmung aber auch in rechtlichen Fällen von *Zweckentfremdungen* zu gebrauchen. Zweckentfremdungen entstehen, wenn ein Betreiber das Medizinprodukt nicht gemäß der vorgeschriebenen Zweckbestimmung verwendet. Ein gutes Beispiel für solch einen Verstoß zeigt C. Harböck¹⁷:

„Der Betreiber verbindet das Medizinprodukt mit anderen Produkten, die nicht explizit vom Hersteller dafür aufgeführt sind.“

¹⁴ RL 93/42/EWG, Artikel 1, Abs. 2i. [\[IQ02\]](#)

¹⁵ RL 93/42/EWG, Artikel 3. [\[IQ02\]](#)

¹⁶ RL 93/42/EWG, Anhang IX, Punkt 2.1. [\[IQ02\]](#)

¹⁷ C. Harböck. „Richtlinienkonforme Softwareentwicklung von Medizinprodukten am Beispiel eines internetbasierten CTG Monitors mit LabVIEW“. [\[HARB2001\]](#), S. 13

Somit entsteht ein völlig neues Medizinprodukt, welches nicht CE zertifiziert ist. Geschieht dies, tritt der Betreiber sofort an die Stelle des Herstellers und kann dafür strafrechtlich belangt werden.

Ein wichtiges Merkmal der Zweckbestimmung ist die kurze, prägnante Aufführung sämtlicher Eigenschaften sowie sämtlicher Leistungen, die das Produkt leisten bzw. nicht leisten kann. Ein Beispiel für eine mögliche rechtlich konforme Gliederung einer Zweckbestimmung zeigt T. Creter et al.¹⁸ (Abbildung 4) und Dr. G Weisser¹⁹.

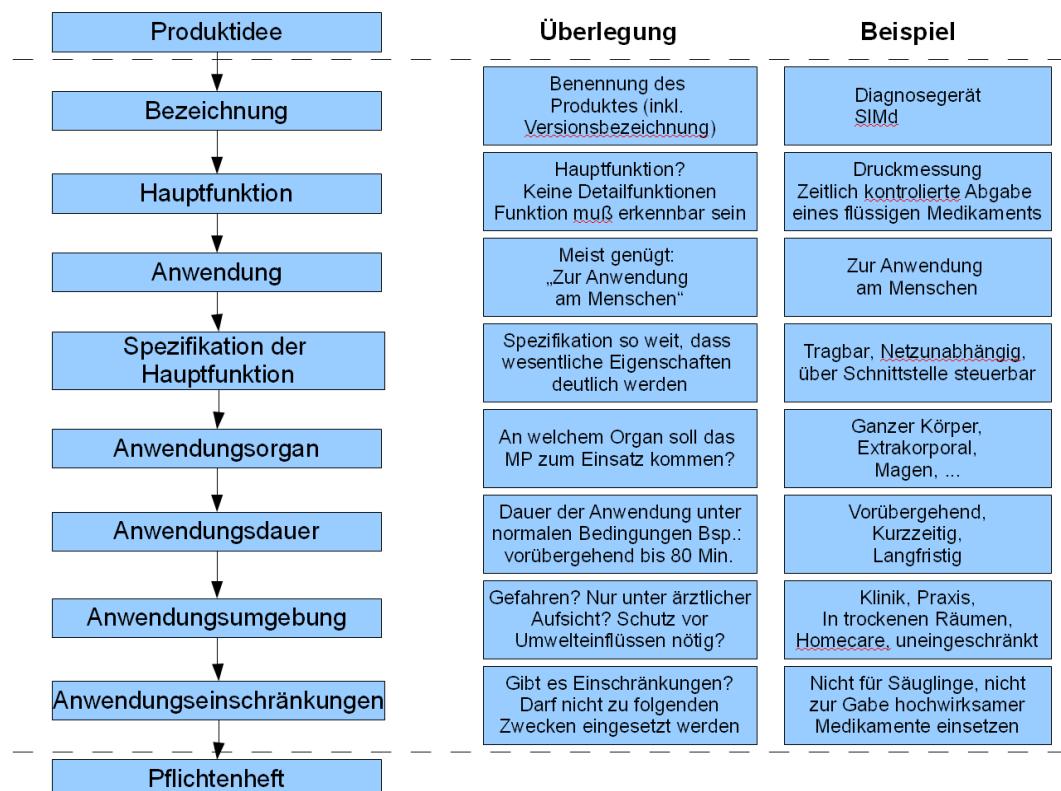


Abbildung 4: Rechtlich konforme Gliederung einer Zweckbestimmung²⁰

Laut dieser Gliederung könnte eine Zweckbestimmung für unser Produkt folgendermaßen aussehen:

„Der HomeCare Medidisp ist durch die elektronisch gesteuerte Abgabe von Arzneimitteln zur Unterstützung der häuslichen Therapie eines Patienten gedacht. Das System ist in eine Hardwarekomponente (HcMd-H 1.0) und eine Softwarekomponente (HcMd-S 1.0) geteilt. Es darf nur eine Kombination aus den beiden genannten Komponenten verwendet werden. Die Software HcMd-S 1.0 ist für die Steuerung der Hardware HcMd-H 1.0 und die darüber korrekte Arzneimedikation über einen langfristigen Zeitraum zuständig. Der Einsatzbereich des Systems liegt im

¹⁸ T. Creter et al. „Software als Medizinprodukt“. [CRHO1999]

¹⁹ Dr. G. Weisser. „Medizinische Zweckbestimmung für UMMPerfusion (OsiriX Plugin)“. [IQ05]

²⁰ Selbst erstellt nach T. Creter et al. [CRHO1999] und Dr. G. Weisser [IQ05]

häuslichen Umfeld bei älteren Patienten. Das System HomeCare Medidisp ist nicht für den Gebrauch Minderjähriger gedacht. Die Software HcMd-S 1.0 darf von einem unabhängigen Windows System (Windows XP und aufwärts) aus betrieben werden. Das PC System muss über eine Touchscreen Funktion verfügen. Die Ansteuerung des HomeCare Medidisp erfolgt über eine USB 2.0 Schnittstelle.“

Der erste Satz der HomeCare Medidisp Zweckbestimmung greift den Aspekt der Therapieunterstützung von A. Gärtner²¹ auf. Damit sollte unser Produkt ebenfalls als Medizinprodukt gelten können. Hinzu kommt, dass der HomeCare Medidisp nicht allein aus einer Standalone Software besteht, sondern aus der Kombination eines elektronischen Dispensers und deren Steuersoftware. Aufgrund dieser Gründe wird in den folgenden Schritten davon ausgegangen, dass die Software als Medizinprodukt nach §3, Abs. 1 des MPG einzustufen ist.

2.2.2 Relevante Vorgaben aus dem MPG

Ein wichtiger Begriff im MPG ist das Inverkehrbringen von Medizinprodukten. §3, Abs. 11 des MPG definiert *Inverkehrbringen* folgendermaßen:

„Inverkehrbringen ist jede entgeltliche oder unentgeltliche Abgabe von Medizinprodukten an andere. Erstmaliges Inverkehrbringen ist die erste Abgabe von neuen oder als neu aufbereiteten Medizinprodukten an andere im Europäischen Wirtschaftsraum. Als Inverkehrbringen nach diesem Gesetz gilt nicht

- a) die Abgabe von Medizinprodukten zum Zwecke der klinischen Prüfung,*
- b) die Abgabe von In-vitro-Diagnostika für Leistungsbewertungsprüfungen,*
- c) die erneute Abgabe eines Medizinproduktes nach seiner Inbetriebnahme an andere, es sei denn, dass es als neu aufbereitet oder wesentlich verändert worden ist.*

Eine Abgabe an andere liegt nicht vor, wenn Medizinprodukte für einen anderen aufbereitet und an diesen zurückgegeben werden.“

Der letzte Satz des Absatzes greift, wenn das ganze Produkt²² vor Inverkehrbringung veredelt werden muss. Betrachtet man den Abschnitt genauer, fällt auf, dass aber das Testen von Prototypen als Inverkehrbringung gelten kann und somit rechtswidrig ist. Für diesen Fall wurden im MPG klinische Prüfungen eingeführt. Klinische Prüfungen gelten nur für Deutschland und sind beim Testen von Prototypen Pflicht. Das MPG behandelt sämtliche rechtlichen Aspekte einer

²¹ A. Gärtner. [IQ03], S. 8

²² Beispielsweise ein Skalpell

klinischen Prüfung in einem gesonderten Abschnitt (Abschnitt 4). Aufgrund des Umfangs von klinischen Prüfungen, werden in dieser Arbeit klinische Prüfungen nicht weiter verfolgt.

MPG §6 regelt die grundlegenden Voraussetzungen für das Inverkehrbringen und die Inbetriebnahme von Medizinprodukten. Grundsätzlich gilt, dass nur Medizinprodukte mit gültiger CE-Kennzeichnung in Verkehr gebracht werden dürfen (Abs. 1). Laut Abs. 2 dürfen Medizinprodukte das CE-Kennzeichen tragen, wenn die Grundlegenden Anforderungen nach §7 – unter Berücksichtigung der Zweckbestimmung – erfüllt sind und das jeweils vorgeschriebene Konformitätsbewertungsverfahren nach 93/42/EWG von der in §37 Abs. 1 genannten Stelle durchgeführt worden ist.

Grundlegende Anforderungen

Nach §7 des MPG werden die grundlegenden Anforderungen für Medizinprodukte aus den dazugehörigen EU Richtlinien entnommen. Für klassische Medizinprodukte und Software gilt Anhang I, 93/42/EWG. Die folgende Aufzählung zeigt die wichtigsten Anforderungen an den HomeCare Medidisp. Eine vollständige Liste der Anforderungen ist in [ANHANG 10](#) zu finden.

1. Einhaltung Grundsätze der integrierten Sicherheit unter Berücksichtigung der allgemeinen anerkannten Standards.
2. Messskalen, Bedienungs- und Anzeigeeinrichtungen müssen ergonomischen Grundsätzen entsprechen.
3. Programmierbare Elektroniksysteme: Wiederholbarkeit, Zuverlässigkeit und Leistung gemäß Zweckbestimmung müssen gewährleistet sein.

Konformitätsbewertungsverfahren

Das Konformitätsbewertungsverfahren nach 93/42/EWG teilt sich in zwei Schritte auf. Erstens die Risikoklassifizierung des Produktes und zweitens die darauf basierenden Anforderungen an Dokumentation und Qualitätsmanagement für Entwicklung und Produktion. Auf Basis von Klassifizierungsregeln aus Anhang IX der Richtlinie 93/42/EWG erfolgt eine Risikoklassifizierung des Medizinproduktes. Zur Einstufung in die Klassifizierungsregeln soll das Produkt mit Hilfe der Zweckbestimmung kategorisiert werden. Ziel der Klassifizierung ist es, das Risikopotential des Produktes am menschlichen Körper zu ermitteln und einzustufen. Anhand der Risikoklasse wird im zweiten Schritt eine geeignete Entwicklungsmethodik zur Minimierung des ausgehenden Risikos gewählt. Laut 93/42/EWG, Anhang IX gibt es 4 Risikoklassen. Je höher die Risikoklasse, desto höher das ausgehende Risiko des Medizinproduktes und desto höhere Anforderungen gelten an das Qualitätsmanagement.

- Klasse I:
 - CE – Kennzeichnung
 - Grad der Verletzbarkeit durch diese Produkte ist gering
 - Konformitätsbewertungsverfahren für Produkte unter alleiniger Verantwortung des Herstellers
- Klasse IIa:
 - CE – Kennzeichnung mit Nummer der verantwortlichen Stelle
 - Grad der Verletzbarkeit durch diese Produkte ist erhöht
 - Beteiligung einer benannten Stelle für das Herstellungsstadium verbindlich
- Klasse IIb:
 - CE – Kennzeichnung mit Nummer der verantwortlichen Stelle
 - Grad der Verletzbarkeit durch diese Produkte ist hoch
 - Kontrolle durch eine benannte Stelle in Bezug auf die Auslegung der Produkte , sowie ihre Herstellung erforderlich
- Klasse III:
 - CE – Kennzeichnung mit Nummer der verantwortlichen Stelle
 - Grad der Verletzbarkeit durch diese Produkte ist hoch bis extrem hoch
 - Kontrolle durch eine benannte Stelle in Bezug auf die Auslegung der Produkte , sowie ihre Herstellung erforderlich
 - Beim Inverkehrbringen ist eine ausdrückliche vorherige Zulassung im Hinblick auf die Konformität erforderlich

Folgende Kriterien sollen nach 93/42/EWG Anhang IX, Kapitel 1 für die Kategorisierung des Medizinproduktes verwendet werden:

1. Dauer:
 - Vorübergehend: < 60 min
 - Kurzzeitig: <= 30 Tage
 - Langzeitig: > 30 Tage)
2. Ort der Anwendung:
 - Invasives/nicht invasives Produkt (Chirurgisch-invasiv/Implantat)
 - Wiederverwendbares chirurgisches Instrument
 - Nicht in Verbindung mit aktiven Medizinprodukten
3. Art der Anwendung:
 - Aktives Medizinprodukt
 - Betrieb von einer Stromquelle abhängig
 - Aktives therapeutisches/diagnostisches Medizinprodukt
4. Spezialfälle:
 - Anwendung im zentralen Kreislaufsystem/zentralen Nervensystem

Für den HomeCare Medidisp gelten auf Basis seiner Zweckbestimmung²³ folgende Kriterien:

1. Dauer: Langzeit (> 30 Tage)
2. Ort der Anwendung: nicht invasives Produkt
3. Art der Anwendung: aktives therapeutisches Medizinprodukt

Nach der Kategorisierung muss dem Medizinprodukt eine oder mehrere von 18 vorgegebenen Klassifizierungsregeln²⁴ zugeordnet werden. Sind für ein Medizinprodukt mehrere Klassifizierungsregeln anwendbar, wird immer die Regel mit der höchsten Risikoklasse gewählt. Bei Medizinprodukten mit verschiedenen Komponenten kann es vorkommen, dass die Komponenten verschiedene Risikoklassen besitzen. Auch hier wird immer die Regel mit der höchsten Risikoklasse gewählt. Die Klassifizierungsregeln sind in 4 Bereiche unterteilt. Dies betrifft nicht invasive, invasive und aktive Medizinprodukte sowie Sonderfälle. Der ANHANG 1 zeigt eine Zusammenfassung der Klassifizierungsregeln.

Die Klassifizierung unseres Produktes gestaltet sich anhand der Klassifizierungsregeln nicht eindeutig. Unser Medizinprodukt könnte als Kombination von Hard- und Software in Regel 11 fallen. Laut Emergo Group²⁵ gilt für Standalone Software die Regel 12:

“If the software is a medical device, it may be classified as Class I;”

Bei Kombinationen von Software und Hardware gilt laut Emergo Group immer die Risikoklasse des Hardwareparts:

“However, if the software medical device is an integral component of a device as indicated above, it assumes the classification of the device. For example, software that is part of a Class III medical device is viewed as a Class III device.”

Für den HomeCare Medidisp wurde, aufgrund der Kombination von Hardware und Software, Regel 11 gewählt. Er erhält somit die Risikoklassifizierung IIa.

Nach der Risikoklassifizierung muss die weitere Vorgehensweise, in Bezug auf Qualitätsmanagement und Risikominimierung, bestimmt werden. Je nach Risikoklasse und Ausgangslage des Herstellers wird dazu eine passende Vorgehensweise gewählt. Die verschiedenen Verfahren sind in Annex 2 – 7 geregelt und unterscheiden sich grundlegend anhand der Art der Qualitätssicherung und Risikominimierung. Prinzipiell gilt, je höher die Risikoklasse ist, desto stärker ist auf

²³ Für die Zweckbestimmung siehe Kapitel 2.2.1, S. 10

²⁴ 93/42/EWG, Anhang IX, Kapitel 3

²⁵ Emergo Group. „When is Software a medical device?“. [IQ06]

Qualität und Risikominimierung zu achten. Die Risikoklasse beeinflusst damit direkt die Kosten des Produktes. Richtlinie 93/42/EWG, Artikel 11 bestimmt, welcher Annex für die jeweiligen Risikoklassen gilt. Aus den Informationen aus Artikel 11 und nach C. Harböck²⁶ zeigt Abbildung 5.

Seit dem 01.02.2001 lösten die internationalen Normen DIN EN ISO 13485²⁷ und DIN EN ISO 13488²⁸ die beiden europäischen Normen DIN EN 46001, bzw. DIN EN 46002 ab. Die DIN EN ISO 13485 beinhaltet einen zusätzlichen Abschnitt für die Verwendung von Software bei der Prozesssteuerung und neu verfasste Abschnitte über Verpackung, Versand, Schulung und Statistik.

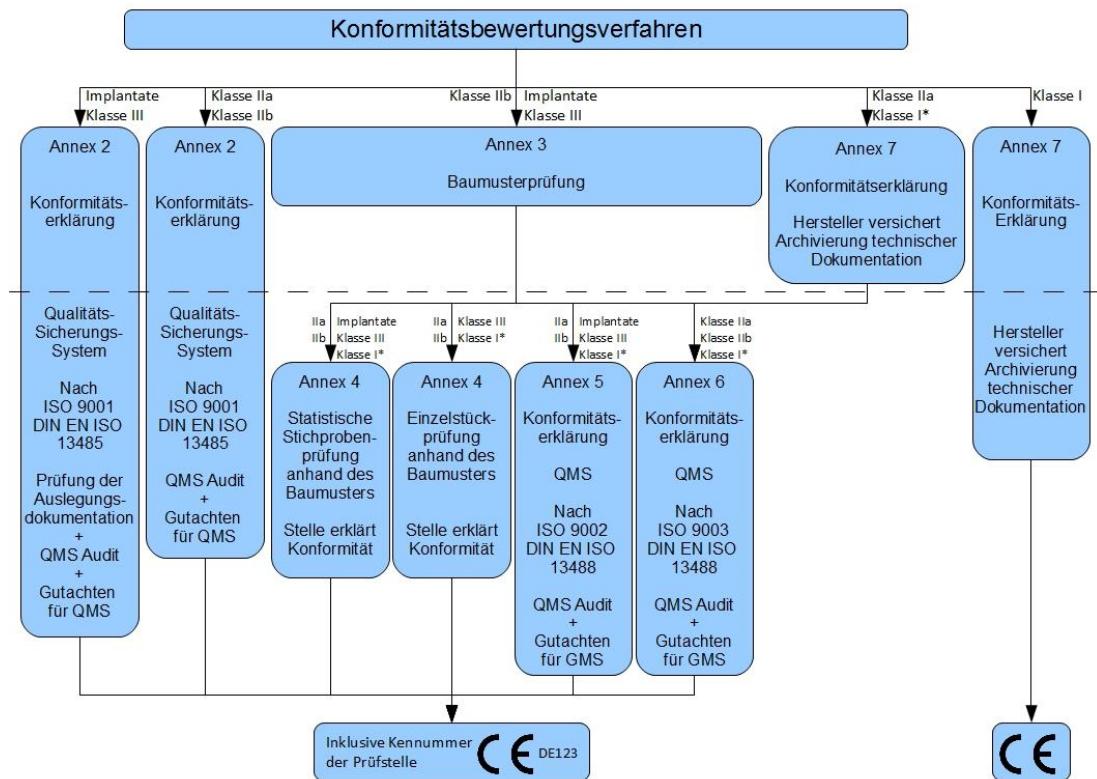


Abbildung 5: Ablauf Konformitätsbewertungsverfahren²⁹

Die gestrichelte horizontale Linie in Abbildung 5 trennt die Prozessschritte *Entwicklung* (oben) und *Herstellung* (unten). Durch diese Aufteilung sind flexible Möglichkeiten für Firmen ohne eigene Entwicklung/Herstellung oder ohne eigenes QMS³⁰ nach DIN EN ISO 13485 gegeben. Grundsätzlich gilt für Medizinprodukte der Klasse I eine Konformitätserklärung nach Annex 7. Dies bedeutet für den Hersteller,

²⁶ C. Harböck. [HARB2001], S. 17

²⁷ Medizinprodukte - Qualitätsmanagementsysteme - Anforderungen für regulatorische Zwecke

²⁸ Ähnlich 13485, für Hersteller ohne eigene Entwicklungsphase

²⁹ Nach C. Harböck. [HARB2001]: Die älteren Normen DIN EN 46001 und DIN EN 46002 wurden durch die aktuelleren Normen DIN EN ISO 13485 und DIN EN ISO 13488 ersetzt.

³⁰ Qualitätsmanagementsystem

dass lediglich die technische Dokumentation des Produktes beim Hersteller hinterlegt werden muss. Annex 7 gilt zusätzlich für Produkte der Klasse IIa und I* (sterile Produkte oder Produkte mit Messfunktion) in der Entwicklungsphase. In der Herstellungsphase müssen Annex 4, 5 oder 6 erfüllt werden.

Wenn der Hersteller ein vollständiges QMS nach DIN EN ISO 13485 besitzt, kann eine Konformität der Produkte mit der Risikoklasse IIa, IIb, und III über Annex 2 erklärt werden. Für Produkte der Klasse III gilt zusätzlich, dass eine Prüfung der Auslegungsdokumentation³¹ von einer benannten Stelle durchgeführt werden muss.

Besitzt der Hersteller kein QMS oder nur ein QMS nach DIN EN ISO 13488, bietet sich für Produkte der Risikoklasse III und IIb die Konformitätserklärung über eine Baumusterprüfung nach Annex 3 und folgend Annex 4, 5, 6 an. Bei einer EG Baumusterprüfung wird am QMS des Herstellers keine Auditierung durchgeführt. Die benannte Prüfstelle bescheinigt muss die Konformität des Baumusters überprüfen und stellt eine EG-Baumusterprüfbescheinigung auf das Baumuster aus.

Der Weg über die Baumusterprüfung und anschließend über Annex 4 bildet, besonders bei Software, einen wesentlichen Nachteil. Bei Annex 4 muss der Hersteller gewährleisten, dass das hergestellte Produkt mit dem Baumuster übereinstimmt. Das bedeutet für die Softwareherstellung, dass die benannte Stelle das fertige Installationsmedium mit dem Baumuster (z.B. bitweise) direkt vergleicht. Hat eine benannte Stelle ein Produkt als konform anerkannt, wird von der benannten Prüfstelle eine Konformitätserklärung für das Produkt ausgestellt. Der Hersteller darf nach erfolgreicher Konformitätserklärung das CE-Kennzeichen nach Artikel 17, 93/42/EWG an sein Produkt anbringen und in Verkehr bringen. Bei den Konformitätsbewertungsverfahren, bei denen zusätzlich eine benannte Prüfstelle involviert war (Annex 2 – 6), wird zusätzlich neben der CE-Kennzeichnung die Kennung der benannten Stelle angebracht. Wird der Weg über Annex 4 eingeschlagen und die benannte Prüfstelle erklärt die Konformität über eine Einzelstückprüfung, wird das CE-Kennzeichen mit Kennung von der benannten Prüfstelle angebracht.

³¹ Technische Dokumentation, beinhaltet alle relevanten Daten des Produktes

Für jedes zertifizierte Produkt muss der Hersteller außerdem folgende Unterlagen mindestens 5 Jahre archivieren.

1. Konformitätserklärung
2. Dokumentation QMS System (Annex 5)
3. Unterlagen über vorgenommene Änderungen am QMS System
4. Technische Dokumentation
5. EG-Baumusterprüfungsbescheinigung
6. Im Rahmen des Konformitätsbewertungsverfahrens Berichte/
Entscheidungen der benannten Stelle

2.3 Konkrete Anforderungen an das Design (Harmonisierte Normen)

Neben den allgemeinen Anforderungen, die in Anhang I der Richtlinie 93/42/EWG definiert sind, kann der Hersteller zur Unterstützung harmonisierte Normen verwenden. Bei harmonisierten Normen wird davon ausgegangen, dass sie den momentanen allgemeinen Stand der Technik repräsentieren. 93/42/EWG, Anhang I, Punkt 2 besagt:

„Die vom Hersteller bei der Auslegung und der Konstruktion der Produkte gewählten Lösungen müssen sich nach den Grundsätzen der integrierten Sicherheit richten, und zwar unter Berücksichtigung des allgemein anerkannten Standes der Technik.“

Somit wird von der Richtlinie indirekt festgeschrieben, dass harmonisierte Normen möglichst zu verwenden sind. Die Grafik aus Abbildung 6 verdeutlicht die Rechtslage. Verwendet der Hersteller keine harmonisierten Normen für die Entwicklung, liegt die Beweislast für die Einhaltung des momentanen Standes der Technik beim Hersteller. Werden hingegen harmonisierte Normen verwendet, wird davon ausgegangen, dass eine konforme Herstellung des Produktes geschieht. Die Beweislast liegt dann nicht mehr beim Hersteller.

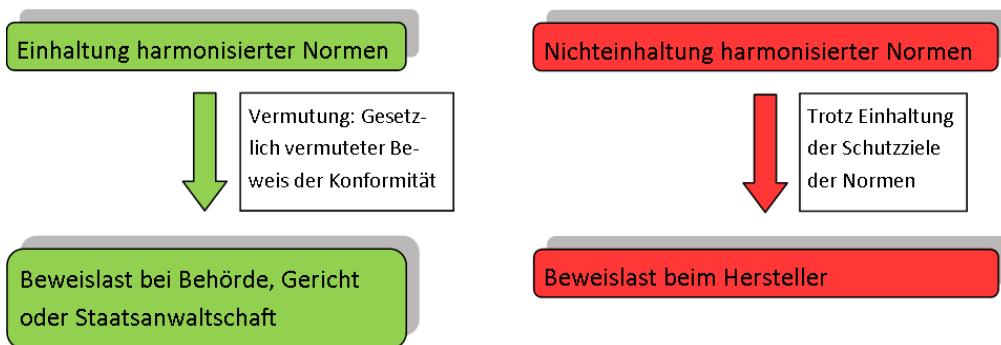


Abbildung 6: Beweislast bei Verwendung harmonisierter Normen

In Kapitel 2.1 wurden sämtliche Normen genannt, die in dieser Arbeit verwendet werden. Die Gebrauchstauglichkeitsnorm DIN EN 62366 ist laut S. Donadel³² seit dem 21.03.2010 bei der Herstellung von Medizinprodukten Pflicht. Ähnlich verhält es sich mit den anderen genannten Normen³³. Diese werden mittlerweile als gängiger Standard bei der Entwicklung von Medizinprodukten angesehen. Für die Entwicklung des Hardwareparts unseres Produktes wäre zusätzlich die Norm DIN EN 60601-1-1 „*Medizinische elektrische Geräte - Teil 1-1: Allgemeine Festlegungen für die Sicherheit – Ergänzungsnorm: Festlegungen für die Sicherheit von medizinischen elektrischen Systemen*“ von Bedeutung. Weitere wichtige harmonisierte Normen sind die DIN EN ISO 13485 und DIN EN ISO 13488. Diese beiden Normen beschreiben Qualitätsmanagementsysteme für die Herstellung von Medizinprodukten und sind bei einem Konformitätsbewertungsverfahren ab Klasse IIa relevant³⁴.

2.3.1 Sicherheitsvorgaben aus DIN EN 60601-1

Wie der Name der DIN Norm³⁵ bereits sagt, beschäftigt diese Norm sich mehr mit Leistungsmerkmalen für elektrische Geräte. Für Software ist diese Norm kaum von Interesse.

Ein Punkt muss jedoch beachtetet werden: Die maximal erlaubte Stärke der Ableitströme (maximal 0,5 mA) in unmittelbarer Patientenumgebung. Diese Begrenzung der Ableitströme zieht eine besondere Problematik mit sich. Laut einer Produktvorstellung von HJP Datentechnik GmbH³⁶, erfüllen normale PC-Systeme die Anforderung an Ableitströme nicht. Bei diesen Systemen liegt die Grenze für Ableitströme bei 3,5 mA. Grund ist die Konstruktion solcher PC-Systeme nach einer nicht harmonisierten Norm. Bei einer Kombination von einem Nichtmedizinprodukt und einem Medizinprodukt – wie es im dem Fall unseres Produktes gedacht ist – muss streng genommen die 0,5 mA Grenze eingehalten werden. Eine Kombination von einem handelsüblichen PC-System mit unserer Steueroftware ist demnach nicht möglich.

Die Norm DIN EN 60601-1-1 schlägt für die Lösung des Problems einen Trenntransformator vor. Eine andere Lösung wäre es, direkt ein konformes PC-System nach DIN EN 60601-1 für die Kombination zu verwenden.

³² S. Donadel. „*Die Norm DIN EN 62366 tritt ab März in Kraft*“. [\[IQ07\]](#)

³³ DIN EN 60601-1, DIN EN 60601-1-4, DIN EN 62304, DIN EN ISO 14971

³⁴ Siehe Kapitel 2.2.2, Abbildung 5, S. 15

³⁵ Medizinische elektrische Geräte; Teil 1: Allgemeine Festlegungen für die Sicherheit einschließlich der wesentlichen Leistungsmerkmale

³⁶ HJP Datentechnik GmbH. „*HJP Medical Line Produktvorstellung*“. [\[IQ08\]](#)

Durch die jüngsten Änderungen der Richtlinie 93/42/EWG, bei denen Standalone Software auch als Medizinprodukt gilt, können streng genommen keine handelsüblichen PC-Systeme mehr verwendet werden. Somit würden Praxiscomputersysteme, KIS-Systeme und andere Standalone Software rechtlich nur auf zugelassenen Kombinationen anwendbar sein³⁷.

Ein weiterer Aspekt aus der DIN EN 60601-1 ist der Fokus auf Schnittstellen des Systems. In unserem System ist eine Kommunikation zwischen zwei Subsystemen angedacht. Zum einen der elektronische Dispenser und zum anderen dessen Steuersoftware, welche auf einer separaten Hardware arbeiten soll. Die Schnittstelle für die Kommunikation muss den Grundlagen dieser Norm entsprechen. Es wird in der Norm darauf hingewiesen, dass Störungen durch ungewollte Spannungen an der Schnittstelle auszuschließen sind. Es müssen korrekte Angaben über Volt- und Frequenzbereich der Schnittstelle bekannt sein. Von der Norm wird außerdem eine eindeutige Kennzeichnung der Schnittstellen vorgeschrieben. Es müssen immer standardisierte Kennzeichnungen für bestimmte Funktionen des Gerätes verwendet werden (Beispielsweise für Ein- und Ausgänge, USB, usw.).

2.3.2 Vorgaben aus DIN EN 60601-1-4 und DIN EN 62304

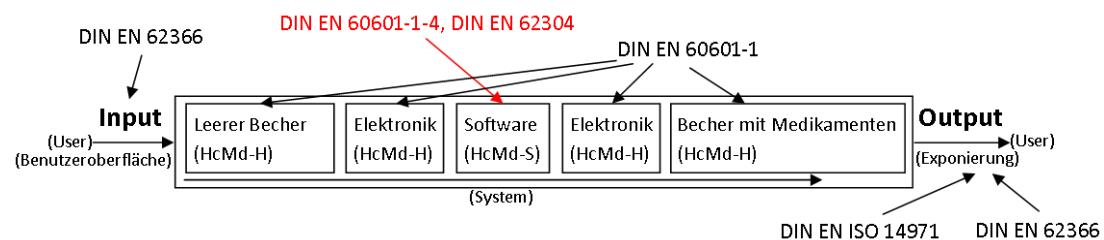


Abbildung 7: Wo setzen die Normen an? Nach Prof. Dr. J. Stettin [IQ09], S. 2

Wie in Abbildung 7 zu sehen ist, greifen die beiden Normen DIN EN 60601-1-4 und DIN EN 62304 direkt bei der Software - genau genommen bei Entwicklung und Design.

DIN EN 60601-1-4

Die Ergänzungsnorm DIN EN 60601-1-4 behandelt im Wesentlichen zwei Hauptthemen. Die Risikominimierung und die Festlegung von Anforderungen für einen Designprozess von Programmierbaren elektrisch medizinischen Systemen (PEMS). Ein PEMS ist laut Norm:

„Ein Medizinisches Elektrisches Gerät oder Medizinisches Elektrisches System, mit einem oder mehreren Programmierbaren elektronischen Subsystemen (PESS).“

³⁷ Beispielsweise im Tresen Bereich einer Praxis

Das programmierbare elektronische Subsystem (PESS) definiert die Norm folgendermaßen:

„Ein System basierend auf einer oder mehreren CPUs einschließlich deren Software und Schnittstellen.“

Der Risikomanagementteil dieser Norm wird durch die DIN EN ISO 14971³⁸ komplett ersetzt. Die Risikomanagementnorm DIN EN ISO 14971 ist laut 93/42/EWG und MPG Pflicht.

Soweit anwendbar, muss das zu entwickelnde System in Subsysteme, beispielsweise PESS, zerlegt werden³⁹. Für jedes Subsystem muss eine eigene Anforderungsspezifikation ausgearbeitet werden. Design und Umsetzung der einzelnen PESS Systeme können in der Entwicklungsphase parallel verlaufen. Abbildung 8 zeigt den ersten Ansatz des strukturellen Aufbaus unseres Medizinproduktes als PEMS/PESS-System. Das rechte PESS (Steuermodul) beinhaltet die Steueroftware (HcMd-S), die Controllersoftware (HcMd-CS) und das PC-System, auf dem sie laufen. Das linke PESS beinhaltet das Hardwaremodul, inklusive Elektronik und Mechanik.

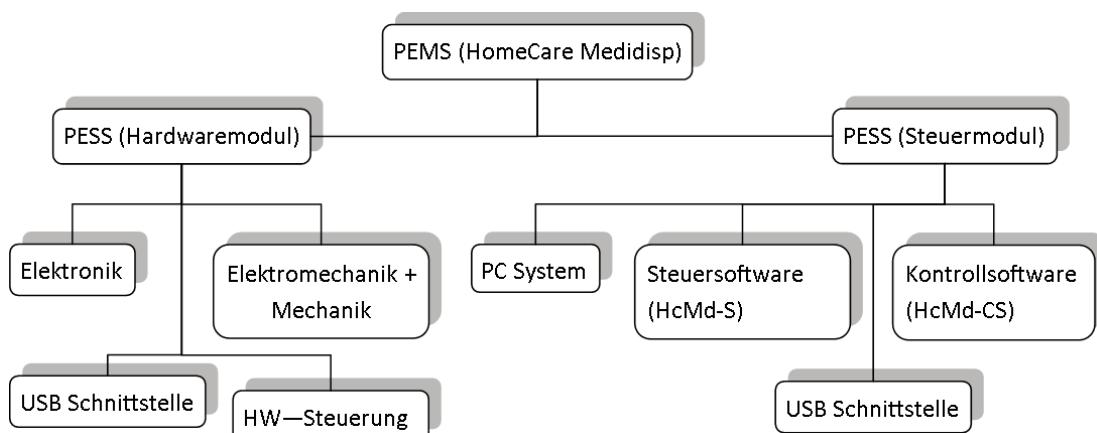


Abbildung 8: PEMS / PESS Systemvorschlag (selbst erstellt)

Für die Entwicklung des Medizinproduktes schlägt die Norm in Anhang DDD beispielhaft ein Entwicklungs-Lebenszyklus-Modell vor. Der Entwicklungs-Lebenszyklus beinhaltet laut Norm die Zeitspanne von Konzeption bis hin zur Validierung des Medizinproduktes. Das vorgeschlagene Modell aus der Norm beschreibt somit eine richtlinienkonforme Entwicklungsmethodik. Die eigens gewählte Entwicklungsmethodik muss laut Norm folgende Eigenschaften aufweisen:

³⁸ bayoonet AG. „ISO 14971 – Risikomanagement für Medizinprodukte und In-vitro-Diagnostika mit dem Qware Riskmanager“. [\[IQ10\]](#)

³⁹ Punkt 52.208.1, S. 13

- Sie muss in Phasen und Aufgaben unterteilt sein, mit jeweils klar definierten Eingängen, Ausgängen und Tätigkeiten.
- Sie muss integrale Prozesse für das Risikomanagement besitzen.
- Sie muss Anforderungen für die Dokumentation enthalten.
- Das Risikomanagement muss sich über den gesamten Entwicklungs-Lebenszyklus erstrecken.

Abbildung 9 zeigt die grafische Darstellung der vorgeschlagenen Entwicklungs-methodik aus der DIN EN 60601-1-4. Die Methodik erinnert an das internationale V-Modell für IT-Systeme. Der sofort erkennbare Unterschied zum V-Modell ist, dass nicht sämtliche Phasen iterativ wiederholt werden. In der vorgeschlagenen Methodik werden lediglich die Design-, Implementierungs-, und Testphasen (Phasen 3-6) iterativ wiederholt. Auffällig ist in Abbildung 9 die einmalige Risiko-Analyse in Phase 2. Risiko-Analyse und Beherrschungsmaßnahmen werden in der Modellgrafik nicht weiter erwähnt, sollen aber laut Tabelle DDD.1 der Norm in den Phasen 1-3 durchgeführt werden. Die Risiko-Analyse, inklusive Beherrschungsmaßnahmen, wird somit immer zum Anfang der vorgeschlagenen iterativen Vorgehensweise durchgeführt (Phase 3). Bei Anwendung der vorgeschlagenen Methodik umfassen die Phasen 3-4 (Design/Implementierung) außerdem die Auswahl von:

- Software-Entwicklungsverfahren
- Elektronischen Bauteilen
- Werkzeuge der rechnerunterstützten Software-Entwicklung (CASE)
- Redundanter Hardware
- Mensch-/PEMS-Schnittstelle
- Energiequellen
- Umgebungsbedingungen
- Programmiersprachen
- Software von Fremdanbietern

In den Phasen 4-7 wird das System, angefangen bei den Modulen, durch Tests verifiziert. Eine Validierung des PEMS als Ganzes wird in der finalen Phase durchgeführt.

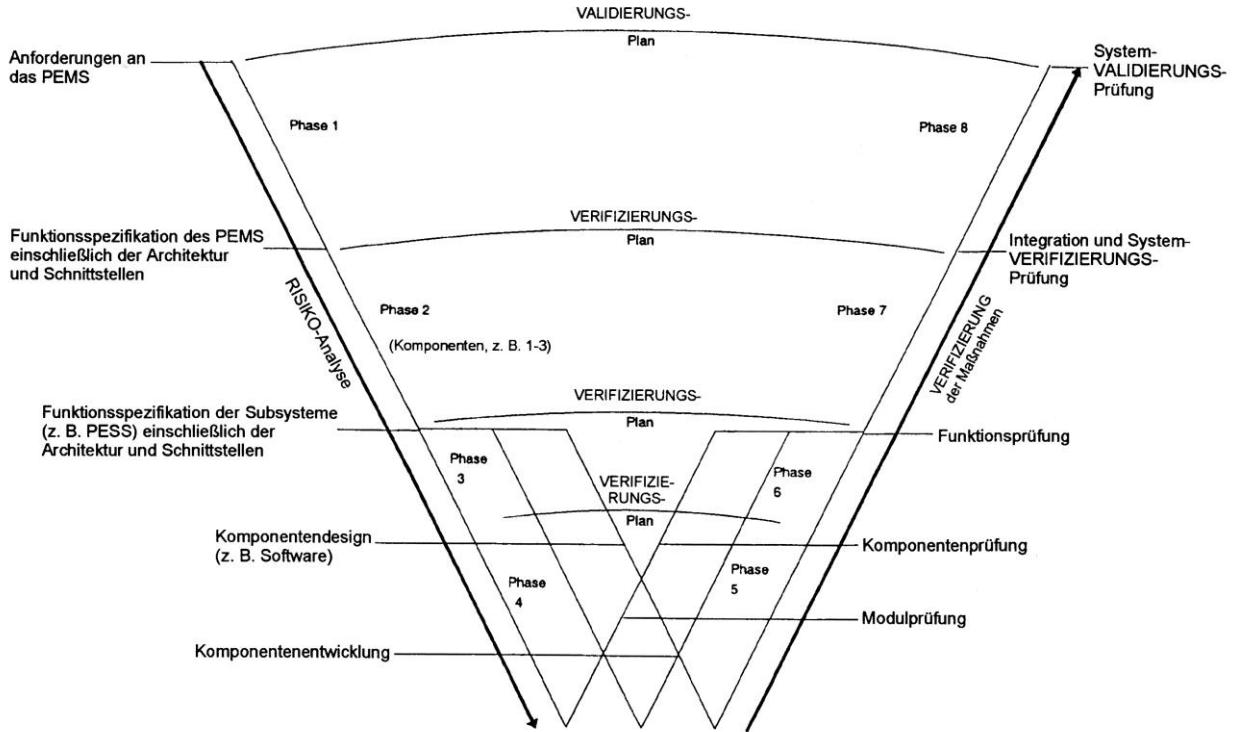


Abbildung 9: Entwicklungs-Lebenszyklus-Modell nach DIN EN 60601-1-4

DIN EN 62304

Die Norm DIN EN 62304 legt ihren Fokus direkt auf die Software und dessen Entwicklungsprozess. Der Einführungsbeitrag⁴⁰ zur Norm besagt folgendes:

„Diese Norm definiert Anforderungen an den Lebenszyklus von Medizinprodukte-Software. Die Zusammenstellung von Prozessen, Aktivitäten und Aufgaben, die in dieser Norm beschrieben werden, legt einen allgemeinen Rahmen für Lebenszyklus-Prozesse von Medizinprodukte-Software fest. Diese Norm gilt für die Entwicklung und Wartung von Medizinprodukte-Software, wenn die Software selbst ein Medizinprodukt ist oder wenn die Software ein eingebetteter oder integraler Bestandteil des fertigen Medizinprodukts ist. Sie deckt nicht Validierung und abschließende Freigabe des MEDIZINPRODUKTS ab, auch dann nicht, wenn das MEDIZINPRODUKT gänzlich aus Software besteht.“

Es werden drei wichtige Punkte aus der Norm kurz vorgestellt. Laut DIN EN 62304 muss jede Software eine Sicherheitsklassifizierung erhalten. Ist keine Zuordnung erfolgt, wird der Software vorerst die Sicherheitsklasse C zugeordnet. Anders als bei dem üblichen Risikomanagementprozess - bei dem das Risiko mit Hilfe von Wahrscheinlichkeit und Schadensgrad identifiziert wird - regelt die DIN EN 62304 die Klassifizierung der Sicherheitsklasse nur über den Schadensgrad. Grund dafür ist,

⁴⁰ beuth.de. „Einführungsbeitrag zur Norm DIN EN 62304“. [IQ11]

dass die Norm davon ausgeht, dass bei Software früher oder später ein Fehler auftritt:

„Wenn die Gefährdung davon herrühren könnte, dass die Software sich nicht gemäß ihrer Zweckbestimmung/Spezifikation verhält, muss die Wahrscheinlichkeit als 100 Prozent angenommen werden.“

Die DIN EN 62304 definiert die Schadensklassen folgendermaßen:

- Klasse A: Keine Verletzung oder Schädigung der Gesundheit möglich
- Klasse B: Keine schwere Verletzung möglich
- Klasse C: Tod oder schwere Verletzung möglich

Nach dieser Definition ist unsere Software, in Verbindung mit gefährlichen Medikamenten, definitiv der Klasse C zuzuordnen. Für jede Sicherheitsklasse gelten laut Norm zusätzliche Bestimmungen für das Risikomanagement. Nach Prof. Dr. J. Stettin⁴¹, gelten folgende zusätzliche Bestimmungen bei den Sicherheitsklassen B und C:

- Soweit angemessen, muss der Hersteller mögliche softwareseitige Ursachen, die zu einer Gefährdungssituation beitragen können, betrachten.
- Wenn ein Ausfall oder unerwartete Ergebnisse einer SOUP⁴² eine mögliche Ursache zu einer Gefährdungssituation sind, muss der Hersteller Listen mit Anomalien für die verwendete Version der SOUP-Komponente veröffentlichen.
- Der Hersteller muss die im Medizinprodukt verwendete SOUP evaluieren, um festzustellen, ob irgendeine der bekannten Anomalien von Ereignissen zu einer Gefährdungssituation führen könnte.
- Risikokontrollmaßnahmen müssen als Software-Anforderung aufgenommen werden.
- Risikokontrollmaßnahmen müssen verifiziert und dokumentiert werden.

Die DIN EN 62304 legt Augenmerk auf die sogenannten SOUP Komponenten eines Medizinproduktes. SOUPS können Bibliotheken, Module oder sogar Betriebssysteme sein. Damit zeigt sich ein nächster kritischer Punkt unseres Medizinproduktes. Die Anwendung der Software (HcMd-S) und der Kontrollsoftware (HcMd-CS) ist auf den Betriebssystemen von Windows XP, 7 und aufwärts geplant. Abbildung 10 zeigt, dass eine komplette Abdeckung durch eigene Tests bei einem Betriebssystem wie Windows XP nahezu unmöglich ist.

⁴¹ Prof. Dr. J. Stettin. „Risikoklassifizierung von Software - Umsetzung des Risikomanagements für Software nach DIN EN 62304 und IEC/TR 80002-1“. [\[1Q09\]](#), S. 12ff

⁴² Software of Unknown Provenance

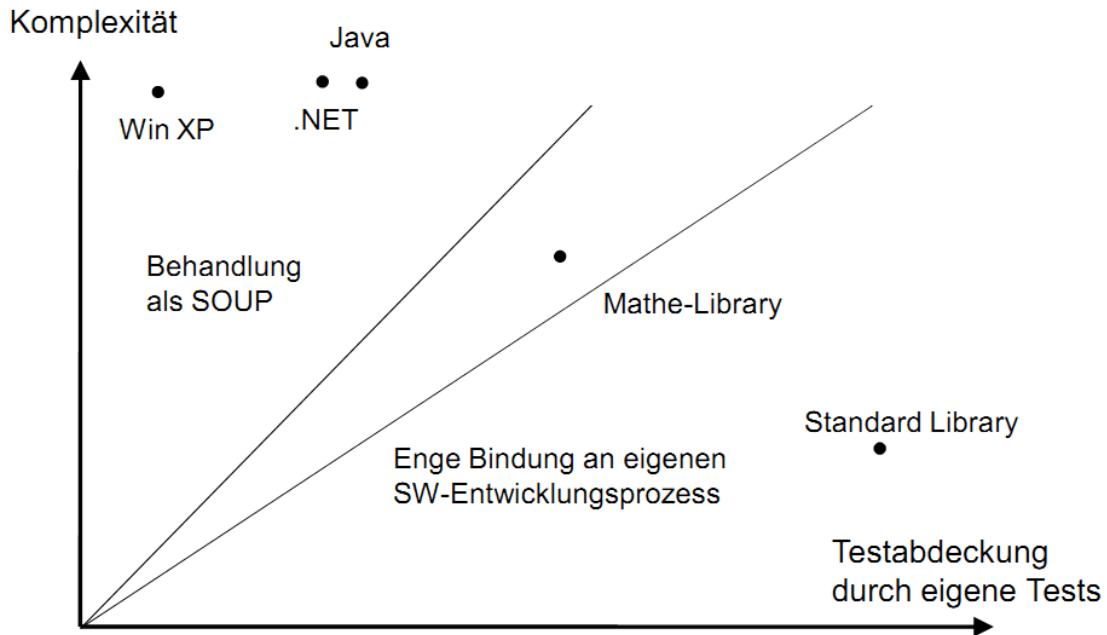


Abbildung 10: Komplexität versus Testabdeckung, Dr. C. Lindner [IQ12], S. 23

Bietet es sich also an, SOUPS möglichst zu umgehen, sollte man dies auch tun. Typische Gefährdungen, die durch SOUPS entstehen können, sind laut Dr. C. Lindner⁴³:

- SOUP reagiert anders als spezifiziert.
- SOUP weist unbekannte Anomalie auf.
- SOUP fällt aus.

Grund für eine unbekannte Anomalie kann bei einem Betriebssystem beispielsweise das vom Anwender unkontrollierte Installieren von Updates und Bugfixes sein. Jede SOUP muss, genauso wie die Software, eine Sicherheitsklassifizierung erhalten und unterliegt den strengerer Anforderungen dieser Norm.

Neben Klassifizierung und Anforderungen an das Risikomanagement, stellt die DIN EN 62304 außerdem ein Entwicklungs-Lebenszyklus Modell (Abbildung 11) als Entwicklungsmethodik für eine Software vor. Diese Methodik nimmt das vorgeschlagene Modell aus der DIN EN 60601-1-4 als Grundlage⁴⁴. Lediglich der gelb umrahmte Teil wurde für die Entwicklung von Software als PESS oder Subsystem verändert. Anstelle von Komponentendesign, Komponentenentwicklung, Modulprüfung und Komponentenprüfung sieht die DIN EN 60601-1-4 in dieser Phase die Punkte Software-Architektur, Software-Design, Softwarecode Verifikation und Softwareverifikation vor.

⁴³ Dr. C. Lindner. „Fremdsoftware in Medizinprodukten. Welche Anforderungen sind zu beachten?“. [IQ12], S. 8

⁴⁴ Siehe Abbildung 9, S. 22: Phase 1-3, PEMS/PESS Entwicklung

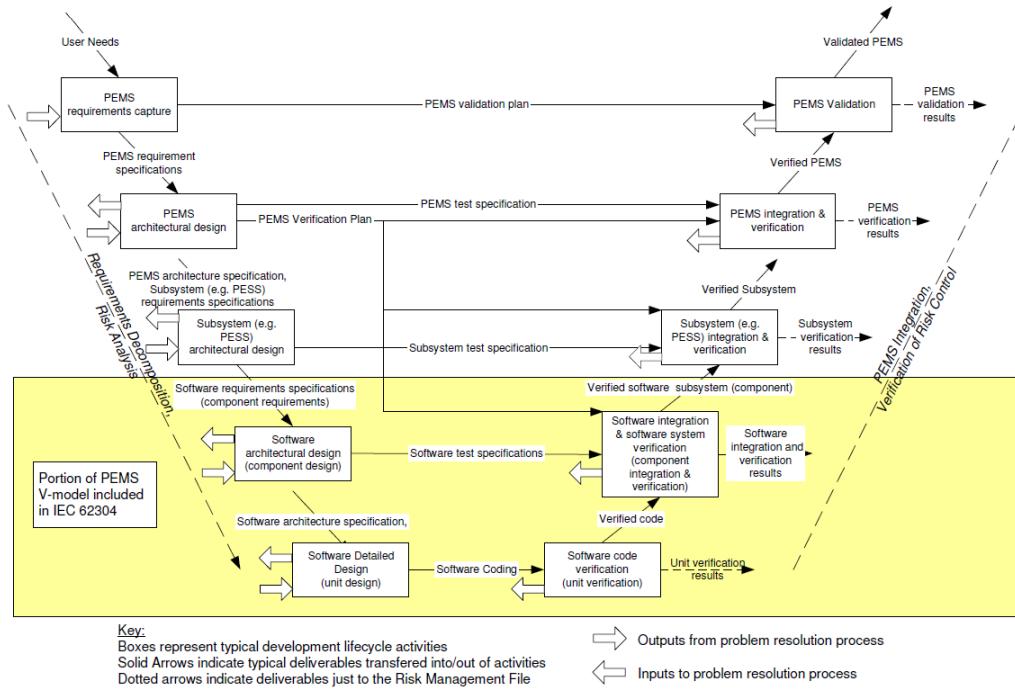


Abbildung 11: Entwicklungs-Lebenszyklus-Modell nach DIN EN 62304

2.3.3 Vorgaben an das Risikomanagement aus DIN EN ISO 14971

Der gesamte Prozess des Risikomanagements wird in der DIN EN ISO 14971 beschrieben. Das RM nach DIN EN ISO 14971 umfasst Analyse, Bewertung und Beherrschung des Risikos in iterativen Phasen, sowie die Bewertung des Restrisikos in einer Risiko-Nutzen-Analyse. Wie bereits bei der DIN EN 60601-1-4 erwähnt, erstreckt sich das Risikomanagement im Entwicklungsprozess über mehrere Phasen und Iterationen. Die iterativen Elemente sind dabei die Risiko-Analyse und Risiko-beherrschungsmaßnahmen in den Softwareentwicklungsphasen 3-6. Prinzipiell gilt, dass zum Ende der Designphase und vor Beginn der Implementierungsphase eine Risiko-Analyse, Bewertung und auf die erkannten Risiken Risikobeherrschungsmaßnahmen durchgeführt werden. Die ergriffenen Maßnahmen haben meist direkten Einfluss auf das Design.

Das wichtigste Dokument des Risikomanagements ist der Risikomanagement-Plan. Er legt die Rahmenbedingungen für das Risikomanagement fest und somit auch für Risiko-Analyse, Bewertung und Maßnahmen. Der Plan enthält folgende Punkte:

- Aufgabengebiet des Plans, Identifizierung und Beschreibung des Medizinprodukts und der Phase des Lebenszyklus, für die jedes Element des Plans anwendbar ist
- Zuordnung von Verantwortlichkeiten und Befugnissen
- Anforderungen an die Überprüfung von Tätigkeiten des Risikomanagements

- Kriterien für die Vertretbarkeit von Risiken einschließlich der Kriterien für die Vertretbarkeit von Risiken, wenn die Wahrscheinlichkeit des Auftretens eines Schadens nicht abgeschätzt werden kann
- Tätigkeiten zur Verifizierung
- ein oder mehrere Verfahren zur Gewinnung sachdienlicher Informationen aus den der Produktion nachgelagerten Phasen

Der Risikomanagement-Prozess ist nicht wesentlicher Bestandteil der Arbeit, deshalb wird nicht näher darauf eingegangen. Lediglich auf Risiko-Analyse, Bewertung und Maßnahmen wird näher eingegangen.

Risiko-Analyse nach DIN EN ISO 14971

Die Risiko-Analyse teilt sich in 3 Schritte auf:

1. Bestimmungsgemäßer Gebrauch und Feststellung von Merkmalen, die sich auf die Sicherheit des Medizinprodukts beziehen
2. Feststellung von Gefährdungen
3. Abschätzung des Risikos bzw. der Risiken für jede Gefährdungssituation

Um den bestimmungsgemäßen Gebrauch und Merkmale zu ermitteln, stellt die Norm in Anhang C bestimmte Beispielfragen zur Verfügung. Nachfolgend sind einige Beispielfragen aufgelistet:

- Welches ist der bestimmungsgemäße Gebrauch und wie soll das Medizinprodukt angewendet werden?
- Soll das Medizinprodukt implantiert werden?
- Ist es vorgesehen, dass das Medizinprodukt mit dem Patienten oder anderen Personen in Berührung kommt?
- Welche Werkstoffe oder Bauteile sind in das Medizinprodukt integriert oder werden zusammen mit dem Medizinprodukt verwendet oder kommen in Berührung mit ihm?
- Wird dem Patienten Energie zugeführt oder entzogen?
- Werden dem Patienten Substanzen zugeführt oder entzogen?
- Bearbeitet das Medizinprodukt biologische Substanzen zur anschließenden Wiederverwendung, Transfusion oder Transplantation?

Nachdem die wesentlichen Merkmale, die Einfluss auf die Sicherheit haben könnten, und der Gebrauch des Medizinproduktes ausgearbeitet wurden, müssen die Gefährdungen festgestellt werden (Feststellung von Gefährdungen). Dazu stellt die Norm verschiedene Techniken zur Identifikation vor.

Risiken Identifizieren

Anhang G, der DIN EN ISO 14971 stellt 5 verschiedene Techniken zur Identifikation von Gefährdungen vor:

1. Einleitende Gefährdungsanalyse (Preliminary Hazard Analysis, PHA)
2. Fehlerbaumanalyse (Fault Tree Analysis, FTA)
3. Fehler-Möglichkeits- und Einflussanalyse (Failure Mode and Effect Analysis, FMEA)
4. Studie über Gefährdung und Beherrschbarkeit (Hazard and Operability Study, HAZOP)
5. Gefahrenanalyse an kritischen Kontrollpunkten (Hazard Analysis on Critical Control Points, HACCP)

Die einleitende Gefährdungsanalyse liegt meist in tabellarischer Form vor. Inhalt der Tabelle ist die Wahrscheinlichkeit des Auftretens, qualitative Bewertung und dazugehörige Abhilfemaßnahmen.

Die FTA arbeitet im top-down-Modus. Die Darstellung erfolgt immer grafisch in Bäumen. Ausgehend vom top-event, werden in der nächstniedrigen Ebene mögliche Ursachen oder Fehlermöglichkeiten für das Auslösen des höher liegenden Events festgestellt. Kombinationen können mit logischen Operatoren (UND/ODER) auf den Ebenen beschrieben werden. Laut Norm können mit dieser Form Kombinationen, die am wahrscheinlichsten das top-event auslösen, aufgedeckt werden.

Die FMEA untersucht die Folgen eines einzelnen Versagens und untersucht diese. Die Technik stellt allgemein die Frage: *Was geschieht, falls...?* Im Gegensatz zur FTA arbeitet die FMEA im bottom-up-Modus. Es wird immer jeweils eine Komponente zu einem Zeitpunkt untersucht. Laut Norm eignet sich diese Technik im Umgang mit menschlichem Versagen (Anwendungsfehlern).

Die HAZOP behandelt hauptsächlich Unfälle, die durch Abweichungen von den Absichten der Gestaltung oder des Betriebs entstehen. Die HAZOP kann auf den Betrieb, bzw. die Funktion angewandt werden (Therapie, als *Absicht der Konstruktion*) oder bei der Herstellung auf Prozesse (Sterilisation). Ziele der Technik sind:

- Erarbeitung einer vollständigen Beschreibung des Medizinprodukts und seiner beabsichtigten Anwendung.
- Systematische Überprüfung jedes Teils des bestimmungsgemäßen Gebrauchs, um zu erkennen, wie Abweichungen von den normalen Betriebsbedingungen und der Absicht der Konstruktion auftreten können.

- Folgen solcher Abweichungen festzustellen und zu entscheiden, ob diese Folgen zu Gefährdungen oder zu Problemen der Betriebsfähigkeit führen können.

Die HACCP ist ein systematisches Vorgehen zur Feststellung, Bewertung und Kontrolle von Gefährdungen. Diese Analyse dient bei Medizinprodukten hauptsächlich zur Ermittlung von Gefährdungen, die ihren Ursprung in Verfahren (z.B. Herstellungsverfahren) haben.

Risiken/Restrisiken bewerten

Im ersten Schritt müssen Akzeptanzkriterien für das Risiko bestimmt werden. Die Richtlinie 93/42/EWG besagt, dass etwaige Risiken, verglichen mit dem Nutzen für den Patienten, vertretbar und mit einem Hohen Maß des Schutzes von Gesundheit und Sicherheit vereinbar sein müssen. Die DIN EN ISO 14971 legt hierzu nicht direkt vertretbare Risiken fest. Vielmehr schlägt die Norm ein bestimmtes Vorgehen zur Ermittlung von vertretbaren Risiken vor und überlässt letztendlich dem Hersteller und der aktuellen gesellschaftlichen Ethik die Entscheidung, welche Schwellwerte oder Verfahren am sinnvollsten einzusetzen sind. Die Norm schlägt folgende Vorgehensweisen zur Ermittlung von vertretbaren Risiken vor:

- Anwendung einschlägiger Normen, die Anforderungen festlegen, deren Durchsetzung die Erreichung der Vertretbarkeit für bestimmte Arten von Medizinprodukten oder bestimmten Risiken bedeutet
- Vergleich mit Risikograden, von bereits im Gebrauch befindlichen Medizinprodukten
- Auswertung der Daten aus klinischen Studien, besonders bei neuen Techniken oder neuen Zweckbestimmungen

Im Fall des HomeCare Medidisp würde nur das erste Verfahren (Anwendung einschlägiger Normen) in Betracht kommen. Durch Verwendung der Normen DIN EN 60601-1, DIN EN 6060-1-4, DIN EN 62366 und DIN EN ISO 14971 wird davon ausgegangen, dass dieser Punkt erfüllt wird. Die drei genannten Vorgehensweisen sollen außerdem unter Berücksichtigung des *Standes der Technik* angewandt werden. Laut Norm bedeutet Stand der Technik:

- Aktuelle Normen für gleiche oder ähnliche Produkte
- Aktuelle Praxis, wie sie bei anderen Produkten des gleichen oder ähnlichen Typs angewendet wird
- Ergebnisse anerkannter Forschung

Sind die Akzeptanzkriterien auf Basis der vorangegangenen Punkte ermittelt, schlägt die Norm die Anzeige dieser Kriterien in Form einer Matrix vor. Die Matrix

soll ermitteln, welche Kombinationen von Schadenswahrscheinlichkeit und Schadensgrad vertretbar oder unvertretbar sind. Eine minimale Matrix beträgt 3x3. Der unvertretbare Bereich ist strikt festgelegt. Dem Hersteller wird dafür die weitere Unterteilung des vertretbaren Bereiches freigestellt. Für die Ermittlung der Wahrscheinlichkeit können qualitative oder quantitative Werte genommen werden. Bei der Ermittlung von quantitativen Werten für die Wahrscheinlichkeit des Auftretens eines Fehlers sollen folgende Faktoren beachtet werden:

- Wie oft wird ein bestimmtes Medizinprodukt angewendet?
- Wie hoch ist die Lebensdauer des Medizinprodukts?
- Welches sind die Populationen der Anwender und Patienten?
- Wie groß ist die Anzahl der Anwender bzw. Patienten?
- Wie lange und unter welchen Umständen unterliegt der Anwender bzw. Patient der Exposition?

Anhand dieser Fragen würde beispielsweise eine geschätzte Anzahl von Anwendungen des HomeCare Medidisp in 5 Jahren bei ungefähr 9375 Anwendungen liegen. Dieser Wert gilt mehr als Richtwert für die Ermittlung der quantitativen Wahrscheinlichkeitsbereiche.

Die folgenden Tabellen zeigen eine halbquantitative Ermittlung der Vertretbarkeit, mittels einer 5x5-Matrix. Tabelle 1 und 2 zeigen vorgeschlagene Werte aus der Norm. Anders als in der Norm vorgeschlagen, wurde ein dreistufiges Modell für die Vertretbarkeit gewählt. Dieses Modell beinhaltet *vertretbar*, *ALARP* und *unvertretbar*. Das dreistufige Modell hat den Effekt, dass die Granularität der Einschätzung der entdeckten Risiken erhöht wird und somit eine bessere Einstufung möglich ist.

ALARP bedeutet *As Low As Reasonably Practicable*⁴⁵. Risiken im *ALARP*-Bereich sollen auf einen Punkt reduziert werden, welcher den höchsten Grad an Sicherheit garantiert, bei gleichzeitig praktikablen Ansätzen. Beispielsweise müssen bei der Produktentwicklung Maßnahmen für identifizierte Produktrisiken nur dann implementiert werden, wenn sie auch vernünftigerweise praktikabel sind (finanziell/technisch mit vertretbarem Aufwand realisierbar).

Für die Ermittlung der Bereiche wurde jedem qualitativen und quantitativen Wert eine Ziffer von 1-5 zugeteilt. Durch Multiplikation der beiden Ziffern wird ein Grenzwert ermittelt, worüber die Bereiche *vertretbar* (1-5), *ALARP* (6-10) und *unvertretbar* (11-25) zugeordnet werden.

⁴⁵ So niedrig, wie vernünftigerweise praktikabel

| Wahrscheinlichkeitsgrad | Wert | Ziffer |
|-------------------------|--------------------------------|--------|
| Häufig | $\geq 10^{-3}$ | 5 |
| Wahrscheinlich | $< 10^{-3}$ und $\geq 10^{-4}$ | 4 |
| Gelegentlich | $< 10^{-4}$ und $\geq 10^{-5}$ | 3 |
| Fernliegend | $< 10^{-5}$ und $\geq 10^{-6}$ | 2 |
| Unwahrscheinlich | $< 10^{-6}$ | 1 |

Tabelle 1: Quantitative Wahrscheinlichkeitsgrade

| Übliche Begriffe | Mögliche Beschreibung | Ziffer |
|------------------|---|--------|
| Katastrophal | Führt zum Tod des Patienten | 5 |
| Kritisch | Führt zu dauernder Behinderung oder einer lebensbedrohlichen Schädigung | 4 |
| Ernst | Führt zu einer Schädigung oder Behinderung, die ein sachkundiges medizinisches Eingreifen erfordert | 3 |
| Gering | Führt zu einer zeitweiligen Schädigung oder Behinderung, die kein sachkundiges medizinisches Eingreifen erfordert | 2 |
| Vernachlässigbar | Unannehmlichkeiten oder zeitweilige Beschwerden | 1 |

Tabelle 2: Qualitative Schweregrade

| Qualitative Schweregrade | | | | | | |
|---|------------------|------------------|----------------|-------------------|-------------------|-------------------|
| Halbquantitative Wahrscheinlichkeitsgrade | | Vernachlässigbar | Gering | Ernst | Kritisch | Katastrophal |
| | Häufig | Vertretbar (5) | ALARP (10) | Unvertretbar (15) | Unvertretbar (20) | Unvertretbar (25) |
| | Wahrscheinlich | Vertretbar (4) | ALARP (8) | Unvertretbar (12) | Unvertretbar (16) | Unvertretbar (20) |
| | Gelegentlich | Vertretbar (3) | ALARP (6) | ALARP (9) | Unvertretbar (12) | Unvertretbar (15) |
| | Fernliegend | Vertretbar (2) | Vertretbar (4) | ALARP (6) | ALARP (8) | ALARP (10) |
| | Unwahrscheinlich | Vertretbar (1) | Vertretbar (2) | Vertretbar (3) | Vertretbar (4) | Vertretbar (5) |

Tabelle 3: Ermittlung Risiko mit 5x5-Matrix

2.3.4 Aspekte aus der DIN EN 62366

Fokus dieser Norm liegt auf der Ermittlung und Verringerung von Risiken, die durch den Gebrauch des Medizinproduktes entstehen können und der gleichzeitigen Verbesserung der Gebrauchstauglichkeit. Die DIN EN 62366 definiert die Gebrauchstauglichkeit folgendermaßen:

„Eigenschaft der Benutzer-Produkt-Schnittstelle, die die Effektivität, Effizienz sowie die Lernförderlichkeit und Zufriedenstellung des Benutzers umfasst.“

Eine andere Definition beschreibt C. Backhaus⁴⁶. Er definiert die Gebrauchstauglichkeit aus dem Zusammenspiel von Bedienbarkeit und Funktionalität. Sind Bedienbarkeit und Funktionalität im Gleichklang, ist die höchste Gebrauchstauglichkeit erreicht. Beispielsweise würde eine zu hohe Funktionalität die Gebrauchstauglichkeit senken, da eine hohe Bedienbarkeit vieler Funktionen kaum gegeben ist.

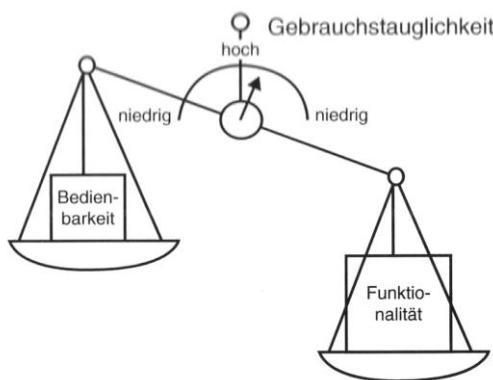


Abbildung 12: Definition Gebrauchstauglichkeit C. Backhaus [BACK2009], S. 17

Die Faktoren Bedienbarkeit und Funktionalität müssen laut C. Backhaus ausgeglichen sein, damit ein größtmögliches Maß an Gebrauchstauglichkeit erreicht wird. Ein gebrauchstauglichkeitsorientierter Entwicklungsprozess soll dieses Gleichgewicht sicherstellen.

Der vorgeschlagene Prozess aus der DIN EN 62366 soll die Sicherheit von Patient, Benutzer und anderen Personen *mit* Blick auf die Gebrauchstauglichkeit garantieren. Aus diesem Grund ist das Risikomanagement nach DIN EN ISO 14971 und der gebrauchstauglichkeitsorientierte Entwicklungsprozess aus der DIN EN 62366 eng miteinander verzahnt (Abbildung 13). Die gestrichelten Linien A, B, C, D und E stellen den Informationsfluss zwischen den Prozessen dar. A, B und C bilden den von der DIN EN 62366 geforderten Informationsfluss ab, D und E den formalen

⁴⁶ C. Backhaus. „Usability Engineering in der Medizintechnik: Grundlagen – Methoden – Beispiele“. [BACK2009], S. 16f

Informationsfluss mit der DIN EN ISO 14971. Der gebrauchstauglichkeitsorientierte Prozess teilt sich in folgende Punkte auf:

1. Spezifikation der Anwendung
2. Häufig benutzte Funktionen
3. Ermittlung von Gebrauchstauglichkeit mit zusammenhängenden Gefährdungen und Gefährdungssituationen
4. Hauptbedienfunktionen
5. Spezifikation der Gebrauchstauglichkeit
6. Validierungsplan für Gebrauchstauglichkeit
7. Gestaltung und technische Umsetzung der Benutzer-Produkt-Schnittstelle
8. Verifizierung der Gebrauchstauglichkeit
9. Validierung der Gebrauchstauglichkeit

Die auf Gebrauchstauglichkeit basierenden ermittelten Merkmale und Gefährdungen aus den Analysen der Punkte 2 und 3 fließen zusätzlich in die Risiko-Analyse mit ein (B). Die aus der Risiko-Analyse ermittelten Gefährdungen mit Bezug auf die Gebrauchstauglichkeit des Medizinproduktes beeinflussen den gebrauchstauglichkeitsorientierten Entwicklungsprozess (C). Zusätzlich fließen Erkenntnisse aus dem gebrauchstauglichkeitsorientierten Entwicklungsprozess mit in die Risikobewertung ein (D, E).

Ähnlich wie beim Risikomanagement muss der gebrauchstauglichkeitsorientierte Entwicklungsprozess in iterativen Phasen, meist zum Ende einer Design/Entwicklungsphase, durchgeführt werden. Dabei wird der Fokus auf die schrittweise Verbesserung der Benutzer-Produkt-Schnittstelle gelegt. Mit der Benutzer-Produkt-Schnittstelle ist nicht ausschließlich das GUI einer Software gemeint. Laut DIN EN 62366 ist die Benutzer-Produkt-Schnittstelle ein

„Mittel, über das der Benutzer mit dem Medizinprodukt interagiert“.

Das beinhaltet somit auch die Ergonomie der Hardwareoberfläche. Beispielsweise die Anordnung von Bedienelementen und Anschlüssen.

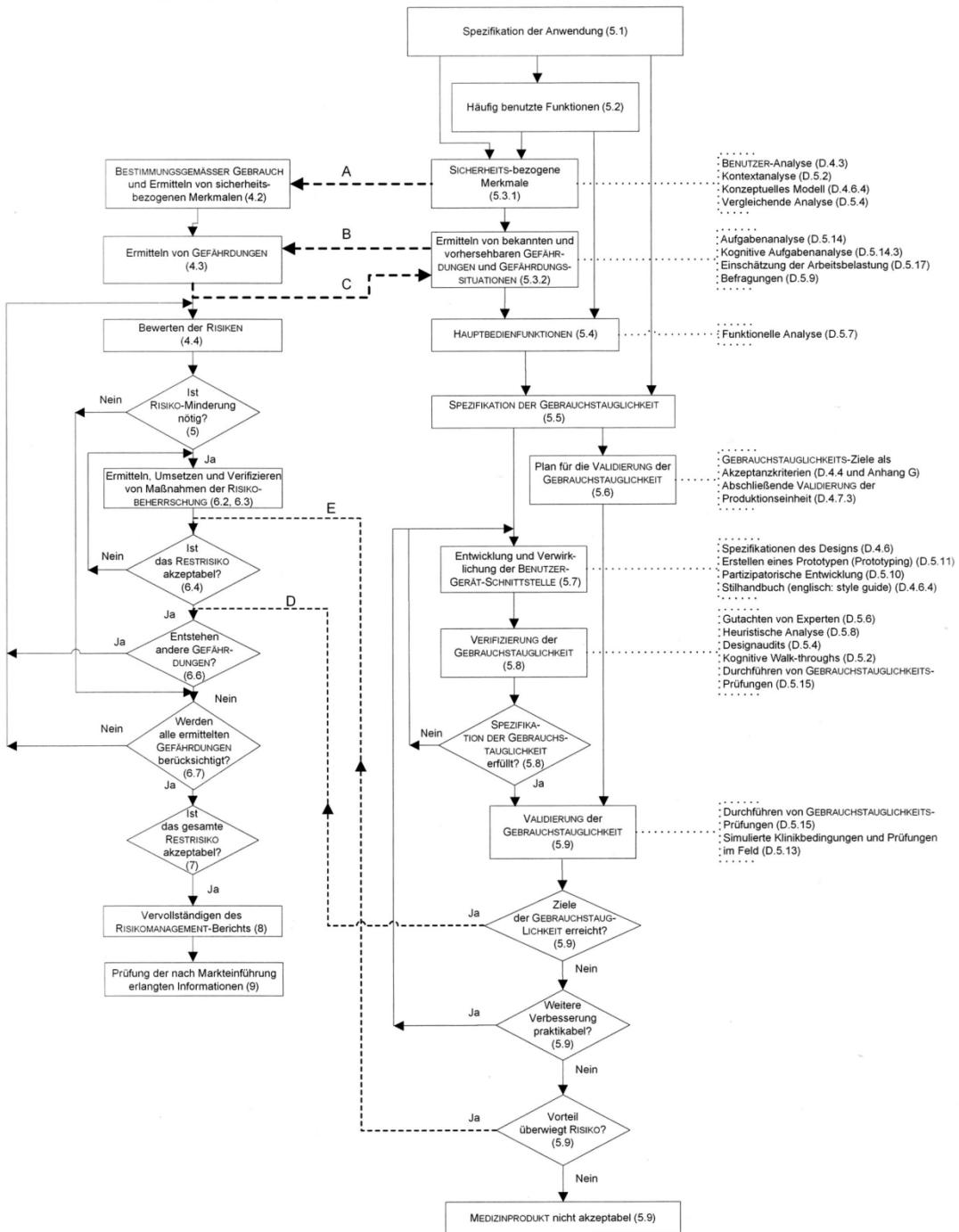


Abbildung 13: Verzahnung zwischen DIN EN 62366 und DIN EN ISO 14971⁴⁷

⁴⁷ Nach DIN EN 62366, S. 26

Benutzer-Taxonomie

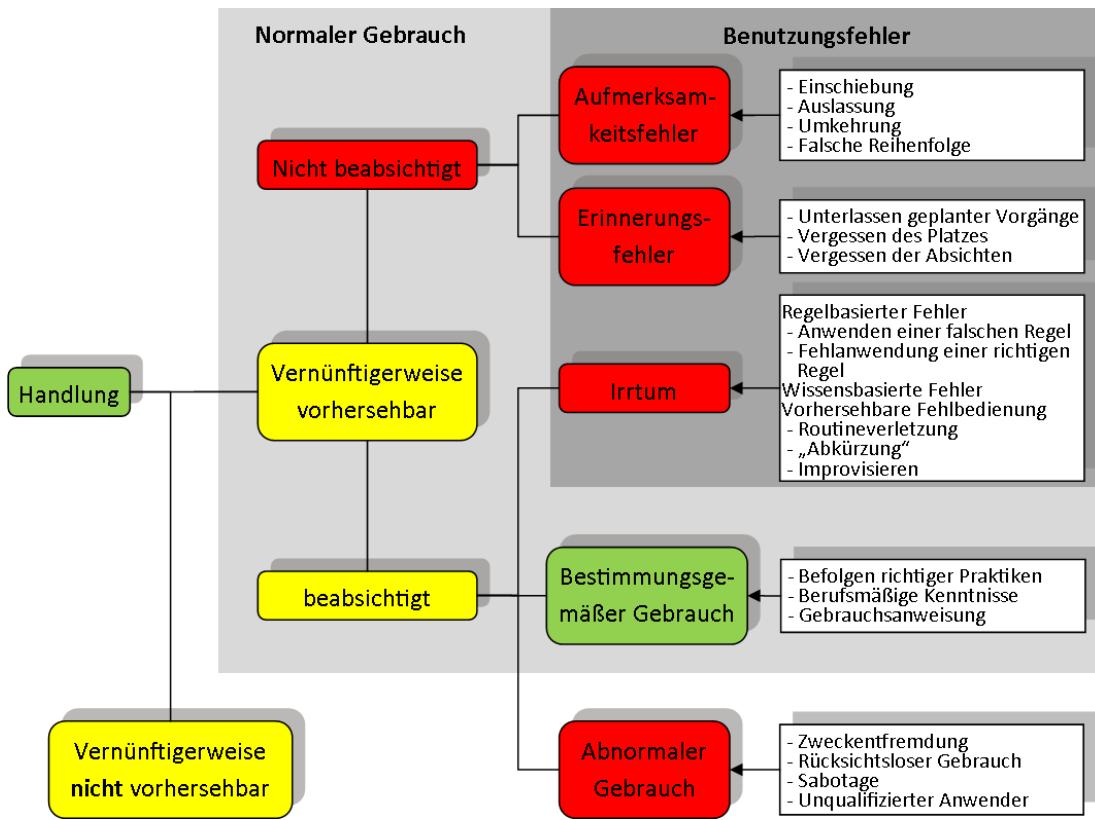


Abbildung 14: Benutzer Taxonomie nach DIN EN 62366, S. 33

Zum besseren Verständnis der Benutzerhandlungen schlüsselt die DIN EN 62366 sämtliche Handlungen und unterlassene Handlungen der Benutzer auf. Jede Handlung ist in eine Kategorie einteilbar. In einer ersten Trennung der Benutzerhandlung (grün, links mittig) unterteilt die Norm in vernünftigerweise *nicht* vorhersehbare und vorhersehbare Handlungen. Die vernünftigerweise vorhersehbaren Handlungen unterteilen sich weiter in beabsichtigt und nicht beabsichtigt.

Laut DIN EN 62366 muss der Hersteller Risiken aufgrund von Unterlassungen⁴⁸, die außerhalb der praktikablen Möglichkeiten des Herstellers zur Risikobeherrschung liegen (Abnormaler Gebrauch) *und* vernünftigerweise nicht vorhersehbaren Handlungen nicht weiter verringern⁴⁹. Beabsichtigte Handlungen, die beispielsweise gemäß der Gebrauchsanweisung sind, fallen in den bestimmungsgemäßen Gebrauch und bilden den gewünschten Fall. Irrtum, Erinnerungsfehler und Aufmerksamkeitsfehler fallen in die Kategorie der Benutzungsfehler.

⁴⁸ Beispielsweise nutzt ein Assistent in einer kritischen den Defibrillator falsch oder gar nicht (Schocksituation), da er mit der momentanen Situation überfordert war.

⁴⁹ Anhang B, S.33

Hauptziel der DIN EN 62366 ist es, Risiken sämtlicher Handlungen der Benutzer rund um Benutzungsfehler – die aufgrund der Gebrauchstauglichkeit zurückzuführen sind – möglichst zu minimieren und den Benutzer zu einer Handlung in den bestimmungsgemäßen Gebrauch zu führen.

2.3.5 Übernommene Prinzipien aus den Normen

Aus der DIN EN 60601-1 können für die Software nur wenige Punkte übernommen werden. Hauptaugenmerk liegt dabei auf der Suche nach einer geeigneten Hardwareplattform für die Steuersoftware. Diese muss die angesprochene 0,5mA-Grenze einhalten. Für die Kommunikation zwischen der Hardwarekomponente und der Softwarekomponente muss außerdem eine richtlinienkonforme Schnittstelle gefunden werden.

Für die Entwicklung des Systems wird das vorgeschlagene Entwicklungs-Lebenszyklus-Modell der DIN EN 60601-1-4 gewählt. Dazu wird das System möglichst in Subsysteme zerlegt. Diese Arbeit fokussiert den Entwicklungsprozess auf Seiten der Steuersoftware (Rechtes PESS Subsystem) und klammert die Entwicklung der Hardwarekomponente vorerst aus. Durch die DIN EN 62304 hat sich herauskristallisiert, dass die Bereitstellung der geeigneten Plattform für die Steuersoftware ein sehr kritischer Punkt ist. Durch Verwendung eines Betriebssystems, welches laut DIN EN 62304 als SOAP gilt, steigt der Aufwand beim Risikomanagement enorm.

Das Risikomanagement nach DIN EN ISO 14971 bildet einen zentralen Punkt in den genannten Normen. Die DIN EN 62366 verknüpft ihren gebrauchstauglichkeits-orientierten Entwicklungsprozess beispielsweise direkt mit dem Risikomanagement nach DIN EN ISO 14971⁵⁰. In der vorgeschlagenen Entwicklungsmethodik nach DIN EN 60601-1-4 und DIN EN 62304, sind Vorschläge gegeben, wann Punkte aus dem Risikomanagement behandelt werden sollten. Aufgrund des Umfangs des Risikomanagements werden in dieser Arbeit nur Risiko-Analyse, Bewertung und Maßnahmen näher behandelt. Für die Einstufung vertretbarer Risiken werden die ausgewählten Normen und der momentane Stand der Technik als Grundlage dienen. Die Risikobeurteilung wird mittels der beschriebenen 5x5-Matrix, inklusive quantitativer und qualitativer Merkmale, durchgeführt.

Der gebrauchstauglichkeitsorientierte Entwicklungsprozess aus der Norm DIN EN 62366 wird, soweit wie möglich, komplett angewandt.

⁵⁰ Siehe Abbildung 13, S. 33

3 Richtlinienkonformes Design

Grundsätzlich werden bei einer richtlinienkonformen Entwicklung die nötigen harmonisierten Normen als Basis genommen. Die DIN EN 60601-1-4 sowie die DIN EN 62304 schlagen jeweils ein Entwicklungsmodell vor. Diese Modelle ähneln sich sehr stark und beinhalten Prinzipien der modernen Softwareentwicklung, wie zum Beispiel iterative Vorgehensweisen. Wie bereits erwähnt, geben die Normen relativ viel Freiraum bei der Wahl der gewünschten Entwicklungsmodelle. Eine direkte Übernahme der Modelle aus den Normen ist nicht zwingend notwendig. Das Entwicklungsmodell des Herstellers muss laut DIN EN 60601-1-4 aber folgende Eigenschaften aufweisen⁵¹:

- Es muss ein Entwicklungs-Lebenszyklus für das Design des PEMS definiert werden.
- Der Entwicklungs-Lebenszyklus muss in Phasen und Aufgaben unterteilt werden, mit jeweils klar definierten Eingängen, Ausgängen und Tätigkeiten.
- Der Entwicklungs-Lebenszyklus muss integrale Prozesse für das Risiko-Management enthalten.
- Der Entwicklungs-Lebenszyklus muss Anforderungen für die Dokumentation enthalten.
- Das Risikomanagement muss sich über den gesamten Entwicklungs-Lebenszyklus - soweit anwendbar - erstrecken.
- Soweit zutreffend, muss ein definiertes System für die Problemlösung innerhalb und zwischen allen Phasen und Aufgaben des Entwicklungs-Lebenszyklus als ein Teil der Risikomanagement-Dokumentation entwickelt und aufrechterhalten werden. Abhängig vom Problem, kann das System folgende Eigenschaften besitzen:
 - o als ein Teil des Entwicklungs-Lebenszyklus definiert sein;
 - o den Bericht von potentiellen oder vorhandenen Sicherheits- und/oder Leistungsfähigkeitsproblemen erlauben;
 - o eine Beurteilung für jedes Problem bezüglich der damit verbundenen Risiken erhalten;
 - o die Kriterien festlegen (Sicherheit und/oder Leistungsfähigkeit), die für den Abbruch erfüllt werden müssen;
 - o die Handlungen festlegen, die zur Lösung jedes Problems führen;
 - o für jede Tätigkeit die Verfahren für die Validierung festlegen;
 - o die Schritte festlegen, die zur Verifizierung einer dauerhaften Einhaltung dieser Anforderungen unternommen werden.

⁵¹ Punkt 52.203, S. 10

3.1 Entwicklungs-Lebenszyklus-Modell nach DIN EN 60601-1-4

Da unser System aus Hard- und Softwareparts besteht, wurde für die Umsetzung die Entwicklungsmethodik aus der DIN EN 60601-1-4 größtenteils⁵² übernommen. Folgende Schritte sind nach dem Modell durchzuführen:

- Anforderungen an das PEMS (Phase 1)
- Funktionsspezifikation des PEMS, einschließlich der Architektur und der Schnittstellen (Phase 2)
- Funktionsspezifikation der Subsysteme (z.B. PESS), einschließlich der Architektur und der Schnittstellen (Phase 3)
- Komponentendesign (Phase 4)
- Komponentenentwicklung (Phase 4)
- Modulprüfung (Phase 5)
- Komponentenprüfung (Phase 6)
- Funktionsprüfung (Phase 6)
- Integration und System-Verifizierungsprüfung (Phase 7)
- System-Validierungs-Prüfung (Phase 8)

Die Entwicklungsmethodik muss weiterhin an die gegebenen Anforderungen durch den gebrauchstauglichkeitsorientierten Entwicklungsprozess (DIN EN 62366) bzw. die Risikomanagementnorm DIN EN ISO 14971 angepasst werden. Abbildung 15 zeigt diese Anpassungen. Die Maßnahmen der Entwicklungsmethodik nach DIN EN 60601-1-4 sind schwarz gekennzeichnet. Maßnahmen des gebrauchstauglichkeitsorientierten Entwicklungsprozesses (DIN EN 62366) sind grün und Maßnahmen des Risikomanagements (DIN EN ISO 14971) rot gekennzeichnet.

Die DIN EN 60601-1-4 schlägt außerdem für die Softwareentwicklung eines Produktes ein selbst gewähltes Software-Entwicklungsverfahren in den Phasen 3-6 vor. Das Software-Entwicklungsverfahren muss ebenfalls die Anforderungen aus den Normen erfüllen.

⁵² Siehe Abbildung 9, S. 22

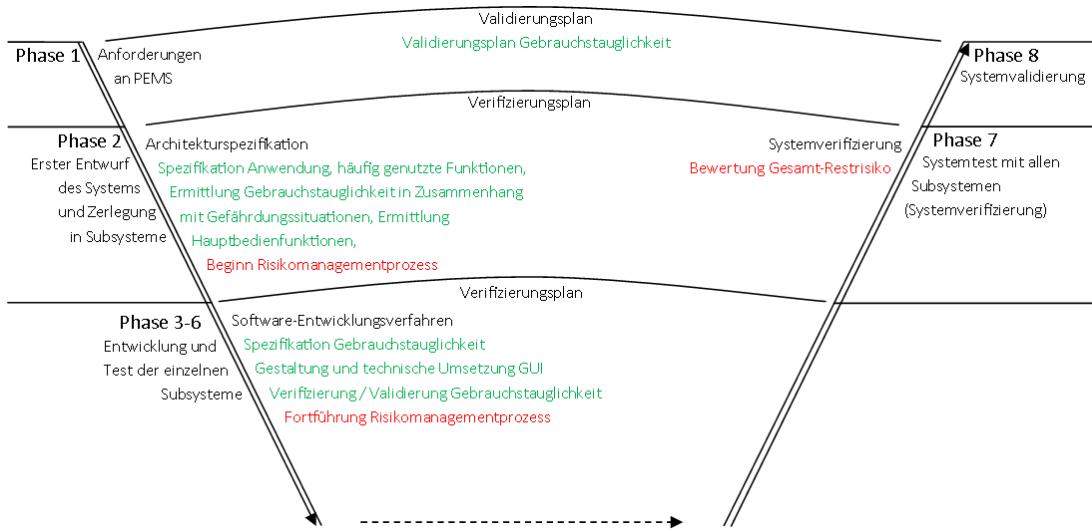


Abbildung 15: Modifiziertes E-L-Modell nach DIN EN 60601-1-4 (selbst erstellt)⁵³

Wesentliche Punkte für den gebrauchstauglichkeitsorientierten Entwicklungsprozess sind die Verifizierung und die Validierung. Laut DIN EN 62366 gelten folgende Anforderungen an den Verifizierungs- und Validierungsprozess (Evaluation):

- Der Hersteller muss während der Verifizierung des Gesamtdesigns des Medizinprodukts die technische Umsetzung der Benutzer-Produkt-Schnittstelle zusätzlich gegen die Anforderungen der Spezifikation der Gebrauchstauglichkeit verifizieren.
- Es wird verlangt, dass aus dem gebrauchstauglichkeitsorientierten Entwicklungsprozess resultierende Änderungen am Design auf mögliche Quellen neuer Gefährdungen hin untersucht werden.
- Begleitpapiere sind Teil der Benutzer-Produkt-Schnittstelle und somit auch Gegenstand der Validierung.
- Bei der Validierung sollen Personen eingesetzt werden, die nicht maßgeblich an der Entwicklung der Benutzer-Produkt-Schnittstelle beteiligt waren.
- Der Hersteller muss das von Problemen mit der Gebrauchstauglichkeit herrührende Risiko bewerten.
- Ergebnisse der Validierung der Gebrauchstauglichkeit sollten bei der Bewertung des Restrisikos berücksichtigt werden.

⁵³ Zusätzliche Anforderungen aus der DIN EN 62366 und DIN EN ISO 14971 (eigene Ausarbeitung)

Für die Umsetzung kann der Hersteller eigene Methoden oder vorgeschlagene Methoden aus der Norm verwenden. Folgende Methoden werden vorgeschlagen:

- Cognitive Walkthrough (Verifizierung)
- Designaudits (Verifizierung)
- Gutachten von Experten (Verifizierung)
- Heuristische Analyse (Verifizierung)
- Simulierte klinische Umgebungen und Feldprüfungen (Validierung)
- Gebrauchstauglichkeitsprüfung (Verifizierung und Validierung)

Eine ebenfalls normenkonforme Vorgehensweise zur Ermittlung der Gebrauchstauglichkeit stellt C. Backhaus⁵⁴ vor. Diese Vorgehensweise ist in 4 Schritte unterteilt: *Prozessanalyse*, *Prozessunterstützung*, *Usability-Evaluation* und *Gebrauchstauglichkeit* (Abbildung 16).

Auffällig sind in der Abbildung die Prozessmodule (PM). Prozessmodule werden anhand des zu unterstützenden Prozesses ermittelt. Jedes Prozessmodul muss einen Prozess in der Prozesskette darstellen. Die Module können später im System einzelne Funktionen oder sogar ganze Komponenten des Systems darstellen.

Die Ergebnisse aus der *Prozessanalyse* und *Usability-Evaluation* werden zusammengeführt und münden in einer umfassenden Bewertung der Gebrauchstauglichkeit. Die Bewertung kann in qualitativer oder quantitativer Form erfolgen⁵⁵. C. Backhaus definiert diese Bewertung⁵⁶ folgendermaßen:

„Ziel dabei ist es, die Auswirkungen der ermittelten Defizite auf den Gesamtprozess zu untersuchen und dadurch Aussagen zur Gebrauchstauglichkeit der untersuchten Medizintechnik zu machen.“

In dem bereits beschriebenen Vorgehensmodell der DIN EN 60601-1-4 könnten die Schritte *Prozessanalyse* und *Prozessunterstützung* den Phasen 1 und 2, die *Usability-Evaluation* und die *Gebrauchstauglichkeit* den Phasen 3-8 zugeordnet werden. Die Anwendung der 4 Schritte zur Evaluation ist nicht zwingend notwendig. Bei einem Produktbenchmarking können beispielsweise alle 4 Schritte verwendet werden, wohingegen bei der Produktentwicklung nur *Prozessanalyse* und *Usability-Evaluation* relevant sein können.

⁵⁴ C. Backhaus. [\[BACK2009\]](#), S. 65ff

⁵⁵ Siehe Abbildung 16: Punkt *quantitative Synthese?* Kategorie *Gebrauchstauglichkeit*

⁵⁶ C. Backhaus. [\[BACK2009\]](#), S. 73

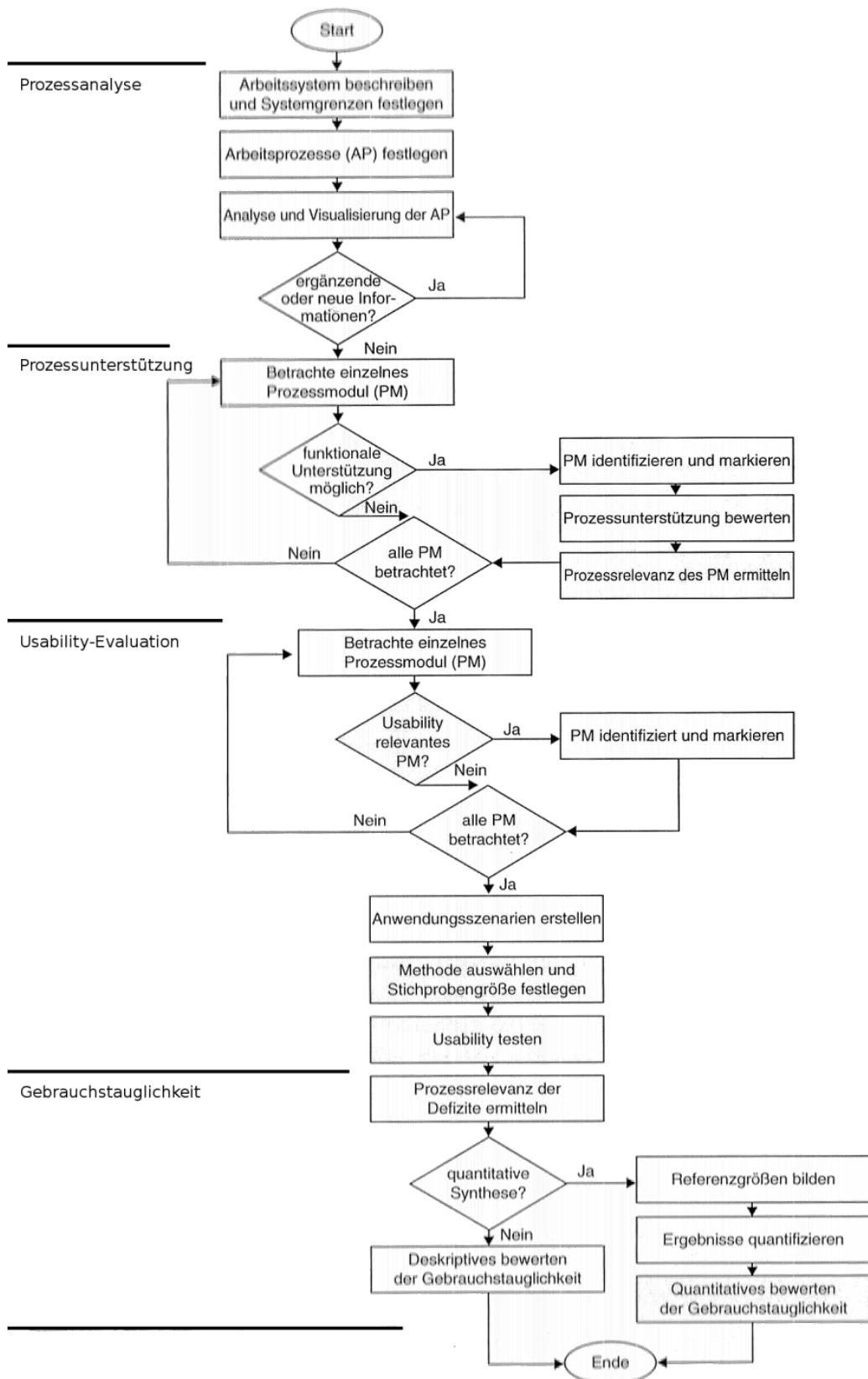


Abbildung 16: Bewertung der Gebrauchstauglichkeit von Medizinprodukten⁵⁷

⁵⁷ C. Backhaus. [BACK2009], S. 78

In einem Beispiel von C. Backhaus wird die Produktentwicklung⁵⁸ einer Infusionstherapiebaureihe für Intensivmedizin und stationäre/ambulante Pflege vorgestellt.

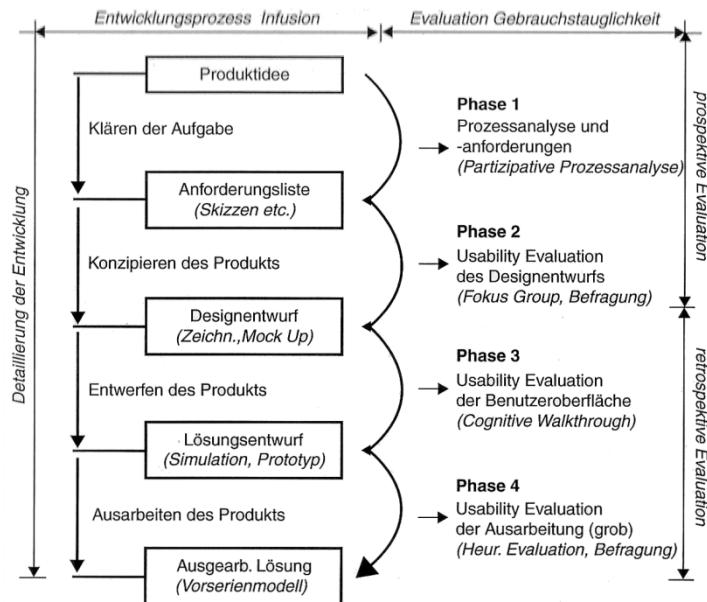


Abbildung 17: Beispiel Evaluation einer neuen Infusionstherapiebaureihe⁵⁹

Es wird ein iteratives entwicklungsbegleitendes Vorgehen zur kontinuierlichen Verbesserung der Gebrauchstauglichkeit angewandt (Abbildung 17). Zu Beginn der Entwicklung wird eine Prozessanalyse durchgeführt. Laut C. Backhaus [BACK2009] definiert sich eine Prozessanalyse folgendermaßen:

„Die Aufgabe der Prozessanalyse ist es, Arbeitsabläufe zu dokumentieren und durch ein gezieltes Hinterfragen, die Möglichkeit einer technischen Unterstützung durch den Einsatz von Medizintechnik zu klären.“

Solche Prozessanalysen werden häufig zur Analyse von Arbeitsprozessen im klinischen Alltag verwendet und ermitteln, an welcher Stelle das Produkt den Prozess unterstützen kann. Während der Entwicklung wird iterativ eine Usability-Evaluation, angepasst an die jeweiligen Phasen, durchgeführt. Für diese Arbeit wird das gleiche Vorgehen für die Evaluation der Gebrauchstauglichkeit angestrebt.

Somit ist in Phase 1-2 die Prozessanalyse und in Phase 3-6 die Usability-Evaluation angesiedelt. Die Prozessanalyse unterstützt die Spezifikation der Gebrauchstauglichkeit und hilft bei der Ermittlung von Gefährdungen im Zusammenhang mit der Gebrauchstauglichkeit. Die Usability-Evaluation erfüllt den Punkt der Verifizierung/Validierung der Gebrauchstauglichkeit aus der Norm DIN EN 62366.

⁵⁸ C. Backhaus. [BACK2009], S. 110ff

⁵⁹ C. Backhaus. [BACK2009], S.111

3.2 Phase 1 - Anforderungsanalyse

Das Entwicklungs-Lebenszyklus-Modell der DIN EN 60601-1-4 legt in der Phase 1 viel Wert auf die Anforderungsanalyse. Neben der Anforderungsspezifikation für das System, werden in dieser Phase sämtliche Dokumente und Pläne erstellt. Zu den Punkten der Phase 1 gehören somit:

- Erstellung Risikomanagement-Plan
- Erstellung Verifizierungs-Plan
- Erstellung Validierungs-Plan
- Wahl des Entwicklungs-Lebenszyklus
- Erste Gefährdungs- und Ursachenermittlung
- Anforderungen zur Risiko-Beherrschung
- Erstellung Risiko-Abschätzung
- Erstellung PEMS-Anforderungsspezifikation

Aus der DIN EN 62366 geht hervor, dass mit dem gebrauchstauglichkeits-orientierten Entwicklungsprozess begonnen wird. Somit ergeben sich weitere Punkte:

- Erstellung Gebrauchstauglichkeitsakte
- Erstellung Validierungsplan für Gebrauchstauglichkeit

Wie bereits erwähnt, liegt in der ersten Phase das Hauptaugenmerk auf der Anforderungsspezifikation. Laut DIN EN 60601-1-4 muss die Anforderungsspezifikation risikobezogene Funktionen detailliert aufführen⁶⁰:

„Das beinhaltet Funktionen zur Beherrschung von Risiken, welche durch:

- a) Umgebungsbedingungen;
- b) sonstige Ursachen innerhalb des PEMS;
- c) mögliche Fehlfunktionen

hervorgerufen werden.“

Außerdem sagt die DIN EN 60601-1-4⁶¹:

„Die Anforderungsspezifikation muss die Informationen enthalten, die notwendig sind, um sicherzustellen, dass die Risikobeherrschungsmaßnahmen die identifizierten Risiken ausreichend verringern.“

⁶⁰ Punkt 52.306.2, S. 12

⁶¹ Punkt 52.306.3

Laut C. Harböck⁶² werden von der Zertifizierungsstelle folgende Kriterien geprüft:

- Ersteller, Ausgabedatum, (Version) und Freigabe
- Allgemeine Beschreibung/Programmvorhaben
- Beschreibung DV-Systemumgebung
- Ein-/Beschränkungen der HW/SW
- Betriebssysteme, Programmiersprachen, Entwicklungsumgebung
- Vernetzung
- Liste der gewünschten/ungewünschten Funktionen und Eigenschaften
- Benutzerschnittstellen Sprache, (Bildschirm-) Ein- und Ausgabemasken, Kommandos, Tasturbefehle, Druckerausgaben, E/A-Geräte
- Sonstige Schnittstellen (zu anderen Systemen)
- Validierung von Fremdsoftware
- Gefährdungsmindernde Anforderungen
- Fehlermeldungen (Fließen in Gebrauchsanweisung ein)
- Qualifikationsvorgaben für den Anwender
- Datenmodelle, Datenschutz, Datensicherheit
- Vorschriften (spezielle Normen oder Anforderungen)

Die Umsetzung dieser Kriterien soll den Inhalt der Anforderungsspezifikationen aus der DIN EN 60601-1-4 darstellen.

Einen Ansatz zur Identifikation von möglichen gebrauchstauglichkeitsorientierten Anforderungen zeigt C. Backhaus. Er stellt für die Anforderungsermittlung eine Prozessanalyse mit Hilfe der TOPICS-Methode⁶³ vor (Abbildung 18). Mit dieser Methode werden in Krankenhäusern normalerweise Prozesse und deren Workflows verbessert. Die TOPICS-Methode teilt sich in die Schritte *Partizipative Prozessvisualisierung*⁶⁴ und *Problemspezifischer Prozessvergleich*⁶⁵ auf.

⁶² C. Harböck. [\[HARB2001\]](#), S.62

⁶³ Together Optimizing Process in Clinical Systems

⁶⁴ C. Backhaus. [\[BACK2009\]](#), S. 44: Visualisierung, Verifizierung, Analyse und Bewertung der Arbeitsprozesse im Untersuchungsumfeld.

⁶⁵ C. Backhaus. [\[BACK2009\]](#), S. 44: Verbesserung defizitärer Prozessmodule, die in der partizipativen Prozessflussvisualisierung ermittelt wurden.

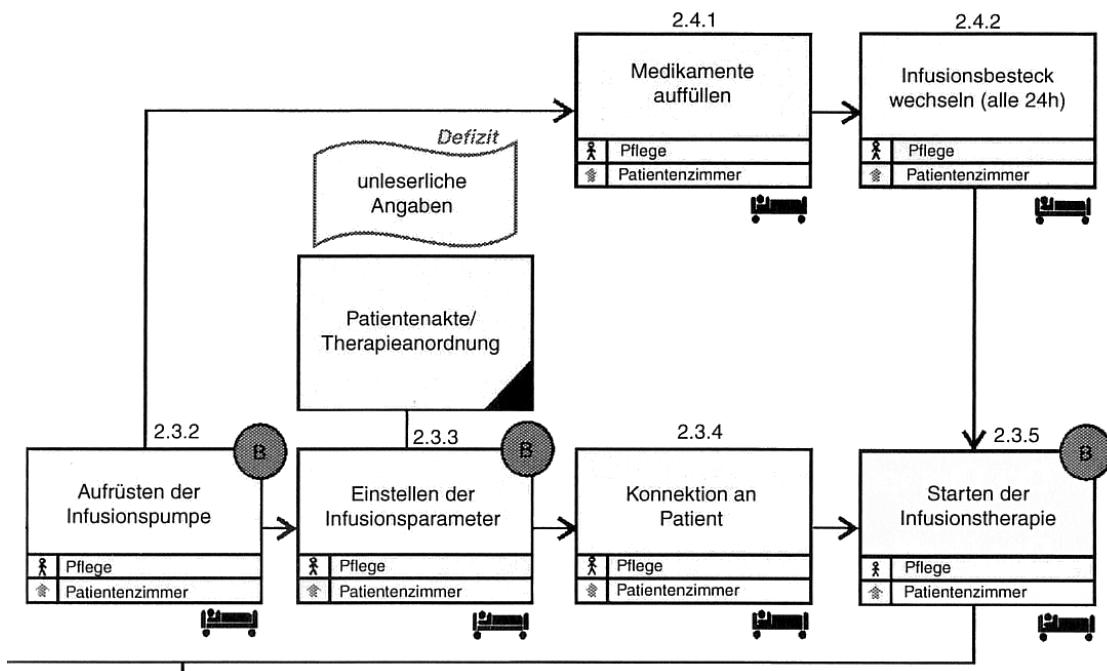


Abbildung 18: Beispiel TOPICS [BACK2009], S. 115

Anschließend werden die Prozessmodule, bei denen das Medizingerät den Prozess unterstützt, markiert. Die erkannten Module – in Abbildung 18 mit dem *B* gekennzeichnet – sind gleichzeitig nach C. Backhaus auch die Usability-relevanten Prozessmodule des Produktes. Über die erkannten Module können Prioritäten oder prozessgebundene Anforderungen für die Entwicklung ermittelt werden (Tabelle 4).

| Prozessrelevanz | Anzahl identifizierter Prozessmodule | |
|-----------------|--------------------------------------|--|
| | Prozessunterstützung möglich | Durch TIVA-Einheit ⁶⁶ bereits unterstützt |
| Hoch | 12 | 11 |
| Mittel | 4 | 1 |
| Niedrig | 4 | 0 |
| Gesamt | 20 | 12 |

Tabelle 4: Prozessunterstützung und Prozessrelevanz [BACK2009], S. 85

⁶⁶ TIVA-Einheit: Total-Intravenöse-Anästhesie-Einheit

3.3 Phase 2 - Entwurf des Systems

Nachdem Anforderungsspezifikationen und nötige Dokumente erstellt sind, fordert die DIN EN 60601-1-4 einen ersten Entwurf des Systems und die Zerlegung in Subsysteme (PESS). Laut DIN EN 60601-1-4, DIN EN 62366 und DIN EN ISO 14971 sind in Phase 2 folgende Schritte angesiedelt:

- Architektspezifikation PEMS
- Spezifikation der Anwendungen
- Ermittlung der häufig genutzten Funktionen
- Ermittlung von mit Gebrauchstauglichkeit zusammenhängenden Gefährdungen und Gefährdungssituationen
- Festlegung der Hauptbedienfunktionen
- Beginn Risikomanagementprozess

Architektspezifikation PEMS

Die DIN EN 60601-1-4 schlägt in Phase 2 die Erstellung der grundlegenden Architektspezifikation des Gesamtsystems (PEMS) vor. Die Architektspezifikation muss die Architektur des Systems und dessen Schnittstellen enthalten.

Die Architektur muss so weit vorangeschritten sein, dass alle Subsysteme des PEMS definiert sind. Des Weiteren verlangt die DIN EN 60601-1-4⁶⁷, dass die Architektspezifikation folgendes berücksichtigen muss:

- Die Architektur muss den Anforderungsspezifikationen genügen.
- Die Architektur-Spezifikation eines PEMS und seiner Subsysteme muss die Anforderungen zur Risikobeherrschung erfüllen.
- Zur Verringerung der Gefährdungswahrscheinlichkeit muss die Architektspezifikation Anforderungen für Folgendes enthalten:
 - Bauteile hoher Zuverlässigkeit
 - Funktionen zur Ausfallsicherheit
 - Redundanz
 - Diversität
 - Defensives Design
 - Begrenzung potentiell gefährlicher Effekte
- Zuordnung von Risikobeherrschungsmaßnahmen zu Subsystemen und Komponenten des PEMS
- Ausfallverhalten von Komponenten und dessen Effekten

⁶⁷ Punkte 52.207.1 bis 52.207.5, S.12

- Common-Cause-Ausfälle
- Systematische Ausfälle
- Testintervall, Testdauer und diagnostischer Abdeckungsgrad
- Wartbarkeit
- Schutz gegen absichtliche oder unbeabsichtigte menschliche Eingriffe

Spezifikation der Anwendungen

Der gebrauchstauglichkeitsorientierte Entwicklungsprozess der DIN EN 62366 beginnt mit der Spezifikation der Anwendungen. Die DIN EN 62366⁶⁸ definiert die Spezifikation der Anwendungen folgendermaßen:

„Der in dieser Norm beschriebene gebrauchstauglichkeitsorientierte Entwicklungsprozess beginnt mit dem Festlegen der grundlegenden Eigenschaften, die sich auf die Art der Benutzung des Medizinprodukts auswirken ... Die Merkmale der Anwendung bilden den grundlegenden Ausgangspunkt zur Bestimmung von bekannten und vorhersehbaren Gefährdungen und Gefährdungssituationen, die mit der Benutzer-Produkt-Schnittstelle zusammenhängen...“

Nach der Definition legt die DIN EN 62366 vier Punkte für die Spezifikation der Anwendung fest:

- Vorgesehene medizinische Indikation
- Vorgesehenes Benutzerprofil
- Vorgesehene Nutzungsbedingungen
- Physikalische Funktionsweise

Es wird darauf hingewiesen, dass die Spezifikation der Anwendung als Grundlage für die Spezifikation der Gebrauchstauglichkeit dient.

Ermittlung der häufig benutzen Funktionen

Die DIN EN 62366⁶⁹ weist nachdrücklich darauf hin, dass Mängel von häufig benutzten Funktionen wesentlich schwerwiegender sind als von weniger häufig genutzten Funktionen:

„Unangemessene Gebrauchstauglichkeit der häufig genutzten Funktionen kann die Sicherheit durch erhöhte Wahrscheinlichkeit von Benutzungsfehlern beeinträchtigen. Eine Ansammlung von selbst relativ unbedeutenden Mängeln der Benutzer-Produkt-Schnittstelle kann, insbesondere wenn häufig genutzte Funktionen

⁶⁸ Anhang A.1, Zu 5.1, S. 27 oder S. 13, Punkt 5.1

⁶⁹ Anhang A.1, Zu 5.2, S. 28

beteiligt sind, die Wahrscheinlichkeit von Benutzungsfehlern und deren Folgen erhöhen.“

Aus diesem Grund muss der Hersteller eine Analyse zur Ermittlung von häufig genutzten Funktionen durchführen und die ermittelten Hauptbedienfunktionen auflisten. Laut DIN EN 62366 gehören zu den häufig benutzten Funktionen auch diejenigen, die nur im Notfall verwendet werden. Denn, wenn ein Notfall eintritt, müssen diese Funktionen die höchste Sicherheit gewährleisten, auch in Bezug auf die Gebrauchstauglichkeit.

Ermittlung der mit Gebrauchstauglichkeit zusammenhängenden Gefährdungen und Gefährdungssituationen

Die DIN EN 62366 verweist auf die Risikoanalyse gemäß DIN EN ISO 14971. Laut DIN EN 62366⁷⁰ müssen in die Risikoanalyse im Rahmen der Ermittlung zwei zusätzliche Aspekte einfließen:

- 1) Ermittlung sicherheitsbezogener Merkmale mit Schwerpunkt Gebrauchstauglichkeit
- 2) Ermittlung bekannter oder vorhersehbarer Gefährdungen und Gefährdungssituationen

Grundgedanke der Risikoanalyse gemäß DIN EN ISO 14971 ist eigentlich die Ermittlung bekannter oder vorhersehbarer Gefährdungen und Gefährdungssituationen (Aspekt 2). In der DIN EN 62366 wird jedoch ausdrücklich darauf hingewiesen, dass bei der Ermittlung auch Gefährdungen für Patienten, Benutzer und andere Personen berücksichtigt werden müssen, die aufgrund der Benutzer-Produkt-Schnittstelle zustande kommen. Um das zu gewährleisten, schlägt die DIN EN 62366 folgende Punkte vor, auf die bei der Ermittlung geachtet werden sollte:

- Aufgabenbezogene Forderungen definieren (z.B. Das Medizinprodukt muss mit einer Hand transportierbar sein.)
- Analyse des Benutzungskontextes (z.B. Räumliche Rahmenbedingungen, soziale Rahmenbedingungen, usw.)
- Eventuell fehlerhaftes mentales Modell des Benutzers untersuchen (z.B. Der Benutzer glaubt aus der Ergonomie heraus, dass das Gerät so funktioniert, bedient es aber falsch.)
- Iterative Überprüfung der Benutzer-Produkt-Schnittstelle (Evaluierung)

⁷⁰ Punkte 5.3, 5.3.1 und 5.3.2

Festlegung der Hauptbedienfunktionen

Als nächsten Schritt muss der Hersteller die Hauptbedienfunktionen festlegen. Zu den Hauptbedienfunktionen können häufig benutzte Funktionen und sicherheitsbezogene Funktionen zählen. Die DIN EN 62366 schlägt als Hilfe für die Ermittlung eine Funktionsanalyse vor. Vorgeschlagene Methoden für eine Funktionsanalyse sind:

- Arbeitsablaufdiagramme
- FAST (Functional Analysis Systems Technique)
- Computersimulationen
- SAINT (Systems Analysis of Integrated Network of Tasks)

Beginn Risikomanagementprozess

Laut DIN EN ISO 14971 muss sich der Risikomanagementprozess über den gesamten Entwicklungs-Lebenszyklus des Medizinproduktes erstrecken. Bestandteile des Prozesses sind:

- Risiko-Analyse
- Risiko-Bewertung
- Maßnahmen zur Risikominimierung
- Ermittlung des Restrisikos
- ggf. Risiko-Nutzen-Analyse

Wie in Kapitel 2.3.3 zu lesen, baut der Risikomanagementprozess nach DIN EN ISO 14971 auf einer iterativen Vorgehensweise auf. Die erste Iteration in der Entwicklung sollte deshalb bereits an den Anforderungen und den ersten Architekturen durchgeführt werden, damit schwerwiegende Fehler und Gefährdungen bereits frühzeitig erkannt werden können.

3.4 Phase 3 – 6 - Software-Entwicklungsverfahren

Für die Phasen 3-4 kann laut DIN EN 60601-1-4 ein Software-Entwicklungsverfahren verwendet werden⁷¹. Neben den grundlegenden Anforderungen aus den Normen⁷² wird vom Software-Entwicklungsverfahren explizit eine iterativ inkrementelle Vorgehensweise verlangt. Das Risikomanagement muss in diesen Prozess mit eingegliedert werden. Verifizierungsmethoden für die Phasen 5-7 schlägt die DIN EN 60601-1-4 nicht direkt vor. Sinnvoll ist aber eine bereits integrierte Verifizierung im angewandten Software-Entwicklungsverfahren, sodass das gesamte Verfahren die Phasen 3-6 umfasst. Nachfolgend werden dazu zwei Software-Entwicklungsverfahren vorgestellt.

3.4.1 Das V-Modell XT

Laut C. Harböck⁷³ ist das V-Modell für die Entwicklung von Medizinprodukten am besten geeignet.

„Weiterhin wird in einem Vergleich von Wasserfall-Modell, Prototypen-Modell, Spiralmodell, Evolutions-Modell und des V-Modells, das V-Modell (Submodell SE) aufgrund seiner stark risikoorientierten Weise, als der beste Entwicklungs-Lebenszyklus für Medizinprodukte bezeichnet.“

Das Besondere am V-Modell ist, dass hier lediglich Aktivitäten und Ergebnisse definiert werden. Dabei ist keine strikte zeitliche Abfolge gefordert. Laut DBBI⁷⁴ ist das V-Modell XT folgendermaßen zu definieren:

„Das V-Modell ist ein Leitfaden zum Planen und Durchführen von Entwicklungsprojekten unter Berücksichtigung des gesamten System-Lebenszyklus. Dabei definiert das V-Modell die in einem Projekt zu erstellenden Ergebnisse und beschreibt die konkreten Vorgehensweisen, mittels derer diese Ergebnisse erarbeitet werden. Darüber hinaus legt das V-Modell die Verantwortlichkeiten der einzelnen Projektbeteiligten fest. Es regelt also wer, wann, was in einem Projekt zu tun hat.“

„Der Namenszusatz XT steht für eXtreme Tailoring. Dies soll die hohe Anpassungsfähigkeit des V-Modells an die unterschiedlichsten Anforderungsprofile der verschiedenen Projekte und Organisationen verdeutlichen.“

⁷¹ Siehe Kapitel 2.3.2, S. 19

⁷² Siehe Kapitel 2.3, S. 38f und Abbildung 15, S. 38

⁷³ C. Harböck. [\[HARB2001\]](#), S. 51

⁷⁴ Die Beauftragte der Bundesregierung für Informationstechnik (DBBI). „Häufig gestellte Fragen zum V-Modell XT“. [\[IQ13\]](#)

Das neueste V-Modell ist das V-Modell XT 1.3, welches am 01.02.2009 veröffentlicht wurde. Die Fähigkeit des Tailorings (Anpassbarkeit) war bereits im V-Modell 97 gegeben. Grundidee des V-Modells ist es, die hohe Komplexität des Gesamtsystems, durch die Zerlegung in kleine überschaubare Module (Teilsysteme), möglichst weit zu verringern. Mit zunehmender Entwicklungszeit wird das System somit in immer kleinere Teile (Module) zerlegt. Die weniger komplexen Module können dann in einem hohen Detailgrad design werden. Die Tests des Systems beginnen bei den Modulen und enden in einem Test des Gesamtsystems. Setzt man den Detailgrad mit der Entwicklungszeit in Relation, erhält man das typische V-Modell (Abbildung 19).

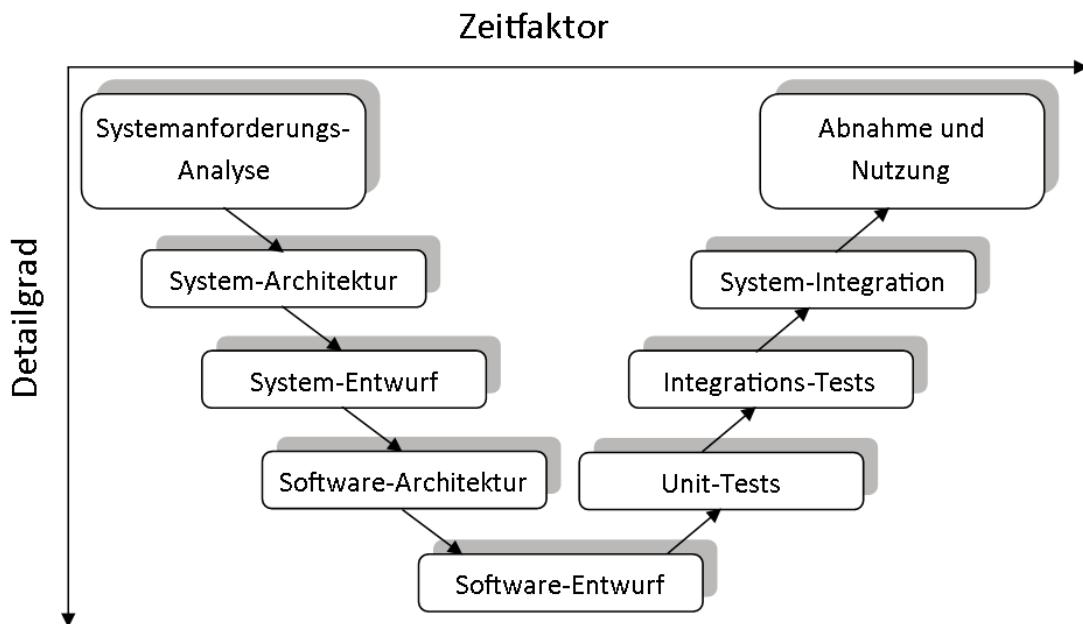


Abbildung 19: Typisches vorgehen nach V-Modell (selbst erstellt)

Im V-Modell werden bestimmte Pflichtdokumente definiert. Diese Dokumente werden als Produkt bezeichnet. Jedes Produkt kann beliebig oft erweitert und bearbeitet werden, bis das gewünschte Ergebnis erreicht ist (Abbildung 20).

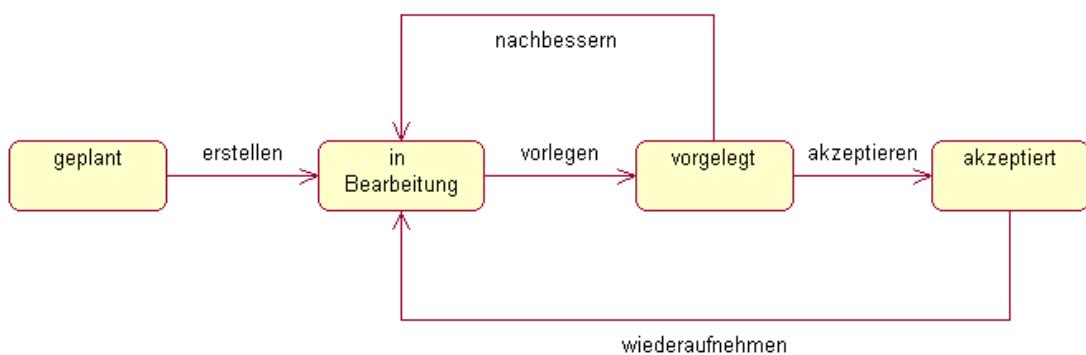


Abbildung 20: iterativ inkrementelles Vorgehen im V-Modell (Wikipedia)

Um die Dokumente (Produkte) korrekt zu erstellen und zu bearbeiten, werden im V-Modell verschiedene Aktivitäten definiert. Jede Aktivität beinhaltet genaue Informationen über:

- Das zugehörige Produkt, welches benötigt und/oder verändert wird (Produktfluss).
- Notwendige Arbeitsschritte, um das Produkt korrekt zu bearbeiten (Abwicklung).

Neben den Aktivitäten existieren noch sogenannte Tätigkeiten, welche zu Vorgehensbausteinen zusammengefasst werden. Ein Teil der Vorgehensbausteine ist beispielhaft in Tabelle 5 abgebildet. Im V-Modell existieren zwei elementare Sichten, zu denen die Vorgehensbausteine zugeordnet werden können. Die Auftraggeber- (AG) und Auftragnehmer-Sicht (AN). Je nach Szenario, können die Vorgehensbausteine im Projekt verwendet werden oder nicht (das bereits beschriebene Tailoring). Die als V-Modell Kern gekennzeichneten Vorgehensbausteine finden in jedem V-Modell Anwendung und bilden die Basis jedes Modells.

| Vorgehensbaustein | V-Modell Kern | Systementwicklung sprojekt (AG/AN) | Systementwicklung sprojekt (AG) | Systementwicklungsprojekt (AN) |
|----------------------------------|---------------|------------------------------------|---------------------------------|--------------------------------|
| Projektmanagement | X | X | X | X |
| Qualitätssicherung | X | X | X | X |
| Konfigurationsmanagement | X | X | X | X |
| Problem- und Änderungsmanagement | X | X | X | X |
| Systemerstellung | | X | | X |
| Anforderungserstellung | | X | X | |
| Benutzbarkeit und Ergonomie | | X | | X |
| Lieferung und Abnahme (AN) | | X | | X |
| Lieferung und Abnahme (AG) | | X | X | |
| Vertragsabschluss (AG) | | | X | |
| Vertragsabschluss (AN) | | | | X |

Tabelle 5: Vorgehensbausteine in Abhängigkeit des Projekttyps

3.4.2 Agile Softwareentwicklung

Die agile Softwareentwicklung genießt, durch das Interesse an eXtreme Programming (XP), seit Ende der 1990er Jahre eine stetig wachsende Popularität. Anfangs als leichtgewichtige Softwareentwicklung bezeichnet, wurde bei einem Treffen in Utah 2001 die Bezeichnung *agile* Softwareentwicklung ausgewählt. Bei diesem Treffen wurde gleichzeitig der Begriff der agilen Softwareentwicklung definiert und das sogenannte *Agile Manifest*⁷⁵ erstellt. Das Agile Manifest beinhaltet die Grundprinzipien der agilen Softwareentwicklung in Form von Werten:

„Wir erschließen bessere Wege, Software zu entwickeln, indem wir es selbst tun und anderen dabei helfen. Durch diese Tätigkeit haben wir diese Werte zu schätzen gelernt:“

- **Individuen und Interaktionen** mehr als Prozesse und Werkzeuge
- **Funktionierende Software** mehr als umfassende Dokumentation
- **Zusammenarbeit mit dem Kunden** mehr als Vertragsverhandlung
- **Reagieren auf Veränderung** mehr als das Befolgen eines Plans

Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden, schätzen wir die Werte auf der linken Seite höher ein.“

Aus diesen Grundprinzipien heraus haben sich verschiedene Methoden der agilen Softwareentwicklung herauskristallisiert. Dazu gehören laut it-agile⁷⁶:

- Scrum
- eXtreme Programming (XP)
- Kanban
- Feature Driven Development (FDD)
- Crystal
- The Eclipse Way

Im Kern beinhalten alle Methoden ein iteratives Vorgehen. Zielgedanke der iterativen Vorgehensweise ist es, dynamisch auf Änderungen des Systems reagieren zu können. Ein typisches Beispiel dafür ist, dass die erstellte Software an den Anforderungen des Kunden *vorbei* entwickelt wird. Oder, dass der Kunde nach der Anforderungsspezifikation zusätzliche Anforderungen wünscht. Diese Problematik ist bei der agilen Softwareentwicklung weniger gegeben. Grund dafür ist, dass der Gesamtdesignprozess meist in viele kleine Prozesse aufgeteilt wird, welche jeweils eine Iteration darstellen (Abbildung 21). Dabei zeigt sich, je kleiner die Iterationen gehalten werden, umso *agiler* ist die Entwicklung des Systems.

⁷⁵ K. Beck et. al. „Manifesto for Agile Software Development“. [\[IQ14\]](#)

⁷⁶ It-agile. „Agile Methoden“. [\[IQ15\]](#)

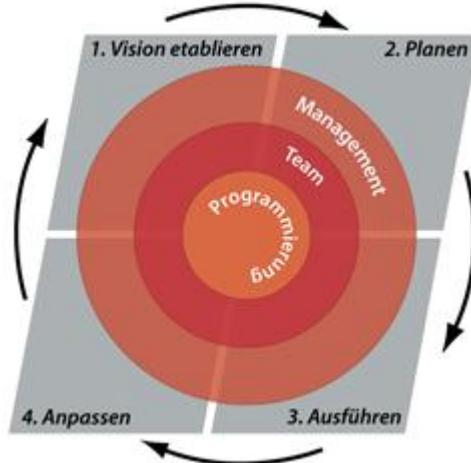


Abbildung 21: Iterativer Zyklus in der agilen Softwareentwicklung [IQ16]

Scrum

Das Scrum ist eines der berühmtesten agilen Methoden. Es besitzt feste Rollen und zeichnet sich durch seine einfache Struktur aus. Dadurch lässt sich die Methode schnell umsetzen. Im Mittelpunkt des Scrums steht das sich selbstorganisierende Entwicklerteam. Die Hauptrollen im Scrum übernehmen der ScrumMaster und der Product-Owner. Der Owner hat die Aufgabe, Anforderungen zu definieren, zu priorisieren und eventuell auch zu tauschen.

Jeder Entwicklungszyklus wird im Scrum über einen sogenannten Sprint festgelegt. Jeder Sprint dauert ungefähr zwei bis vier Wochen. Während des Sprints ist es untersagt, das Team zu stören - zum Beispiel durch das Einwerfen neuer Anforderungen. Der Product-Owner legt seine Anforderungen im Product-Backlog ab. Vor jedem Sprint legt das Team fest, welche Anforderungen sie in dem kommenden Sprint erfüllen können. Die im Sprint zu bearbeitenden Anforderungen liegen im Sprint-Backlog. Hat der Sprint begonnen, wird der Sprint-Backlog nicht mehr verändert.

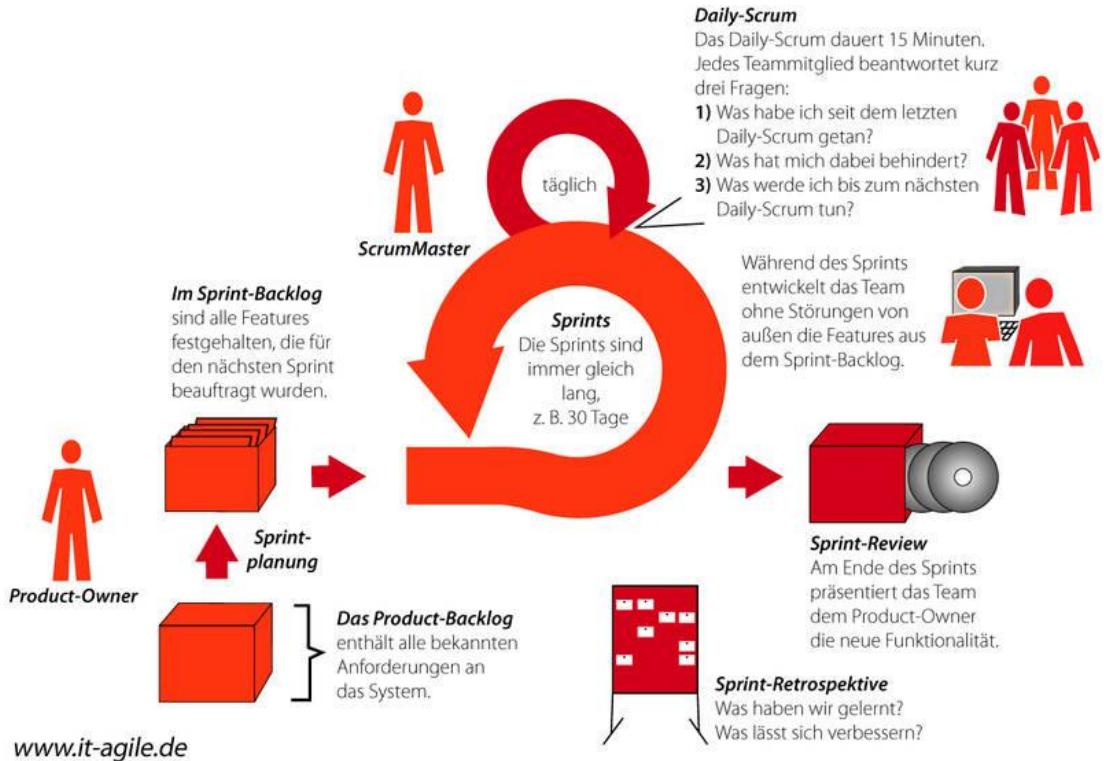


Abbildung 22: Übersicht Scrum [IQ17]

Zentraler Punkt während des Scrums ist das tägliche (Daily) Scrum. Im Daily-Scrum werden drei wesentliche Fragen behandelt:

- Was habe ich seit dem letzten Daily-Scrum getan?
- Was hat mich dabei behindert?
- Was werde ich bis zu nächsten Daily-Scrum tun?

Sinn dieses Meetings ist, dass das Entwicklerteam dynamisch auf Probleme reagieren und seine Ressourcen während des Scrums an die Problempunkte verlagern kann. Für jede Anforderung im Sprint werden Tests definiert, die Teil des Sprints sind. Eine Anforderung ist erst dann erledigt, wenn auch deren Test positiv ausfällt.

Am Ende jedes Sprints werden zwei Meetings durchgeführt. Zum einen das Sprint-Review mit dem Product-Owner und zum anderen ein sogenanntes Sprint-Retrospektive. Das Sprint-Review zeigt dem Product-Owner das Ergebnis des letzten Sprints. In der Sprint-Retrospektive reflektiert das Team die eigene Leistung während des Sprints und sucht gemeinsam nach Optimierungsmöglichkeiten für den nächsten Sprint.

3.4.3 V-Modell oder agile Softwareentwicklung?

Das V-Modell ist in der Medizintechnik mittlerweile etablierter Standard geworden. Grund ist, dass sich das V-Modell sehr gut eignet, um komplexe Systeme mit einem hohen *Sicherheitsstandard* zu entwickeln. Die neuen agilen Methoden genießen jedoch eine immer höhere Beliebtheit. Wie bereits erwähnt, ähnelt die vorgeschlagene Entwicklungsmethodik⁷⁷ der DIN EN 60601-1-4 sehr stark dem V-Modell. Für die Entwicklung neuer Prototypen ist das V-Modell - meiner Meinung nach - nur begrenzt einsetzbar, denn durch die in dem V-Modell angestrebte Vorgehensweise sind große Iterationszyklen angestrebt, bei denen man nur sehr langsam auf neu hinzukommende Anforderungen reagieren kann. Beim Prototyping ist es jedoch üblich, dass ständig neue Anforderungen in die Entwicklung mit einfließen. Abbildung 23 verdeutlicht dies.

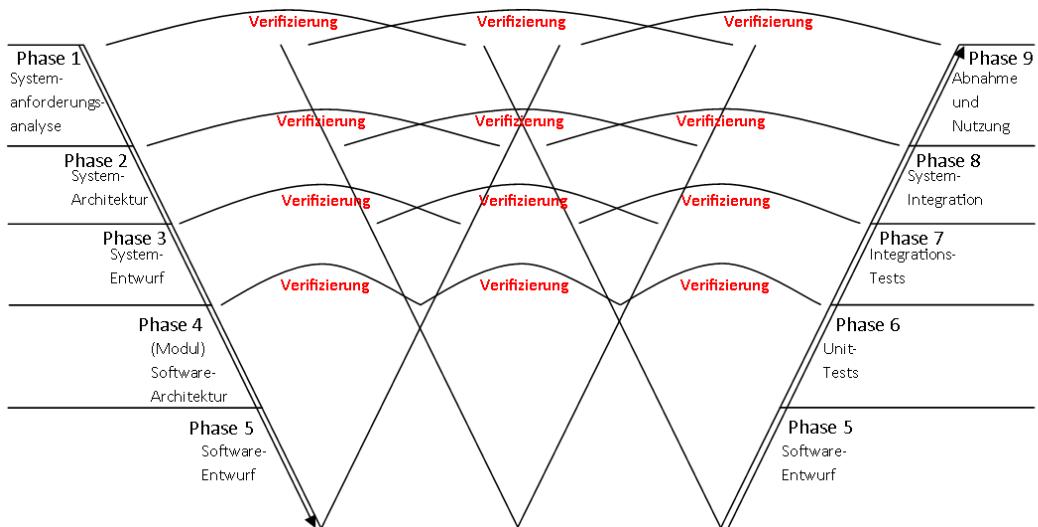


Abbildung 23: Beispiel für iterative Vorgehensweise im V-Modell (selbst erstellt)

Anders verhält es sich bei der agilen Scrum-Methode. Hier werden wesentlich kürzere Iterationen angestrebt. Mit jedem Sprint können zuerst kritische Teile des Prototyps entwickelt und untersucht werden. Gegebenenfalls kann man dann im darauffolgenden Sprint auf die vorherigen Probleme reagieren. Durch die kürzeren Iterationen können in solchen Fällen Kosten und Energie gespart werden. Dies ist einer der wesentlichen Vorteile von agilen Methoden beim Prototyping. Nachteil wiederrum ist, dass beim Scrum das komplexe System nicht zerlegt wird. Somit steigen Risiken durch zu hohe Komplexität und Architekturfehler.

Die beste Lösung wäre eine Kombination aus beiden Methoden. Diesen Gedanken unterstützt das Promo-Video der Firma Eckelmann AG⁷⁸. Für die Entwicklung von

⁷⁷ Siehe Kapitel 2.3.2, Abbildung 9, S. 22

⁷⁸ Eckelmann AG. „ECKELMANN AG: V-Modell und agile Softwareentwicklung nach Scrum“. [EKAG2010] (Siehe im Literaturverzeichnis unter Internetquellen, S. 108)

Prototypen vermaschen sie das V-Modell mit der agilen Scrum-Methode. Laut dem Eckelmann-Video wird im V-Modell⁷⁹ die Scrum-Methode in den Phasen 3-6 angewandt.

Aus diesen Gründen wurde die Scrum-Methode als Software-Entwicklungsverfahren für die Phasen 3-6 in der vorgeschlagenen Entwicklungsmethode gewählt. In den vorangegangenen Kapiteln wurden bereits die Phasen 1 und 2 der Methodik vorgestellt. Im Eckelmann-Video wurde darauf hingewiesen, dass vor Beginn der Scrum-Methode die Anforderungen ausgiebig nach dem V-Modell ausgearbeitet wurden. In dieser Arbeit werden in den Phasen 1 und 2 die Anforderungen des Gesamtsystems richtlinienkonform nach DIN EN 60601-1-4 ausgearbeitet.

Zu Beginn der dritten Phase müssen nach DIN EN 60601-1-4 zusätzlich die Anforderungen des Subsystems, inklusive Spezifikationen auf Basis der Gesamtspezifikation, ausgearbeitet werden. Diese Anforderungsspezifikation dient als Grundlage für die User-Stories der kommenden Scrum-Methode in den Phasen 3-6. Die Anforderungsspezifikation des Subsystems beinhaltet außerdem die Architektur und die Spezifikation der Schnittstellen.

Des Weiteren muss die Scrum-Methode nach DIN EN 62366 den Inhalt für folgende Schritte geben:

- Spezifikation der Gebrauchstauglichkeit
- Gestaltung und technische Umsetzung der GUI
- Verifizierung/Validierung der Gebrauchstauglichkeit

Die Spezifikation der Gebrauchstauglichkeit beinhaltet weitere Anforderungen, die als Grundlage in die User-Stories einfließen. Gestaltung, technische Umsetzung der GUI und Verifizierung/Validierung der Gebrauchstauglichkeit müssen während des Sprints im Scrum durchgesetzt werden. Gleiches gilt für die im Risikomanagement vorgeschriebenen Maßnahmen nach DIN EN ISO 14971⁸⁰.

⁷⁹ Siehe beispielsweise Abbildung 23, S. 55

⁸⁰ Siehe Kapitel 2.3.3, S. 25

3.5 Phase 7 – 8 - Abschluss der Entwicklung

Nach der Evaluation der Submodule verlangt die DIN EN 60601-1-4 eine vollständige Evaluation (Phase 7: Verifizierung/Phase 8: Validierung) des Gesamtsystems. Die Norm DIN EN 60601-1-4 schlägt folgende Methoden zur Verifizierung des Gesamtsystems vor:

- Walkthroughs und Besichtigen
- Statische/dynamische Analysen
- White-/Black-Box-Prüfungen

Welche Methoden der Hersteller für die Evaluation des Medizinproduktes letztendlich verwendet, ist ihm freigestellt.

Ähnlich verhält es sich auch mit den Vorgaben⁸¹ aus der DIN EN 62366. Die vorgestellte Vorgehensweise von C. Backhaus ist, meiner Meinung nach, eine gute und richtlinienkonforme Vorgehensweise⁸².

Durch die Rechtslage in Deutschland⁸³ ist für die Validierung in Phase 8 außerdem eine klinische Studie unumgänglich. Die klinische Studie kann vom Hersteller so gestaltet werden, dass sie gleichermaßen das Gesamtsystem und die Gebrauchstauglichkeit validiert.

⁸¹ Siehe Kapitel 3.1, S. 37

⁸² Siehe Abbildung 16, S. 40

⁸³ Siehe Kapitel 2.2.2, S. 11

4 Software des Medikamentendispensers im HomeCare-Bereich

4.1 Anforderungsanalyse des Gesamtsystems

Rechtliche und grundlegende Anforderungen an das System wurden bereits in den Kapiteln 1 und 2 angesprochen. Der ANHANG 11 zeigt eine Übersicht dieser Anforderungen.

In dieser Arbeit wird der Fokus auf technische und usabilitybezogene Anforderungen gelegt. Andere Anforderungen, wie beispielsweise die Dokumentation, werden nicht näher betrachtet. Nach C. Harböck⁸⁴ müssen zu den bereits ermittelten Anforderungen außerdem die folgenden Punkte in der Ausgangsspezifikation behandelt werden⁸⁵:

Ersteller, Ausgabedatum, (Version) und Freigabe

- Ersteller: FH-Science
- Ausgabedatum: 01.11.2011
- Version: 1.0
- Freigabe: 18+

Allgemeine Beschreibung/Programmvorhaben

Siehe Zweckbestimmung in Kapitel 2.2.1, S. 10.

Ein-/Beschränkungen der HW/SW

Die Anwendung der Steuersondware ist auf PC-Systemen mit Touchscreens vorgesehen. Das PC-System muss die Anforderungen der DIN EN 60601-1 erfüllen.

Betriebssysteme, Programmiersprachen, Entwicklungsumgebung (DV-Systemumgebung)

- Betriebssysteme: Windows XP/7
- Programmiersprachen: C# (.NET 4.0), XML/XSL
- Entwicklungsumgebung: Visual Studio 2010

Vernetzung

In Version 1.0 ist eine Vernetzung mit anderen Geräten nicht vorgesehen.

⁸⁴ C. Harböck. [HARB2001], S. 62

⁸⁵ Die Punkte wurden mit den Informationen zu dieser Arbeit ergänzt.

Liste der gewünschten/ungewünschten Funktionen und Eigenschaften

Die Funktionsspezifikation ist tabellarisch dargestellt und erläutert alle risikobezogenen Funktionen des PEMS. Die Anforderung des Zertifizierungsamtes⁸⁶ deckt sich mit der Anforderung aus der DIN EN 60601-1-4.

| Index | Funktion | Gewünschte Funktion | Mögliche Fehlfunktion |
|-------|---|--|---|
| F1 | Medikamentenausgabe | Korrekte Ausgabe der Medikamente nach einem im Verordnungssatz beschriebenen Gabe-Schema | Fehlmedikation (Falsche Ausgabe oder unterlassene Ausgabe) |
| F2 | Kommunikation mit der Hardware via Touchscreen mit dem Benutzer | Zur Bedienung der Hardwarekomponente Zur Bedienung der Softwarekomponente Warnhinweise | Fehlbedienung, bedingt durch unzureichende Gebrauchstauglichkeit Fehlbedienung, bedingt durch Softwarefehler Ausfall (Hardware-/Softwarefehler) |
| F3 | Kommunikation Steuer-SW und Hardwarekomponente | Korrekte Übertragung der Daten, in beide Richtungen via USB | Ausfall der Hardware Fehlinterpretation der Befehle Falschübertragung der Befehle |
| F4 | Bereitstellung von Informationen via Touchscreen | Korrekte Bereitstellung von Informationen | Bereitstellung von Fehlinformationen |
| F5 | Controlling durch unabhängige Softwarekomponente | Kontrolle der Eingabe (Verordnungsdatensatz) Kontrolle der Ausgabe (Befehl zur Hardware) Aufzeichnung von Einnahmeinformationen (Zur Kontrolle des Patienten für den Arzt) | Fehlinterpretation (Verordnungsdatensatz /Befehl) Ausfall der Controllersoftware |

⁸⁶ Siehe Kapitel 3.2, S. 43

| Index | Eigenschaft | Beschreibung |
|-------|-----------------------------|--|
| E1 | Handlich/Hohe Bedienbarkeit | Durch die Verwendung im HomeCare-Bereich und der gewünschten Benutzergruppe ist eine hohe Bedienbarkeit Voraussetzung. |
| E2 | Einfach zu bedienen | Der Benutzer muss schnell die gewünschten Funktionen verstehen und anwenden können. |
| E3 | Intuitiv zu bedienen | Der Benutzer muss zu jeder Zeit im Ablauf die Übersicht besitzen, was er als nächstes machen muss. |
| E4 | Einfaches Nachfüllen | Die Lagerbehältnisse müssen einfach nachzufüllen sein, ohne dass Verwechslungen oder andere Fehler auftreten können. |

Tabelle 6: Gewünschte/Ungewünschte Funktionen/Eigenschaften (selbst erstellt)

Benutzerschnittstellen Sprache, (Bildschirm-) Ein- und Ausgabemasken, Kommandos, Tasturbefehle, Druckerausgaben, E/A-Geräte

- Sprache: Deutsch
- Ein-, Ausgabemasken: keine vorgesehen
- Kommandos: keine vorgesehen
- Tasturbefehle: Standard Windows
- Druckerausgaben: keine vorgesehen
- E/A-Geräte: Touchscreen/Tastatur/Maus

Sonstige Schnittstellen (zu anderen Systemen)

Es ist eine USB-Schnittstelle für die Kommunikation zwischen Hardwarekomponente und Softwarekomponente vorgesehen.

Validierung von Fremdsoftware

Verwendung von Fremdsoftware nicht vorgesehen.

Gefährdungsmindernde Anforderungen

Es gelten die Anforderungen aus den Normen.

Qualifikationsvorgaben für den Anwender

Die Usability der Benutzeroberfläche für den Patienten und die Sicherheit des Systems haben die höchste Priorität.

4.1.1 Ermittlung von Anforderungen aus einer Prozessanalyse

Um die Anforderungsanalyse in Bezug auf die Gebrauchstauglichkeit zu vervollständigen, ist eine Prozessanalyse⁸⁷ nach C. Backhaus vorgesehen. Aus der Prozessanalyse heraus können Prozessanforderungen ermittelt werden, welche als Anforderungen an das Medizinprodukt gestellt werden. Der gewünschte zu unterstützende Prozess ist die häusliche Medikamententherapie. Daraus ergeben sich drei Prozessszenarien:

1. Häusliche Medikamententherapie vorbereiten
2. Häusliche Medikamententherapie
3. Häusliche Medikamententherapie nachbereiten

ANHANG 2 zeigt die Prozessszenarien im Detail.

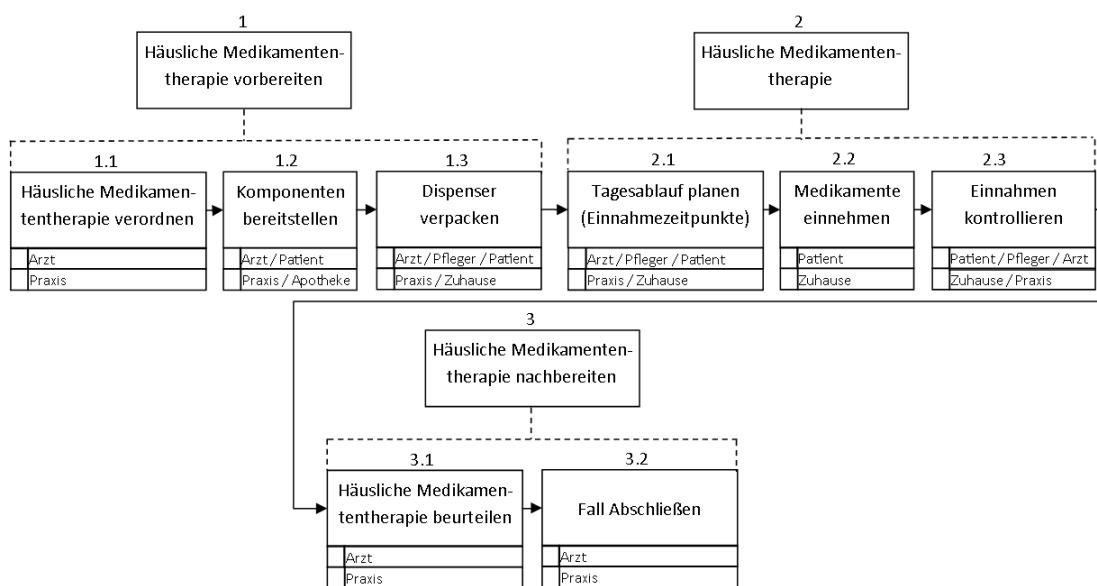


Abbildung 24: Prozessanalyse nach C. Backhaus (selbst erstellt)

Aus der Prozessanalyse heraus haben sich folgende Prozessanforderungen für Version 1.0 herauskristallisiert:

Legende: M = Must Have, N = Nice to Have

| Nr. | Prozessanforderung | Priorität |
|-----|---|-----------|
| 1 | Die Anwendung muss in allen Bedienschritten selbsterklärend sein. | M |
| 2 | Die Einlagerung der Medikamente muss einfach gehalten sein. | M |
| 3 | Die Eingabe des Gabe-Schemas muss verständlich sein. | M |
| 4 | Ein integrierter Terminkalender muss vorhanden sein. | N |
| 5 | Ein Erinnerungssystem muss vorhanden sein. | M |
| 6 | Individuelle Benutzerwünsche müssen berücksichtigt werden. | N |
| 7 | Erfassung ausgegebener Medikamente, inklusive Ausgabezeitpunkt | M |

⁸⁷ Siehe Kapitel 3.2, S. 43

| | | |
|----|--|---|
| 8 | Prüfung des Gabe-Schemas, mit Hinweisen an den Benutzer bei Auffälligkeiten | M |
| 9 | Ergonomisches Erinnerungssystem für kognitiv schwächere Benutzer | M |
| 10 | Belehrungssystem bei nicht eingehaltenen Terminen | M |
| 11 | Das Gerät soll eine Einweisung und Hilfestellung bereitstellen. | N |
| 12 | Das Gerät muss einfach und schnell aufzurüsten sein. | N |
| 13 | Die Alarme müssen eindeutig und verständlich sein. | M |
| 14 | Das Gerät muss mobil einsetzbar sein. | N |
| 15 | Das Gerät muss leicht zu reinigen sein. | M |
| 16 | Das Gerät soll alle Entscheidungen des Benutzers dokumentieren. | M |
| 17 | Das Gerät soll fällige Wartungsintervalle anzeigen. | M |
| 18 | Es soll ein Standby-Modus vorhanden sein. | M |
| 19 | Beinhaltende Medikamente sollen erkennbar sein. | M |
| 20 | Das Gerät soll bei lösbarer Fehlern eine automatische Fehlerbeseitigung besitzen. | N |
| 21 | Das Gerät soll über eine Online-Verbindung mit Daten gespeist werden (z.B. Gabe-Schema/pharmak. Profil). | N |
| 22 | Das Gerät soll die aufgezeichneten Daten für eine Auswertung vorbereiten. | N |

Tabelle 7: Prozessanforderungen aus der Prozessanalyse (selbst erstellt)

4.1.2 Festlegung des USB Protokolls

Vor dem ersten Entwurf muss eine weitere wichtige Anforderung analysiert werden. Für die Kommunikation zwischen Hardware und Software wurde eine USB-Schnittstelle gewählt. Alle nötigen Grundlagen für die USB-Kommunikation sind in ANHANG 3 zu finden.

Für die Übertragung über diese Schnittstelle muss ein Protokoll bestimmt werden. Dabei stellt sich die Frage, ob ein standardisiertes Protokoll für die Übertragung genügt oder dafür extra ein rechtlich konformer Treiber entwickelt werden muss. Aus der Spezifikation des Treibers oder des Standards heraus muss die Deskriptor-Struktur der Hardware angepasst werden. Über verschiedene Gespräche und Recherchen haben sich drei Möglichkeiten für die Umsetzung einer USB-Schnittstelle ergeben. Zwei Möglichkeiten gehen über Device Classes und die dritte Möglichkeit ist ein eigener Treiber.

Die führenden Soft- und Hardwarefirmen haben relativ früh nach der Entwicklung des USB-Standards mit der Entwicklung von sogenannten Device Classes angefangen. Diese Device Classes geben gewisse Standards für bestimmte Gruppen von USB Devices vor. Zum Beispiel stellt die HID Class (Human Interface Device Class) einen USB-Standard für Controller-Geräte dar (Maus, Tastatur, Temperaturmesser, etc.).

Seit kürzerer Zeit existiert auch eine spezielle Klasse für USB-Geräte aus dem medizinischen Sektor. Diese Klasse heißt PHDC⁸⁸. Folgende Möglichkeiten haben sich nun daraus ergeben:

- Verwendung eines eigenen Treibers
- Verwendung der HID-Klasse
- Verwendung der PHD-Klasse

Der wesentliche Unterschied zwischen Device-Klassen und einem eigenen Treiber ist, dass bei Device-Klassen die Deskriptoren einer standardisierten Norm folgen. Verwendet das USB Device eigene Deskriptoren, ist ein eigens dafür entwickelter Treiber nötig. Verwendet das USB Device aber standardisierte Deskriptoren, nach beispielsweise HID, benötigt man *keinen* zusätzlichen Treiber.

Personal Healthcare Device Class (PHDC)

Im November 2007 wurde dem USB-Standard die *Personal Healthcare Device Class* hinzugefügt. Ziel dieser Klasse ist es, Interoperabilität zwischen USB Hosts und USB Devices aus dem Gesundheitssektor zu gewährleisten und dabei gleichzeitig den hohen medizinischen Standards gerecht zu werden.

Der Focus der Klasse liegt dabei auf USB Devices, die standardisierte Daten, welche auf dem ISO/IEC 11073-20601 *Base Exchange Protocol* Standard basieren, in sporadischen oder periodischen Abständen zu den jeweiligen USB Hosts zu senden (PC, iPod, Handy, etc.). Die PHD-Klasse besitzt dafür eigens einen eigenen Deskriptor-Typ, der das Protokoll festlegt. Die übertragene Datenmenge kann bei PHDC von kleinen Messungen bis hin zu großen Paketen variieren.

Dem Host selbst werden keine größeren Aufgaben zugeteilt. Der Host soll lediglich als Empfänger dienen und hin und wieder das Device mit Metadaten oder anderen Informationen versorgen.

Abbildung 25 zeigt einen Ausschnitt aus den angegebenen Use Cases⁸⁹ der PHDC Klasse. Ein wesentlicher Nachteil von PHDC ist, dass diese Klasse momentan noch kein USB 3.0 unterstützt. Dies wird in Zukunft aber wahrscheinlich nachgereicht.

⁸⁸ Personal Healthcare Device Class

⁸⁹ USB.org. „USB – Device Class Definition for Personal Healthcare Devices“. [\[PHDC2007\]](#), S.9f

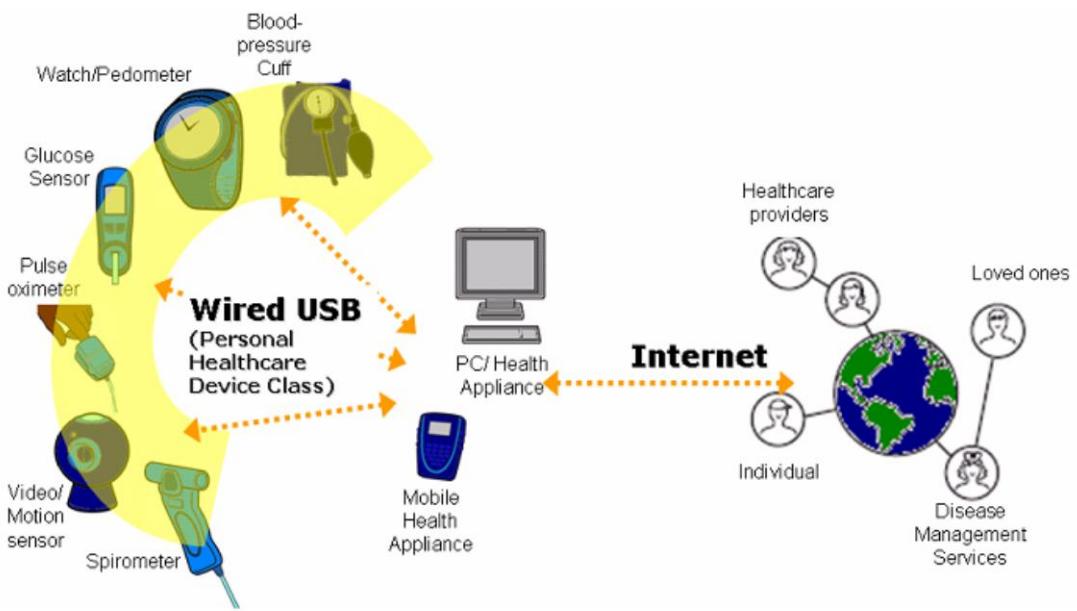


Abbildung 25: Umfeld USB-PHD-Klasse [PHDC2007], S. 9

PHDC unterstützt außerdem verschiedene Übertragungskonfigurationen, welche für bestimmte Anwendungen, wie z.B. EKG-Applikationen, unabdingbar sind. Die Konfigurationen beschreiben einmal die maximale Übertragungstoleranz, sowie die Ausfallsicherheit. Je nachdem wie kritisch die Daten sind und wie schnell sie übertragen werden müssen, kann eine der folgenden Konfigurationen gewählt werden:

| Übertragungsart mit Geschwindigkeit/Ausfallsicherheit | Eigenschaft | Übertragungsart |
|---|-----------------------------------|-----------------|
| Low/Good | <20ms, gute Ausfallsicherheit | Interrupt |
| Medium/Good | <200ms, gute Ausfallsicherheit | Bulk |
| Medium/Better | <200ms, bessere Ausfallsicherheit | Bulk |
| Medium/Best | <200ms, beste Ausfallsicherheit | Bulk |
| High/Best | <2s, beste Ausfallsicherheit | Bulk |
| Very high/Best | <20s, beste Ausfallsicherheit | Bulk |

Tabelle 8: Übertragungsmodi PHDC

Für eine Übertragung von Host zu Device stehen nur die Konfigurationen *Medium/Best*, *High/Best* und *Very high/Best* zur Verfügung. Diese eignen sich um mittlere, sowie große Datenmengen zu übertragen. Die Konfiguration *Low/Good* ist für kleinste Datenmengen in schnellen periodischen Abständen gedacht (EKG).

Human Interface Device Class (HID)

Die HID-Klasse ist schon sehr lange auf dem Markt. Die letzte offizielle Revision⁹⁰ ist vom 27.06.2001. Hauptsächlich ist die Klasse für Eingabegeräte gedacht, die Interaktionen mit dem System erlauben (z.B. Maus/Tastatur/ Gamecontroller). Aber auch interaktionslose Endgeräte, wie z.B. Barcode-Lesegeräte, Voltmeter oder Thermometer sind über die HID-Klasse steuerbar.

Die HID-Klasse unterstützt Rückmeldungen zum Device, um beispielsweise spezielle Displays oder Audio-Rückmeldungen zu unterstützen. Es wird jedoch nur der Interrupt-Transfer-Modus von der Human-Interface-Device-Klasse unterstützt. Somit ist eine Übertragung von großen Datenmengen über die HID-Klasse nicht möglich.

Für die Spezifikation der HID Devices werden HID-, Report- und Physical-Deskriptoren verwendet. Jedes Device kann mehrere HID-Deskriptoren besitzen und jeder HID-Descriptor kann mehrere Physical-, sowie Report-Deskriptoren besitzen. Abbildung 26 zeigt eine beispielhafte Descriptor-Struktur.

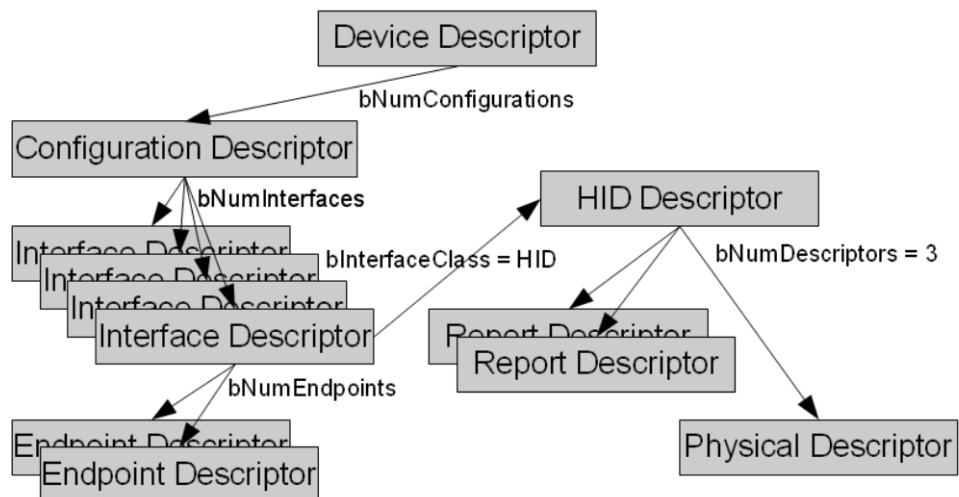


Abbildung 26: Descriptor HID Beispiel (selbst erstellt)

Ein Report Descriptor beschreibt alle vom Device erzeugten Daten und deren Inhalt (Beispielsweise eine Position oder ein Button Status). Physical-Deskriptoren sind optional und beschreiben, welches Körperteil ein bestimmtes HID-Controller-Element bedient (Beispielsweise linker Maus Button = rechter Finger).

⁹⁰ USB.org. „USB – Device Class Definition for Human Interface Devices (HID)“. [HIDC2001]

HID, Treiber oder PHDC

Von den ersten Überlegungen her würde die HID-Klasse für eine Realisierung der Kommunikation zwischen Device und Software vollkommen ausreichen. Die Größe der zu übertragenden Befehle und Antworten beschränkt sich nur auf wenige Byte und eine Übertragung im Interrupt-Modus ist ebenfalls möglich, jedoch würde das Device logisch geschlussfolgert nicht in die Kategorie *Human Interface* fallen. Ebenfalls ist die doppelte Sicherheit mit der HID-Klasse nicht gewährt.

Die PHDC hingegen besitzt für diese Problematik eine eigene Datenübertragungssicherheit. In den Bulk-Übertragungsmodi wird die Übertragung durch die Hardware dreimalig wiederholt. Mit diesem Vorgehen ist eine sehr hohe Datensicherheit gegeben.

Letztendlich kommen nur die PHDC oder ein eigener Treiber im Bulk-Modus in Frage. Da in der Medizintechnik der Trend hin zur Interoperabilität und Standardisierung zeigt, ist es am sinnvollsten die PHDC für das Device zu wählen. Über die PHDC ist das Device vom Host direkt als Medizinprodukt identifizierbar. Es muss kein eigener Treiber geschrieben werden.

Wegen dieser Gründe wurde für diese Arbeit eine Umsetzung über die PHDC gewählt. Der ANHANG 12 zeigt die Spezifikation der Hardwaredeskriptoren für die PHDC für unser Projekt.

4.2 Erster Entwurf des Gesamtsystems

4.2.1 Architekturspezifikation PEMs (Gesamtsystem)

Eine grundlegende Systemidee und Spezifikation mit Einbeziehung der Umwelt zeigt bereits Kapitel 1.1 und 1.2.

Architektur PEMs

Anhand der Grundlagen wird der HomeCare Medidisp in Abbildung 27 als PEMs⁹¹ dargestellt. Dabei wird Fokus auf das Subsystem des Steuermoduls gelegt.

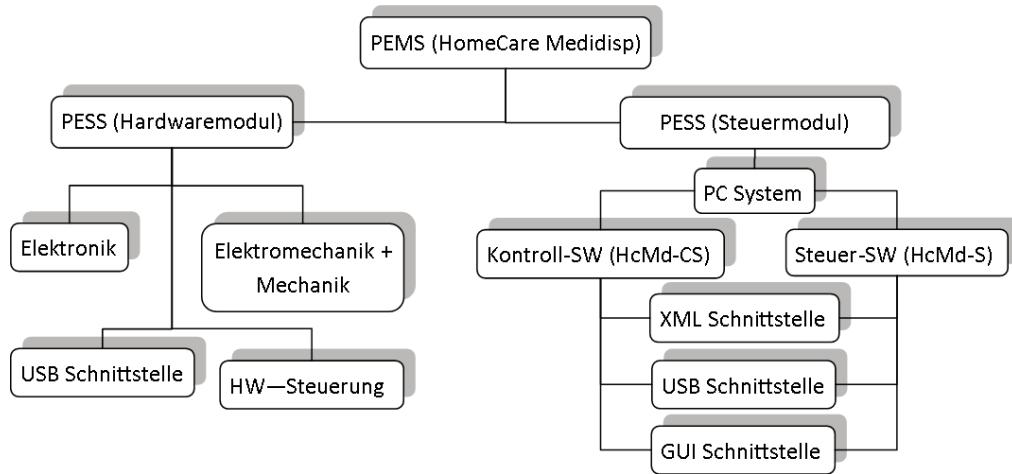


Abbildung 27: PEMs/PESS Struktur für den HC Medidisp⁹²

Das PEMs des Steuermoduls (rechts) besteht aus einem PC-System mit Betriebssystem, den beiden Softwaremodulen *HcMd-CS* und *HcMd-S* und deren systemübergreifenden Schnittstellen. Das PEMs des Hardwaremoduls (links) spiegelt sämtliche Hardwarekomponenten des elektronischen Dispensers wieder. Die Steuersoftware nutzt die USB- und XML-Schnittstelle zur Kommunikation mit der Hardware und zum Einlesen der therapiebezogenen Daten. Die Kontrollsoftware soll die Kommunikation über diese Schnittstellen überwachen. Zwei kurze Beispiele:

- Fall 1:
 - Steuer-SW: *Ist diese Ausgabe korrekt? Ich nutze dazu folgende pharmakologische Profile*
 - Kontroll-SW: *Ja, ist sie.*
- Fall 2:
 - Kontroll-SW: *Stimmen meine pharmakologischen Profile mit deinen überein?*
 - Steuer-SW: *Ja.*

⁹¹ Zweite Überlegung. Der erste Ansatz ist in Kapitel 2.3.2, Abbildung 8, S. 20 zu sehen.

⁹² Nach DIN EN 60601-1-4, S. 26

Schnittstellen PEMS

XML-Schnittstelle: Die XML-Schnittstelle dient zum Einlesen der Verordnungsdatensätze und pharmakologischen Profile, sowie zur Ausgabe von Daten. Es ist angedacht, dass die Kontrollsoftware Patientenprofile⁹³ erstellt und diese über die XML-Schnittstelle ausgibt. Die Kontrollsoftware überwacht außerdem den gesamten Austausch über die XML-Schnittstelle. Die XML-Datensätze werden über Schema-Dateien modelliert und somit ist die Konsistenz der Datensätze gegeben.

GUI-Schnittstelle: Das GUI wird auf ein Touchscreen angepasst sein. Bei der Steuersoftware soll dabei besonders Wert auf die eingeschränkte Benutzergruppe legen. Das GUI der Kontrollsoftware soll praktisch angelegt sein.

USB-Schnittstelle (Kapitel 4.1.2): Die USB-Schnittstelle dient zur Kommunikation zwischen Hardwarekomponente und Softwarekomponente. Die Kommunikation über die USB-Schnittstelle muss zweifach, durch die Kontroll- und Steuersoftware, kontrolliert werden.

Weitere Anforderungen durch die Architektur

PC-System: Das PC-System des PESS (Steuermobil) soll laut DIN EN 60601-1 gewisse Ableitströme aufweisen. Durch die Verwendung im HomeCare-Bereich ist es fragwürdig, diesen Punkt einzuhalten und deshalb wird darauf verzichtet. Es kann ein handelsüblicher PC verwendet werden.

Abschirmung PESS (Hardwaremodul): Die Abschirmung des PESS (Hardwaremodul) muss soweit ausreichend sein, dass das PESS durch Interferenzen aus dem häuslichen Umfeld nicht beeinflusst werden kann.

Sicherheit der USB-Schnittstelle (Kapitel 4.1.2): Die Kommunikation über die USB-Schnittstelle muss soweit abgesichert sein, dass eine fälschlicherweise interpretierte Verordnung nicht möglich ist. Außerdem darf die Kommunikation nicht über Umwelteinflüsse beeinflussbar sein.

4.2.2 Spezifikation der Anwendungen

Wie in Kapitel 3.3 beschrieben, verlangt die DIN EN 62366 für den gebrauchstauglichkeitsorientierten Entwicklungsprozess eine Spezifikation der Anwendung. Die nächsten Punkte enthalten die vorgesehene klinische Indikation, vorgesehene Nutzungsbedingungen, physikalische Funktionsweise und die vorgesehenen Benutzerprofile.

⁹³ Beispielsweise: Einnahmezeitpunkte, Nutzung

Vorgesehene klinische Indikation

Unterstützung der häuslichen Medikamententherapie. Dabei werden Medikamente verwendet, die den gesamten Körper beeinflussen können. Zum Beispiel Betablocker.

Vorgesehene Nutzungsbedingungen

Die Anwendung ist im häuslichen Umfeld angedacht. Es ist darauf zu achten, dass die Lagerbehältnisse der Medikamente korrekt gereinigt sind. Gleich wie bei einem Dispenser. Es wird davon ausgegangen, dass das System ungefähr 3 bis 6 Mal am Tag verwendet wird. Dabei ist das System an einen Ort mit einer Stromquelle gebunden. Das System befindet sich ständig in Bereitschaft (Standby) und sendet gegebenenfalls Erinnerungen. Das Steuermodul kann räumlich getrennt vom Hardwaremodul sein und ist ebenfalls auf eine Stromquelle angewiesen. Es ist aber darauf zu achten, dass die Kabelverbindung zwischen den Modulen ständig gegeben ist. In späteren Versionen wird Wireless USB angestrebt.

Physikalische Funktionsweise

Nach Erinnerung oder ohne Erinnerung wird ein Becher mit bestimmter Kennung in den Abgabeschacht gestellt. Das System erkennt den Becher, sowie deren rechtmäßigen Besitzer und füllt die nach dem Verordnungsdatensatz eingegebenen Gabe-Schema und auf Basis weiterer Prüfungen die erforderlichen Medikamente in den Becher.

Vorgesehene Benutzerprofile

Das System soll auf zwei Hauptgruppen von Benutzerprofilen zugeschnitten sein.

Hauptgruppe 1: Der behandelnde Hausarzt sowie Hersteller und Servicepersonal.

Hauptgruppe 2: Der Patient im häuslichen Umfeld.

Es wurden detaillierte Benutzerprofile nach dem Beispiel von C. Johner et. al.⁹⁴ erstellt. Diese sind in ANHANG 4 zu finden.

⁹⁴ C. Johner et. al. „*Basiswissen Medizinische Software: Aus- und Weiterbildung zum Certified Professional for Medical Software*“. [CJOH2011]

4.2.3 Ermittlung der häufig genutzten Funktionen

Aus der vorangegangenen Prozessanalyse (Kapitel 4.1.1) wurden folgende häufig genutzte Funktionen abgeleitet:

- *Funktion 1:* Statusrückmeldungen über bestimmte Einnahmetermine oder vergessene Einnahmen (Erinnerungssystem)
- *Funktion 2:* Medikamentenausgabe über einen Becher
- *Funktion 3:* Eingabe von Verordnungen (Verordnungsdatensätze)
- *Funktion 4:* Eingabe von pharmakologischen Profilen
- *Funktion 5:* Ausgabe des Patientenprotokolls (Protokollierung der Kontrollsoftware)
- *Funktion 6:* Wartung/Auffüllen der Lagerbehälter für Medikamente

4.2.4 Festlegung der Hauptbedienfunktionen

Aus den häufig genutzten Funktionen müssen die Hauptbedienfunktionen des Systems bestimmt werden. Die Hauptbedienfunktionen eines Systems verlangen besondere Aufmerksamkeit in der Risikoanalyse. Grund ist die besonders häufige Nutzung dieser Funktionen. Die zweithöchste Priorität erhalten die häufig genutzten Funktionen.

Hauptbedienfunktionen 1 (Funktion 1, aus Kapitel 4.2.3):

Die erste Hauptbedienfunktion bildet das Erinnerungssystem des HomeCare Medidisp. Diese Funktion arbeitet ständig im Hintergrund und muss die gewünschten Informationen und Erinnerungen korrekt und gebrauchstauglich an die Benutzergruppe (*Hauptgruppe 2*) weitergeben.

Hauptbedienfunktionen 2 (Funktion 2, aus Kapitel 4.2.3):

Die zweite Hauptbedienfunktion ist die Medikamentenausgabe über einen Becher. Hier ist im Aspekt der Gebrauchstauglichkeit besonders das Zusammenspiel von Software und Hardware zu beachten. Der Benutzer (*Hauptgruppe 2*) muss zu jedem Zeitpunkt wissen, was mit der Hardware zu tun ist.

Hauptbedienfunktionen 3 (Funktion 4, aus Kapitel 4.2.3):

Die dritte Hauptbedienfunktion beschreibt die Eingabe der Verordnungsdatensätze. Der Benutzer (*Hauptgruppe 1*) muss sämtliche Daten intuitiv und korrekt eingeben können. Es dürfen bei der Eingabe keine Fehler aufgrund der Gebrauchstauglichkeit auftreten⁹⁵.

⁹⁵ Beispielsweise die Eingabe einer falschen Verordnung oder falscher Präparate.

4.2.5 Erste Risikomaßnahmen

Risiko-Analyse

Aus dem bestimmungsgemäßen Gebrauch⁹⁶ und den Merkmalen des Systems (Kapitel 4.1 – 4.2.4) werden erste mögliche Gefährdungen ermittelt. Bei der Ermittlung haben sich drei umfassende Gefährdungskategorien herauskristallisiert:

- Fehlmedikation (z.B. Überdosis, unterlassene Medikation)
- Elektronik (z.B. Stromschlag, Ableitströme)
- Kinetischer Krafteinfluss (z.B. Quetschungen, Klemmungen)

Auf Basis der Gefährdungen können mögliche Ursachen für diese Gefährdungen ermittelt werden. Mögliche Ursachen aufgrund der Hardware könnten nach DIN EN 60601-1 sein:

- Ungenügende Isolation
- Stromausfall
- Kurzschluss
- Verschleiß
- Schnittstellen nicht sicher gegen äußere Einflüsse
- Lagerung
- Material
- Felder (elektromagnetisch)
- Überspannung
- Bedienfehler, aufgrund der Gebrauchstauglichkeit

Nach C. Johner et. al.⁹⁷ gibt es bei medizinischer Software typische Ursachen für Gefährdungen. Die folgende Aufzählung ist angepasst an den HomeCare Medidisp:

- Gebrauchstauglichkeit ist nicht gegeben (Fehlbedienungen des Benutzers, aufgrund von zu vielen missverständlichen Informationen oder Alarmen)
- Absturz der Software (Verzögerung/unterlassene Medikation)
- Gebrauchsanweisung ist missverständlich dokumentiert
- Fehler in der Datenspeicherung (Persistenz nicht gegeben)
- Kommunikationsfehler zwischen den Schnittstellen (XML <-> SW, Steuer- und Kontrollsoftware, HW <-> USB <-> SW)
- Fehlinterpretation von Daten
- Gebrauch von fehlerhaften Fremdkomponenten (Open-Source-Bibliotheken)
- nicht genügend Systemressourcen
- zu geringe Datensicherheit

⁹⁶ Siehe Zweckbestimmung: Kapitel 2.2.1, S. 10

⁹⁷ C. Johner et. al. [CJOH2011], S.74

Die ermittelten Gefährdungen (Risiken) und Ursachen müssen weiter untersucht werden. Um den Hergang und eventuell noch weitere Ursachen zu ermitteln, eignet sich in dieser Phase eine Fehlerbaumanalyse (FTA)⁹⁸. Als top-event können die anfangs ermittelten Gefährdungskategorien dienen.

Im Folgenden wird näher auf die FTA des Risikos *Fehlmedikation* eingegangen. Dieses Risiko bezieht sich überwiegend auf den Softwarepart. Die Fehlmedikation stellt gleichzeitig auch das Hauptrisiko des HomeCare Medidsp dar und muss soweit wie möglich minimiert werden. Ein kompletter Fehlerbaum dazu ist in ANHANG 5 zu sehen. Die anderen beiden Gefährdungskategorien (Elektronik/kinetischer Krafteinfluss) beschränken sich mehr auf den Hardwarepart des Systems und werden nicht weiter behandelt.

Risikobewertung

Die Tabelle 9 zeigt die Einstufung der erkannten Gefährdungen aus der FTA. Die Risikobewertung wird nach der vorgeschlagenen Methodik aus Kapitel 2.3.3 vorgenommen. Bei Softwarefehlern (SWF) ist die Festlegung der Wahrscheinlichkeit auf 100 Prozent zu beachten. Grund dafür ist die Anforderung an die Risikobewertung aus der DIN EN 62304 (Kapitel 2.3.2). Alle softwarebedingten Fehler sind in dieser Risikobewertung vorerst mit der größten Wahrscheinlichkeit *Häufig* belegt. Die Einstufung des Schweregrades gestaltet sich in der Risikobewertung des Hauptrisikos etwas schwieriger. Das Hauptrisiko der Fehlmedikation kann in drei verschiedenen Fällen auftreten:

Fall 1. Schaden durch eine toxische Kombination von Medikamenten

Fall 2. Schaden durch das Weglassen von Medikamenten

Fall 3. Schaden durch eine Überdosis von Medikamenten

Fall 1. spiegelt Situationen, bei denen die Wirkstoffe der verabreichten Medikamente miteinander eine toxische Wirkung entfalten können, wieder. Der Schweregrad aller drei Fälle hängt im Wesentlichen von den verwendeten Medikamenten im System ab. Ist ein Medikament bei einer bestimmten Überdosis tödlich, dann ist der Schweregrad als *Katastrophal* einzustufen. Verursacht das Medikament bei einer Überdosis hingegen nur Beschwerden, würde der Schweregrad bei *Gering* liegen. Bei der folgenden Bewertung wird davon ausgegangen, dass Medikamente, die den höchsten Schweregrad verursachen können, verwendet werden.

Die Spalte *Mögliche Fälle* zeigt an, welche der drei oben genannten Fälle bei dieser Gefährdung auftreten können. Da auf den Hardwarepart nicht weiter eingegangen

⁹⁸ Siehe Kapitel 2.3.3, S. 27

wird, ist bei Hardwarefehlern (HWF) vorerst die Auftretenswahrscheinlichkeit auf ein *Minimum* gesetzt. Es wird davon ausgegangen, dass in der Entwicklung des Hardwareparts die Hardwarefehler auf dieses Minimum reduziert werden. Fehler aufgrund der Gebrauchstauglichkeit (GbT) und auf Basis von fehlendem Wissen werden vorerst auf *Wahrscheinlich* festgelegt, da noch keine Auftretenswahrscheinlichkeiten diesbezüglich ermittelt werden können.

| Id.Nr. | Gefährdung | Mögliche Fälle | Wahrsch. | Vertretbarkeit |
|---------------|--|-----------------------|-----------------|-----------------------|
| HWF001 | Ausfall Kommunikation HW/SW Modul (USB) | Fall 2. | Unwahrsch. | Vertretbar |
| HWF002 | Fehler in der Mechanik (z.B. Motor dreht 1x zu viel) | Fall 1. – 3. | Unwahrsch. | Vertretbar |
| HWF003 | Ausfall Sensorik (Bechersensor/ Medikamentensor) | Fall 1. oder 3. | Unwahrsch. | Vertretbar |
| HWF004 | Ausfall Display | Fall 2. | Unwahrsch. | Vertretbar |
| HWF005 | Zu viele Medikamente fallen in den Ausgabeschacht | Fall 3. | Unwahrsch. | Vertretbar |
| HWF006 | Interner HW-Fehler (z.B. Verstopfung) | Fall 2. | Unwahrsch. | Vertretbar |
| SWF001 | Fehler im Modul für Interpretation des Gabe-Schemas | Fall 1. – 3. | Häufig | Unvertretbar |
| SWF002 | Fehler im Modul für Interpretation der pharmakologischen Profile | Fall 1. oder 3. | Häufig | Unvertretbar |
| SWF003 | Ausfall Kommunikation HW /SW Modul (USB-Modul) | Fall 2. | Häufig | Unvertretbar |
| SWF004 | Interner Steuer-SW-Fehler (z.B. fehlerhafte Anzeige) | Fall 1. – 3. | Häufig | Unvertretbar |
| SWF005 | Interner Controller-SW-Fehler | Fall 1. – 3. | Häufig | Unvertretbar |
| SWF006 | Absturz des Betriebssystems | Fall 2. | Häufig | Unvertretbar |
| SWF007 | Software besitzt falsches Datum/Uhrzeit | Fall 2. oder 3. | Häufig | Unvertretbar |

| | | | | |
|-----------|--|-----------------|------------|--------------|
| SWF008 | Fehlentscheidung zur Medikamentenausgabe aufgrund von zu wenig Informationen ⁹⁹ | Fall 2. oder 3. | Häufig | Unvertretbar |
| SWF009 | Fehlentscheidung zur Medikamentenausgabe aufgrund einer Fehlinterpretation ⁹⁹ | Fall 2. oder 3. | Häufig | Unvertretbar |
| AU01 | Stromausfall | Fall 2. | Unwahrsch. | Vertretbar |
| AU02 | Wasserschaden | Fall 2. | Unwahrsch. | Vertretbar |
| Wissen001 | Der Benutzer weiß nicht, wie das Gerät bedient wird. | Fall 2. | Wahrsch. | Unvertretbar |
| Wissen002 | Der Ben. weiß nicht, wie das Gerät eingeschaltet wird. | Fall 2. | Wahrsch. | Unvertretbar |
| Wissen003 | Der Ben. weiß nicht, wie ein VDS eingegeben wird. | Fall 1. – 3. | Wahrsch. | Unvertretbar |
| Wissen004 | Der Benutzer weiß nicht, wie ein pharmak. Profil eingegeben wird. | Fall 1. – 3. | Wahrsch. | Unvertretbar |
| GbT001 | Der Benutzer interpretiert die Anzeige falsch. | Fall 2. oder 3. | Wahrsch. | Unvertretbar |
| GbT002 | Der Benutzer interpretiert den Systemstatus falsch. | Fall 2. | Wahrsch. | Unvertretbar |

Tabelle 9: Erste Risikobewertung für Schwergrad *Katastrophal* (selbst erstellt)

Maßnahmen zur Risikominimierung (Risikokontrollmaßnahmen)

SWF001, SWF002: Durch die Modellierung einer Schema-Datei, können in XML keine Verwechslungen vorkommen. Es muss eine Schema-Datei jeweils für Verordnungsdatensätze und pharmakologische Profile existieren. Das Programm nimmt nur XML-Dateien an, die konform nach den Schema-Dateien aufgebaut sind.

SWF003: Durch die Hardware (Kapitel 4.1.2, ANHANG 3) ist die Sicherheit bei der Übertragung gegeben. Bei der Verwendung der PHDC-Klasse wird ebenfalls von richtlinienkonformer Sicherheit ausgegangen. Nur noch das Steuermodul für die USB-Übertragung ist demnach anfällig. Durch einen Dialog, der die doppelte Sicherheit gewährt, wird die Softwareseite abgesichert. Die Hardware reagiert erst nach dreimaliger Registrierung eines Befehls. Gleiches gilt für die Antworten der Hardware. Der beispielhafte Ablauf in Abbildung 28 verdeutlicht dies.

⁹⁹ Jeweils für die Situationen: Der Patient will mehr Medikamente und darf mehr erhalten als angeordnet *oder* der Patient will mehr Medikamente, darf aber nicht mehr erhalten als angeordnet.

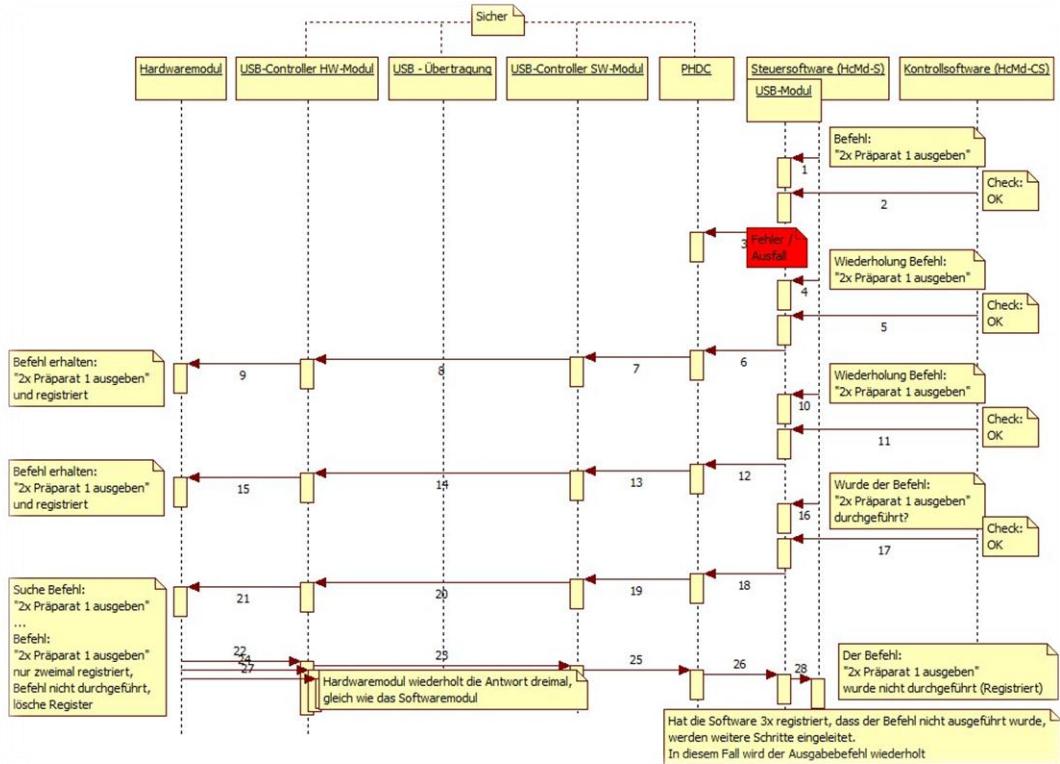


Abbildung 28: Ablauf sicherer USB-Kommunikation (selbst erstellt)

SWF004: Die Integrität der Steueroftware muss durch die Kontrollsoftware gegeben sein. Ohne *OK* der Kontrollsoftware führt die Steueroftware keinen Befehl aus. Als Befehl gelten sowohl Steuerbefehle für die Hardware sowie Aufforderungen der Steueroftware an den Benutzer. Bei jedem Start müssen sich Kontrollsoftware und Steueroftware synchronisieren, sodass sichergestellt ist, dass sie die gleichen Datensätze verwenden.

SWF005: Die Kontrollsoftware muss so einfach wie möglich gehalten werden, damit Fehler aufgrund der Komplexität ausgeschlossen werden können. Die Steueroftware führt keinen Befehl ohne Zustimmung der Kontrollsoftware aus. Bei Ausfall der Kontrollsoftware muss das Betriebssystem neugestartet werden. Tritt der Ausfall dauerhaft auf, muss ein zuständiger Techniker informiert werden.

SWF006: Bei jedem Start des Betriebssystems, müssen sich Kontrollsoftware und Steueroftware synchronisieren (Datensätze). Nach einem Absturz wird das Betriebssystem normalerweise neugestartet. Fällt das Betriebssystem ganz aus, muss ein zuständiger Techniker informiert werden.

SWF007: Vor jedem Befehl der Steueroftware muss die Systemuhr synchronisiert werden. Dies geschieht über zwei Wege. Ist das System mit dem Internet verbunden, wird die Systemuhr automatisch über die NTP-(Network Time Protocol) Technik synchronisiert. Ist das Gerät nicht mit dem Internet verbunden,

synchronisiert eine interne Funkantenne über das DCF77-Signal die Systemuhr. Das Funksystem muss in das PC-System nachgerüstet werden. Die Software ist nur auf einem vom Hersteller modifizierten PC-System, das mit dem Funksystem ausgestattet ist, zugelassen.

SWF008, SWF009: Benutzer der Hauptgruppe 2 (Patienten) müssen darauf hingewiesen werden¹⁰⁰, dass beabsichtigte Falschangaben in den Dialogen der Steuersoftware als Zweckentfremdung gelten. Die Dialoge der Software müssen defensiv designt werden.

Wissen001-004: Benutzer der Hauptgruppe 1 (Ärzte) müssen in das Gerät durch den zuständigen Techniker ausgiebig eingewiesen werden. Die Einweisung beinhaltet auch alle Schritte zur Einweisung der Benutzer der Hauptgruppe 2. Die Einweisung wird dem Hersteller schriftlich durch den Benutzer der Hauptgruppe 1 bestätigt. Benutzer der Hauptgruppe 2 werden durch Benutzer der Hauptgruppe 1 eingewiesen. Es müssen außerdem in der Gebrauchsanleitung ausgiebige Guides für den Gebrauch existieren.

GbT001-002: In der Entwicklung müssen frühestmöglich die Dialoge zwischen Benutzer und System geprüft werden - am besten durch einen frühen Prototyp der Benutzer-Produkt-Schnittstelle.

Bewertung des Restrisikos/Risiko-Nutzen-Analyse

Nach den Risikokontrollmaßnahmen würde eine Bewertung des Restrisikos und die Beurteilung neu entstandener Risiken, aufgrund der Beherrschungsmaßnahmen, stattfinden¹⁰¹. Wird das Restrisiko als akzeptabel eingestuft, wird die Entwicklung fortgeführt. Ist das Restrisiko inakzeptabel, muss eine Risiko-Nutzen-Analyse durchgeführt werden. Überwiegt der medizinische Nutzen, wird die Entwicklung fortgeführt. Andernfalls wird die Entwicklung beendet oder es müssen weitere Risikobeherrschungsmaßnahmen durchgeführt werden, damit das Risiko in den akzeptablen Bereich sinkt. An diesem Punkt wird davon ausgegangen, dass das Restrisiko als akzeptabel eingestuft wurde. Die Entwicklung des Systems kann in die nächste Phase gehen.

¹⁰⁰ Einweisung mit Nachweis/Gebrauchsanleitung

¹⁰¹ Siehe Kapitel 2.3.4, Abbildung 13, S. 33

4.3 Entwicklung nach Scrum

An dieser Stelle der Entwicklung würde ein Mitglied der Firma die Aufgabe des Product-Owners (Kapitel 3.4.2) übernehmen, das Product-Backlog zusammenstellen und die User Stories (Items)¹⁰² priorisieren. Planungselemente¹⁰³, die üblich für ein Scrum und deren Sprints sind, werden in dieser Arbeit nicht weiter verfolgt. Der ANHANG 6 zeigt einen priorisierten Product-Backlog, speziell für unser System. Die Items im Backlog sind in Form von User Stories nach Cohn dargestellt.

Nach Erstellung des Product-Backlogs sucht sich das üblicherweise Entwicklerteam in einem sogenannten Estimation-Meeting – unter Führung des ScrumMasters – die passenden Items für den ersten Sprint heraus. Die Auswahl der Items wird durch drei Faktoren bestimmt: *Aufwand*, *Priorität* und *Sprintzeit*. Der Aufwand jedes Items kann mit Hilfe sogenannter *Story Points* geschätzt werden. Jeder Story Point stellt eine feste Aufwandsgröße dar, beispielsweise entspricht ein Story Point einer Arbeitsstunde eines Teammitglieds. Anhand der Aufwandsschätzung, der festgelegten Sprintzeit und der Priorität der Items werden die Items den Sprints zugeordnet. In unserem Fall hat das Team für den ersten Sprint die Items für die Erstellung eines Simulationsprototyps, inklusive erforderlicher grundlegender Module, festgelegt. Das betrifft die User Stories 1 – 5 (ANHANG 6).

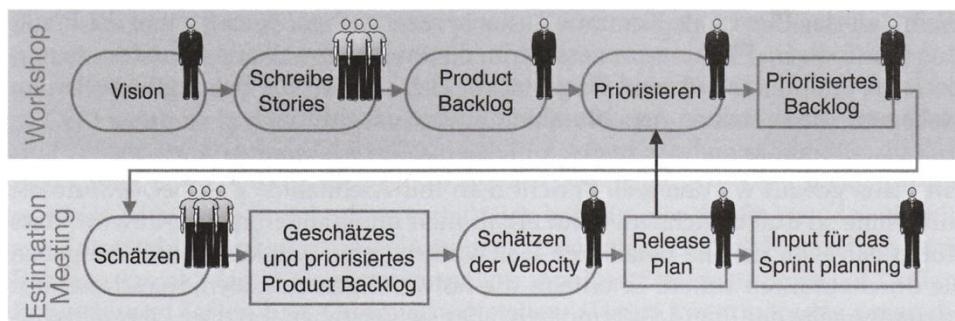


Abbildung 29: Planungsprozess im Scrum [[GLOG2008](#)]

4.3.1 Vorberichtigungen

Use Cases für das HcMd-S

Aus den vorangegangenen Spezifikationen lässt sich das folgende Use-Case-Diagramm (Abbildung 30) für die Steuersoftware HcMd-S erstellen. Die Benutzer-Hauptgruppen sind abgeleitet aus den Benutzergruppen von 4.2.2. Die detaillierte Definition der Gruppen ist in ANHANG 4 zu finden.

¹⁰² B. Gloger. „Scrum – Produkte zuverlässig und schnell entwickeln“. [[GLOG2008](#)]

¹⁰³ Beispielsweise Release-Plan/Produktvision/Daily Scrums. Siehe Kapitel 3.4.2, Abbildung 22, S. 54 oder B. Gloger. [[GLOG2008](#)], S. 177

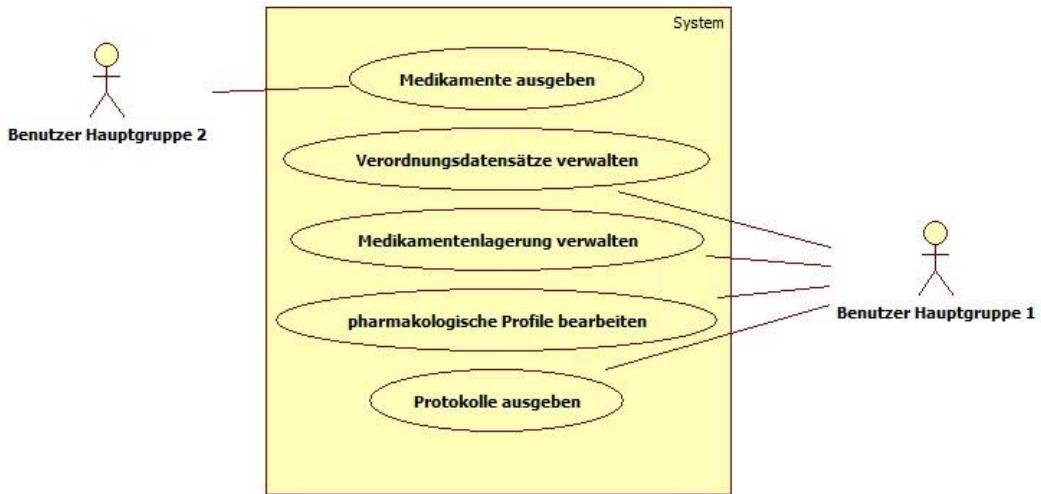


Abbildung 30: Use-Case-Diagramm HcMd-S (selbst erstellt)

Domänenmodell der Fachebene (HcMd-S)

Vor Beginn des detaillierten Designs des Systems ist es sinnvoll, bereits ein grobes Domänenmodell der Fachebene zu erstellen. Abbildung 31 zeigt das Domänenmodell für die Steuersoftware HcMd-S. Eine größere Abbildung ist in ANHANG 7 zu finden. Dieses Domänenmodell ist minimalistisch aufgebaut und zeigt lediglich die Klassenbeziehungen, deren Variablen und Typ.

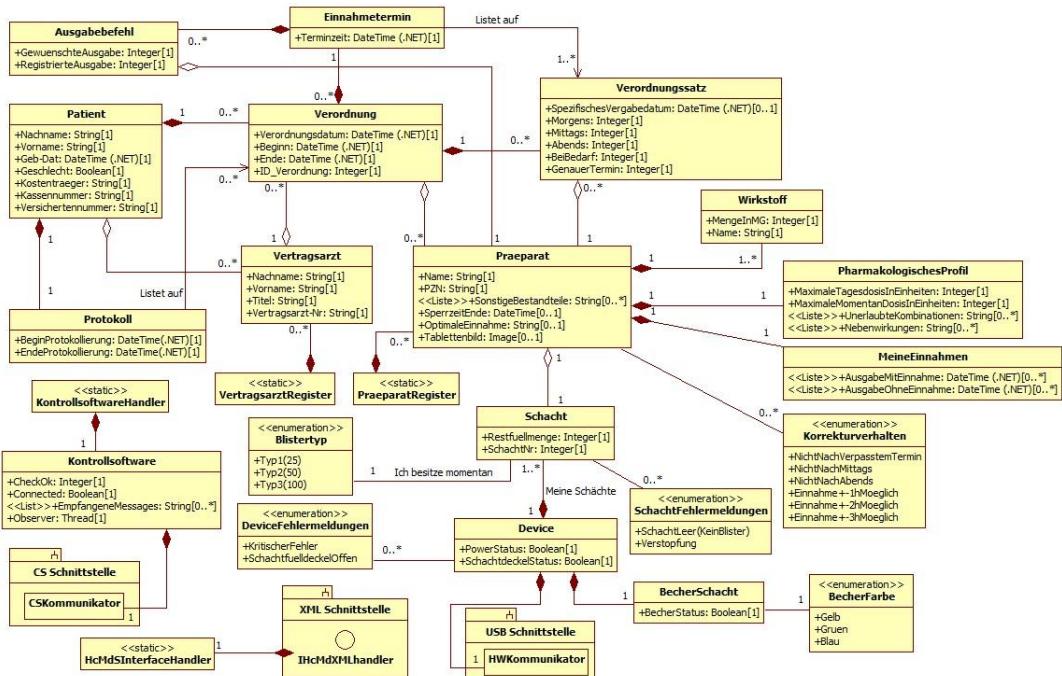


Abbildung 31: Domänenmodell der Fachebene (selbst erstellt)¹⁰⁴

¹⁰⁴ Größere Darstellung in ANHANG 7.

4.3.2 Der erste Sprint

Der folgende Abschnitt enthält die wichtigsten Punkte der jeweiligen User Stories.

4.3.2.1 User Story 1 (ANHANG 6)

Design der Architektur des Systems mit Simulationsmöglichkeit

Die erste User Story hat zur Aufgabe, einen Hardwaresimulationsmodus für die Steuersoftware HcMd-S zu erstellen. Gleichzeitig wird auch die Architektur der Steuersoftware festgelegt. Abbildung 32 zeigt die Architektur der Software mit Simulation. Der Simulationsbetrieb enthält das Simulations-GUI, welches auf die Standard-GUI-Elemente der Steuersoftware zugreift und die Hardwaresimulation.

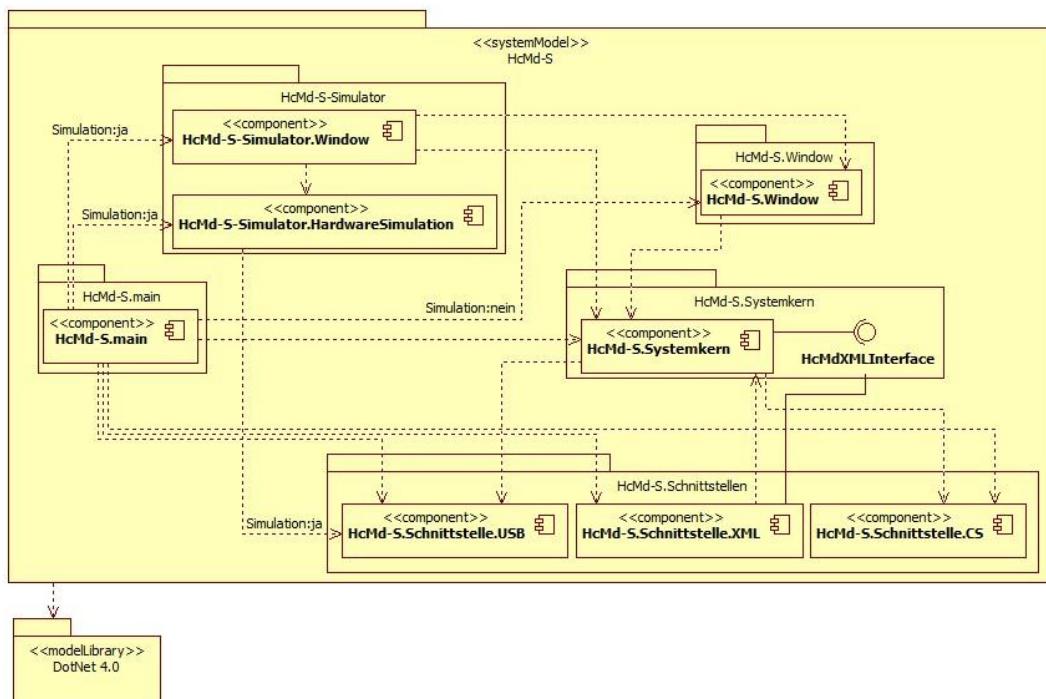


Abbildung 32: Architektur des HcMd-S mit Simulationsbetrieb (selbst erstellt)

Design der Hardwaresimulation

Die Hardwaresimulation besteht aus Simulator, SimulatedDevice und SimulatedSchacht (Abbildung 33). Für die Kommunikation zwischen Simulator und Steuersoftware wurde im Allgemeinen auf ein Observer Design Pattern¹⁰⁵ zurückgegriffen. Dafür wird zum einen auf Simulatorseite die Kommunikation über einen Thread (Für das senden zuständig), sowie Events sichergestellt.

¹⁰⁵ Software-Architektur Design Vorlage für eine Beobachterfunktion.

Der HWSimuController, welcher für die Kommunikation auf Seiten der Steuersoftware zuständig ist, arbeitet mit einem Observer Thread. Dieser Observer Thread prüft, ob etwas in den eingehenden Bus (von Simulator an Steuersoftware) geschrieben wurde. Die Überwachung mittels Observer ist gleich wie bei der realen USB-Kommunikation.

Wird von der Steuersoftware ein Befehl an die Hardware gesendet, löst der HWSimuController ein *OutgoingBusEvent* aus. Das OutgoingBusEvent wird vom Simulator durch die Funktion *myHWSimuController_OutgoingBusEvent()* aufgefangen. Der Simulator sendet die eingehenden Befehle weiter an das virtuelle Device, welches dementsprechend über den Simulatorthread antwortet.

Das GUI der Hardwaresimulation erlaubt außerdem verschiedene Einstellungen des Devices, wie beispielsweise Power On/Off oder Fehlersituationen.

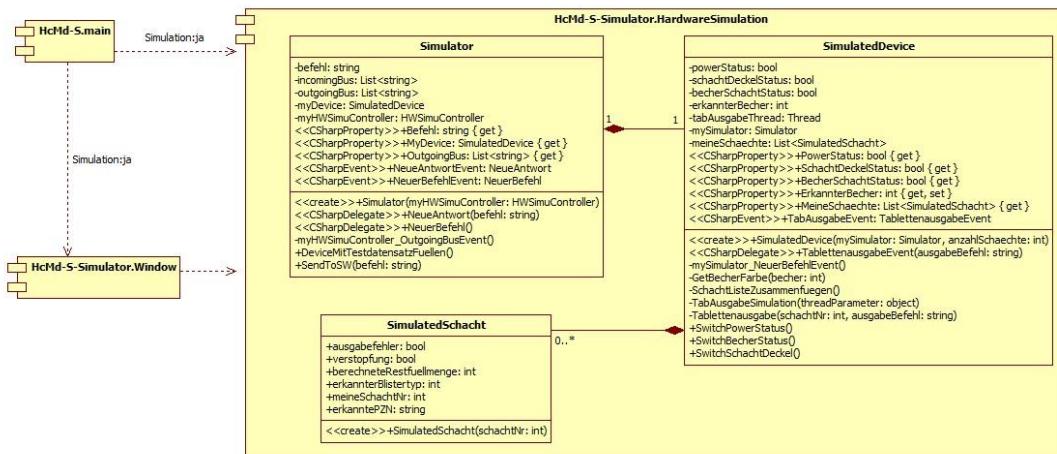


Abbildung 33: Design Hardwaresimulation (selbst erstellt)

Design des GUI im Simulationsbetrieb

Für das GUI existierten zwei Designvorschläge. Der erste Vorschlag (Abbildung 34) wurde wieder verworfen. Grund dafür war, dass die Kontrollsoftware HcMd-CS unabhängig von der Steuersoftware sein soll (eventuell sogar auf einem eigenen Controller) und deshalb über ein eigenständiges GUI verwaltet wird. Außerdem war der Bereich für die Einbindung der Steuersoftware (HcMd-S-GUI-Einbindung) zu klein und der Protokollbereich musste gesplittet werden, da er zu unübersichtlich wurde. Der zweite überarbeitete GUI-Vorschlag (Abbildung 35) wurde umgesetzt. Das fertige Ergebnis ist in Abbildung 36 zu sehen.

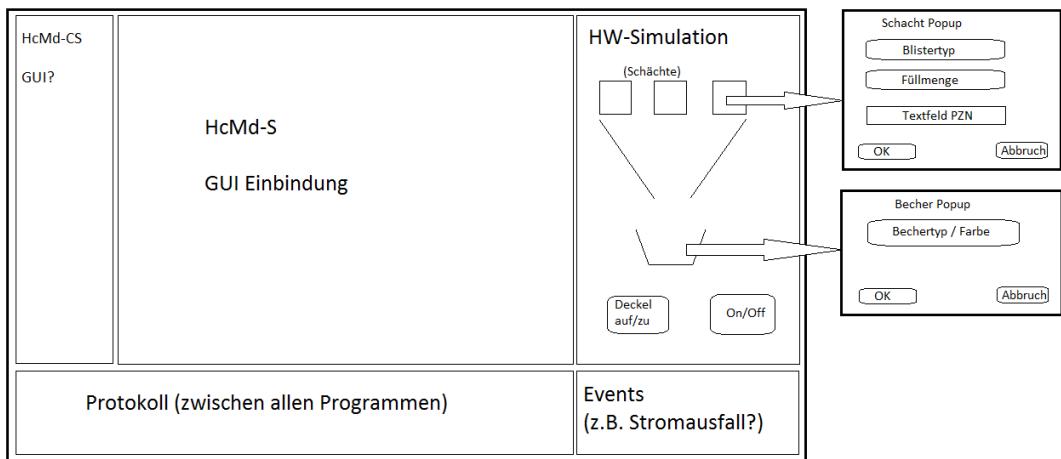


Abbildung 34: GUI-Entwurf 1 für den Simulationsbetrieb (selbst erstellt)

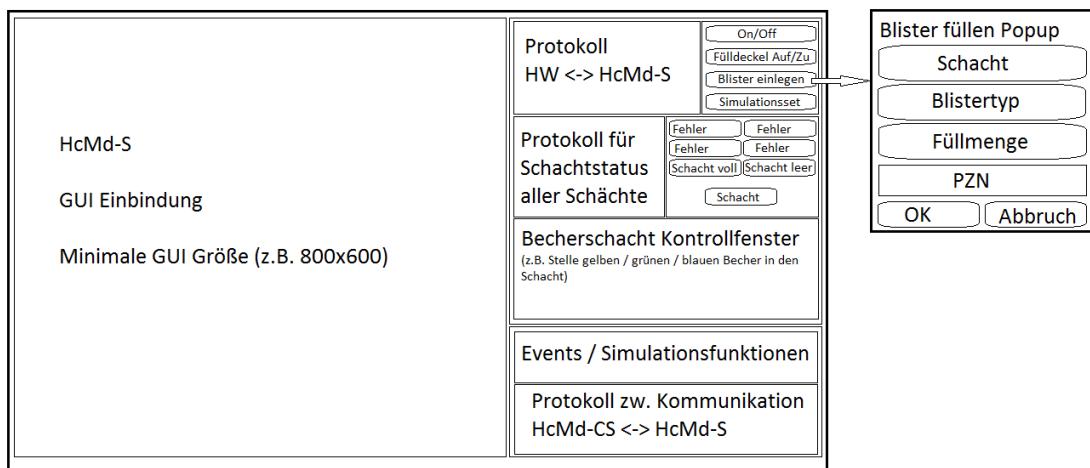


Abbildung 35: GUI-Entwurf 2 für den Simulationsbetrieb (selbst erstellt)

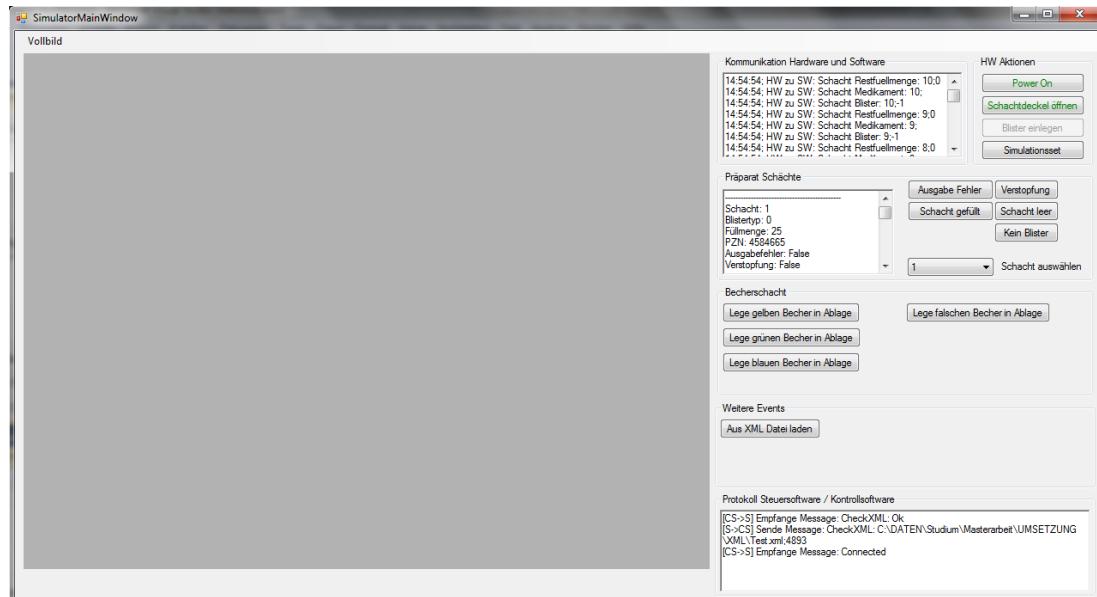


Abbildung 36: GUI für Simulationsbetrieb Final (selbst erstellt)

4.3.2.2 User Story 2 (ANHANG 6)

Ermittlung der Abläufe im GUI mit Hilfe von State-Chart-Diagrammen

Um gewisse Abläufe im GUI darzustellen, werden sogenannte State-Chart-Diagramme verwendet. Es wird für jeden Use Case¹⁰⁶ ein State-Chart-Diagramm erstellt. Die Abbildungen Abbildung 37 bis Abbildung 41 zeigen die State-Chart-Diagramme für sämtliche Use Cases. Aus den State-Chart-Diagrammen gehen die benötigten GUI-Elemente und auch Funktionen heraus, die im Oberflächendesign berücksichtigt werden müssen. Der komplexeste und gleichzeitig wichtigste Use Case ist die Medikamentenausgabe. Er umfasst sämtliche Funktionalitäten für Benutzer der Hauptgruppe 1.

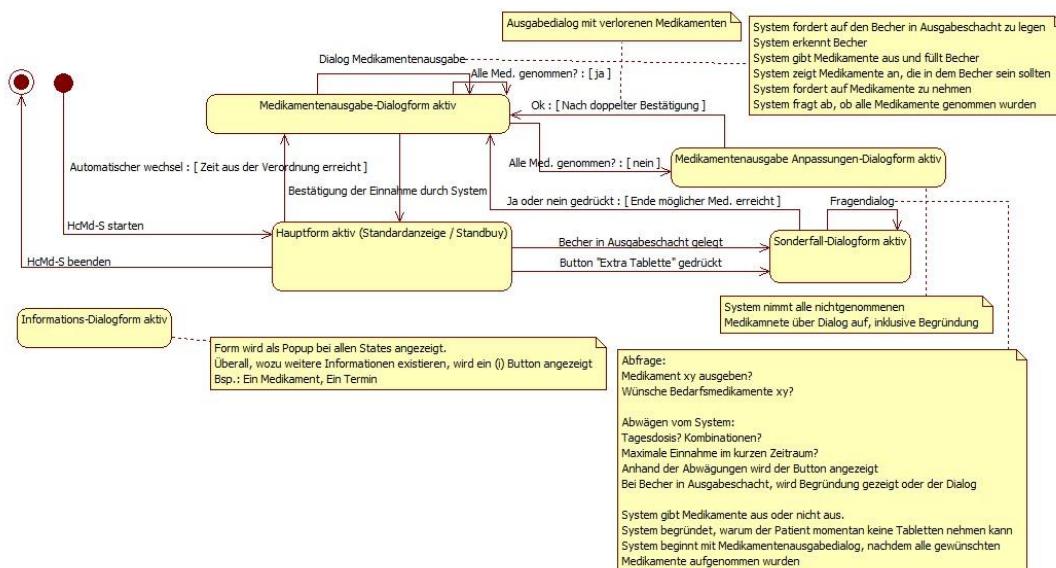


Abbildung 37: State Chart *Medikamentenausgabe* (selbst erstellt)

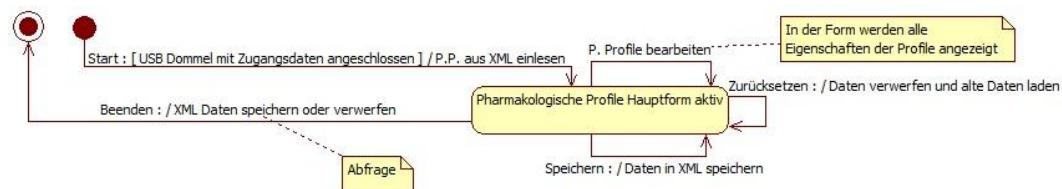


Abbildung 38: State Chart *Pharmakologische Profile Bearbeiten* (selbst erstellt)

¹⁰⁶ Siehe Kapitel 4.3.1, Abbildung 30, S. 78

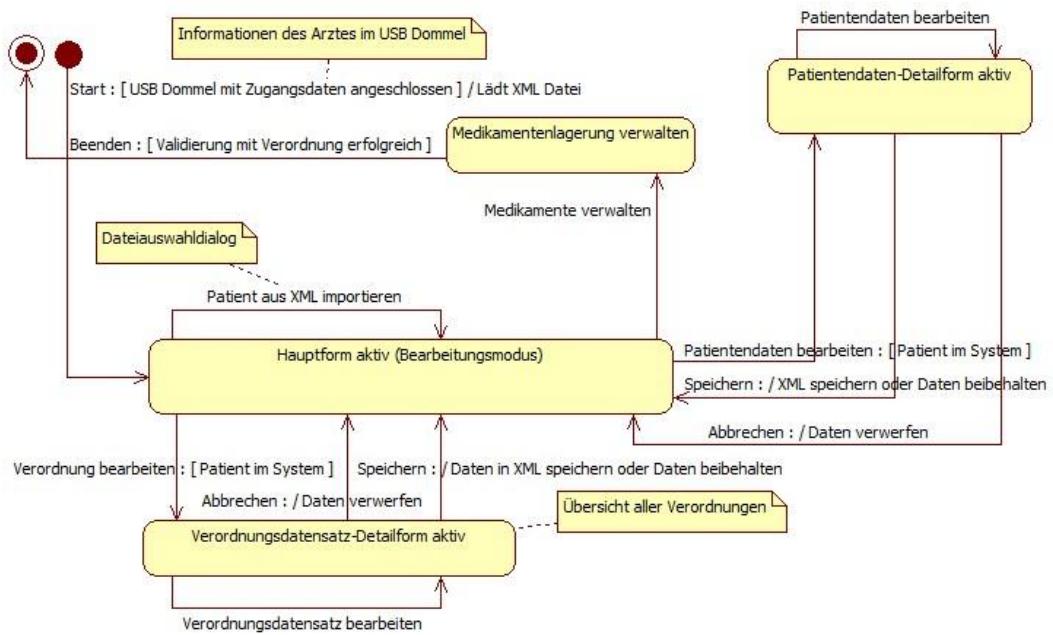


Abbildung 39: State Chart Verordnungsdatensätze verwalten (selbst erstellt)

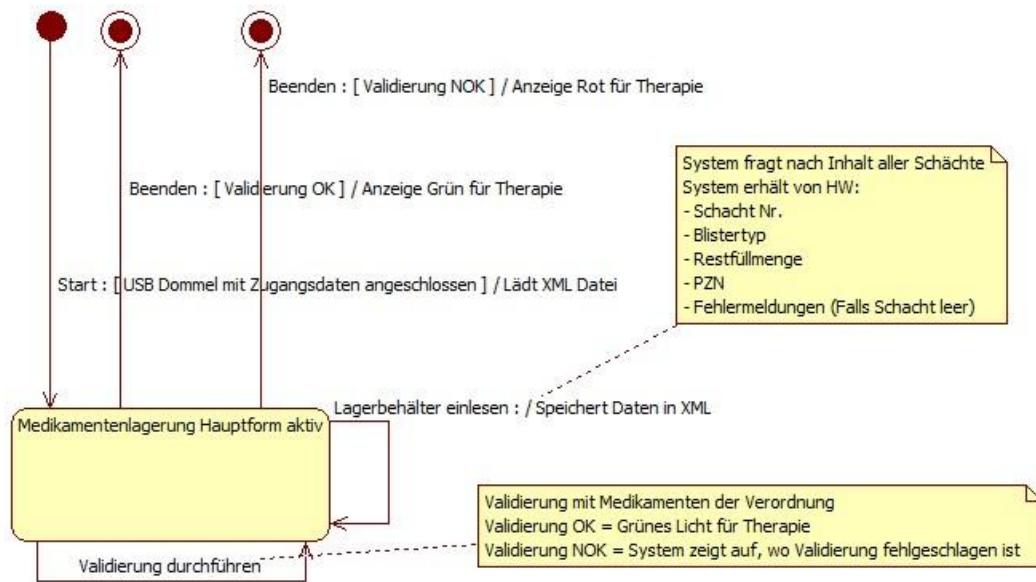


Abbildung 40: State Chart Medikamentenlagerung verwalten (selbst erstellt)

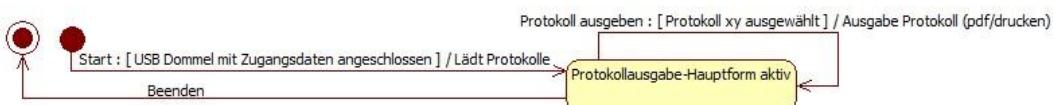


Abbildung 41: State Chart Protokolle ausgeben (selbst erstellt)

Design des GUI der Steuersoftware HcMd-S

In dieser Arbeit wurde der Use Case für die Medikamentenausgabe weiter verfolgt. Dieser Use Case bildet den wichtigsten Punkt der Software. Aus Zeitgründen wurde das Design für die Use Cases für Benutzer der Hauptgruppe 2 nicht weiter behandelt. Abbildung 42 zeigt die erste Idee für den prinzipiellen Aufbau des Medikamentenausgabedialogs. Dieser Entwurf wurde wieder verworfen, da er bei der vorgesehenen Benutzergruppe für zu viel Verwirrung sorgen würde. Der Benutzer muss zu jeder Zeit wissen, was er machen muss. Das GUI soll den Benutzer am besten wie eine *freundliche Schwester* durch den gesamten Dialog führen.

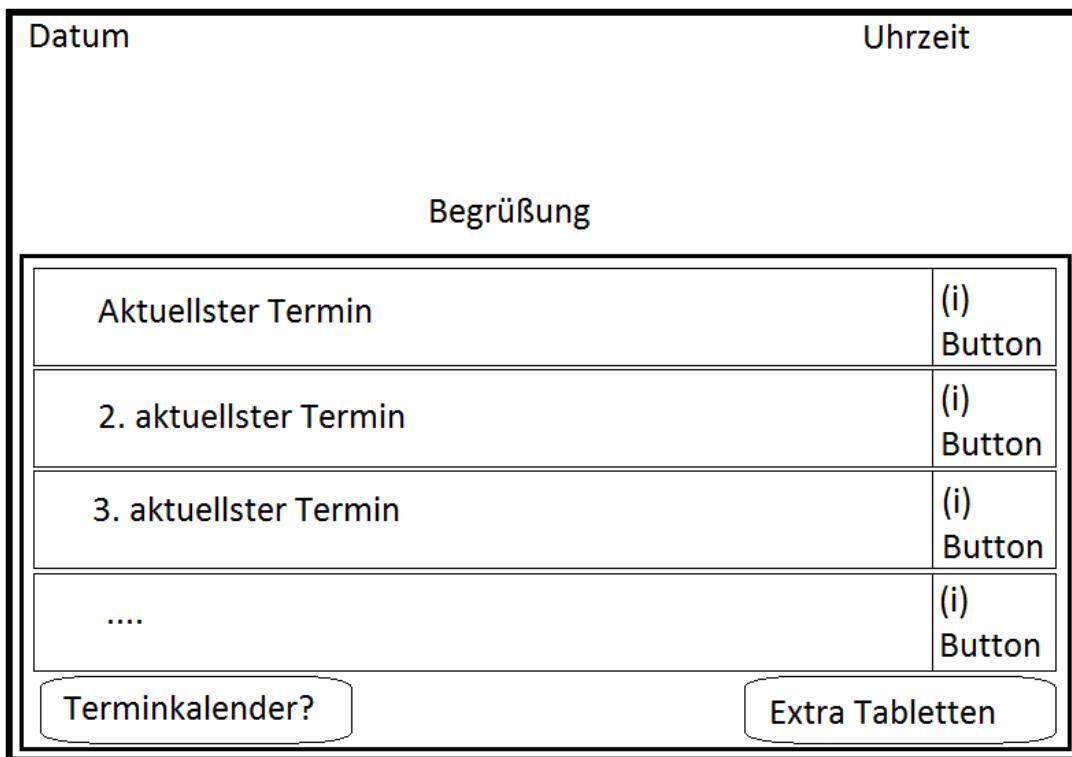


Abbildung 42: Erste GUI-Idee Standby-Monitor (selbst erstellt)

Der zweite GUI-Entwurf ist wesentlich einfacher aufgebaut und passt besser in die gewünschte *schwesterliche* Rolle. Abbildung 43 und Abbildung 44 zeigen den Standby- und Sonderfall-Monitor für den Medikamentenausgabedialog in der ersten finalen Fassung.

Aktuelles Datum: Montag, 22. August 2011

Aktuelle Uhrzeit: 15:00:27

Guten Tag Herr Philipp

Ihr nächster Einnahmetermin ist am 22.08.2011 um 17:00 Uhr.

Wünschen Sie zusätzliche Informationen zum nächsten Termin? Drücken Sie einfach auf den Button mit dem "i".

Ich wünsche Ihnen eine angenehme Zeit.

Sie dürfen zu jeder Zeit Bedarfsmedikamente zu sich nehmen. Drücken Sie einfach auf den Button "Extra Tablette".

Nehmen Sie die Tabletten des kommenden Termins nach der Mahlzeit.

X

Extra
Tablette

i

Abbildung 43: Standby-Monitor, erste finale Version (selbst erstellt)

Aktuelles Datum: Montag, 22. August 2011

Aktuelle Uhrzeit: 15:34:35

Guten Tag Herr Philipp

Möchten Sie dieses Medikament nehmen? Dann drücken Sie bitte auf "Ja". Wenn Sie das Medikament nicht nehmen möchten, drücken Sie bitte auf "Nein".

Coffeignum N 0,2g



Detailinformationen zum Medikament? Bitte "i" drücken.

Nein

i

Ja

Abbildung 44: Sonderfall-Monitor, erste finale Version (selbst erstellt)

Einhaltung der Usability-Heuristiken

Bei dem Design des GUIs wurden außerdem sogenannte Usability-Heuristiken mit beachtet. Die Evaluation mittels Heuristiken zählt laut Römeling¹⁰⁷ zu den Usability-Inspections-Methoden. Usability Inspections ergänzen die klassischen Usability-Tests und können mit wenig Kosten und bereits in der frühen Entwicklungsphase durchgeführt werden. Als Grundlage für die einzuhaltenden Heuristiken dienen die von Prof. Dr. K. Andrews überarbeitete Checkliste von Nielsen und Molich¹⁰⁸, sowie die von der AARP¹⁰⁹ veröffentlichte Liste von Heuristiken für Benutzer über 50 Jahre¹¹⁰. Im Folgenden sind einige Beispiele für die Einhaltung aufgelistet:

- Es gibt keine Tabellen oder andere komplexe Elemente.
- Es wurden 5 verschiedene Farben verwendet (hellblau, blau, rot, grün, gelb). Dabei setzen sich die einzelnen Elemente sichtbar vom Hintergrund ab (Objekte, Aktionen und Optionen klar sichtbar machen).
- Es wurde darauf geachtet, dass jedes Element, wenn es in verschiedenen Monitoren vorkommt, immer an der gleichen Stelle mit der gleichen Konfiguration zu finden ist (Konsistenz). Lediglich der Info Button hat zwei verschiedene Standorte, welche abhängig sind von der gewünschten Information (Medikamenten-/Termindetails).
- Es gibt keine Scrollbars.
- Die anklickbaren Elemente sind groß und können leicht anvisiert werden.
- Jede Aktion an Hardware oder Software erzeugt ein Feedback. Entweder durch eine Aktion der Hardware oder eine visuelle Veränderung der Oberfläche.
- Die Fehlermeldungen sind nach A09 von Prof. Dr. K. Andrews Checkliste aufgebaut. Das heißt:
 - o Einfache Sprache (keine Codes)
 - o Präzise (Fehler genau beschreibend)
 - o Defensiv (Der Benutzer hat keine Schuld)
 - o Konstruktiv (Lösungsweg aufzeigen)Auf Mehrstufigkeit der Fehlermeldungen wurde verzichtet, da laut AARP¹⁰⁹-Liste die Informationen für ältere Benutzer flach gehalten werden sollen.
- Die Aktionen der Benutzer sind auf das Nötigste beschränkt (Fehlervermeidung).
- Soweit wie möglich wurden umkehrbare Aktionen in die Oberfläche mit eingebaut.

¹⁰⁷ N. Römeling. „Usability im World Wide Web. Redesign eines Webauftritts mit Hilfe von Usability Testing“. [\[RÖME2002\]](#)

¹⁰⁸ Ao.Univ.-Prof. Dr. K. Andrews. „Andrews Allgemeine Usability-Heuristiken 2006“. [\[IQ18\]](#)

¹⁰⁹ Association for the Advancement of Retired Persons

¹¹⁰ Jens. O. Meiert. „Usability-Heuristiken“. [\[MEIE2005\]](#)

4.3.2.3 User Story 3 (ANHANG 6)

Design des USB-Moduls

Das USB-Modul ist für zwei verschiedene Übertragungsarten konstruiert (Abbildung 45). Der Simulationsbetrieb (HWSimuController) und der Normalbetrieb über USB (HWControllerPHDC). Beide Klassen erben vom HWController, der grundlegende Funktionen und Variablen deklariert.

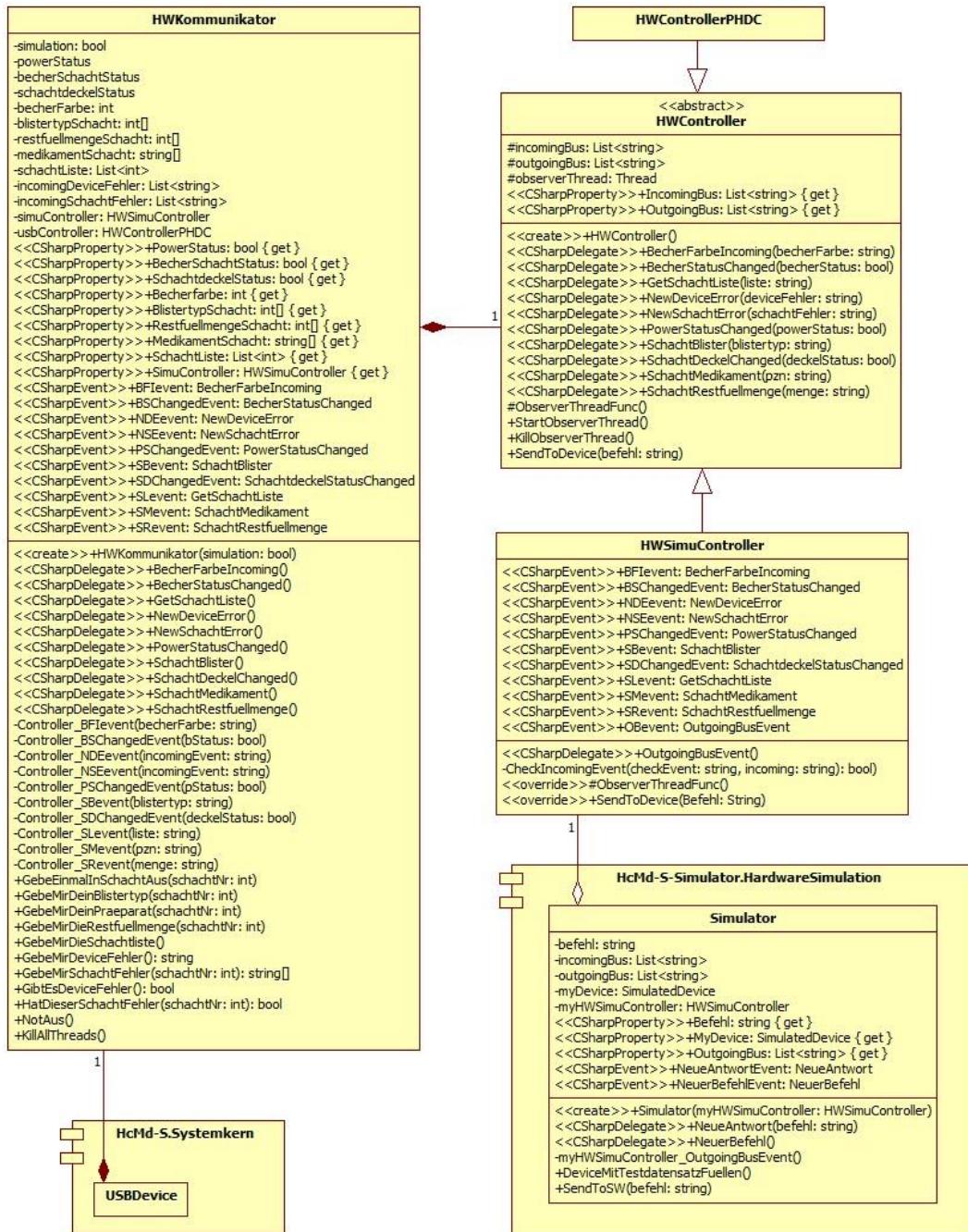


Abbildung 45: Klassendiagramm des USB-Moduls (selbst erstellt)

Der Simulationscontroller ist, wie bereits in 4.3.2.1 erwähnt, mit dem Simulator verbunden. Die Controller sind lediglich für die Übertragung und den Empfang von Befehlen zuständig. Der HW-Kommunikator interpretiert die übertragenen Befehle aus den Controllern und wirft ein auf den erkannten Befehl angepasstes Event aus. Die USB-Device-Klasse reagiert auf das Event und ändert seinen Status. Beispielsweise Power On/Off.

Dreifache Übertragung der Befehle

Die Risikokontrollmaßnahme *SWF003*¹¹¹ hat als Anforderung eine dreifache Übertragung der Befehle gefordert.

```
public void GebeMirDeinPraeparat(int SchachtNr)
{
    // Dreifache Ausgabe
    DateTime timestamp = DateTime.Now;
    simuController.SendToDevice("Befehl Praeparat: "+SchachtNr+";"+timestamp);
    simuController.SendToDevice("Befehl Praeparat: "+SchachtNr+";"+timestamp);
    simuController.SendToDevice("Befehl Praeparat: "+SchachtNr+";"+timestamp);
}
```

Gleich, wie die Risikokontrollmaßnahme es verlangt, sendet die Hardware, beziehungsweise der Kommunikator auf Softwareseite, den gewünschten Befehl dreimal. Der Observer und Eventhandler auf der jeweiligen Empfängerseite überprüfen alle eingehenden Befehle und senden den Befehl erst weiter, wenn drei gleiche Befehle mit dem gleichen Zeitstempel den Controller erreicht haben.

¹¹¹ Siehe Kapitel 4.2.5, Abbildung 28, S. 75

4.3.2.4 User Story 4 (ANHANG 6)

Design des XML-Moduls für den Use Case Medikamentenausgabe

Das XML-Modul hat im Use Case Medikamentenausgabe zwei wesentliche Aufgaben. Erstens, Auslesen und Interpretieren der XML-Datei und zweitens, Speichern der protokollierten Ausgaben in der XML-Datei. Diese Aufgaben erfüllt die Klasse *HcMdMedAusgabexmlHandler* (Abbildung 46). Für die restlichen Use Cases ist die Klasse *HcMdMedBearbeitenxmlHandler* zuständig. Das XML-Modul nutzt, anders als die anderen Schnittstellen, den Systemkern, damit die XML-Schnittstelle die ausgelesenen Informationen direkt in systemkerntypische Klassen zurückgeben kann. Der Systemkern nutzt das Modul über Interfaces, die in einer Container-Klasse im Systemkern selbst abgelegt sind. Beim XML-Modul ist zu beachten, dass jede Änderung der XML-Schema-Datei eine Änderung im Modul zur Folge hat. Das Modul in der mitgelieferten Software nutzt die folgende Schema-Datei.

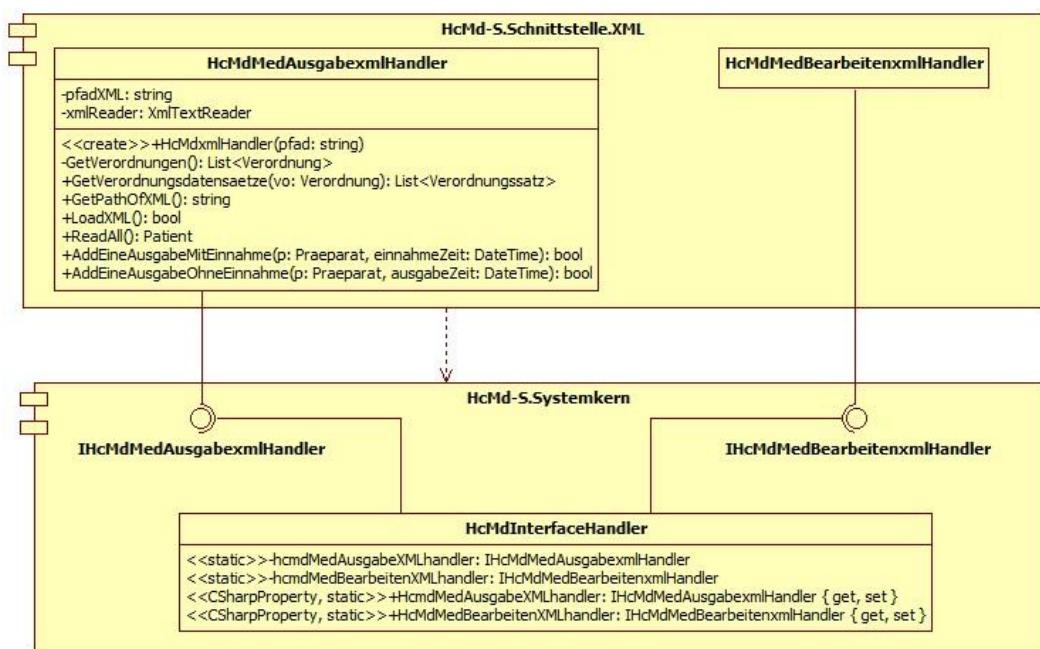


Abbildung 46: Klassendiagramm des XML-Moduls (selbst erstellt)

Design der XSD-Schema-Datei

Die jeweils genutzte XML-Datei der Software enthält alle patienten- und verordnungsbezogenen Daten für einen Therapiezeitraum. Die Abbildungen Abbildung 47 bis Abbildung 50 zeigen Detailansichten der einzelnen Objekte in der dazugehörigen XSD-Schema-Datei. Anhang 8 zeigt eine komplette Übersicht der XSD-Schema-Datei.

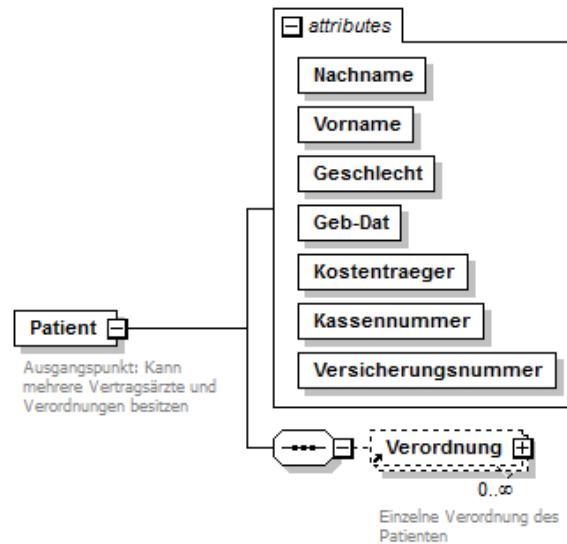


Abbildung 47: Auszug XSD-Schema Patient (selbst erstellt)

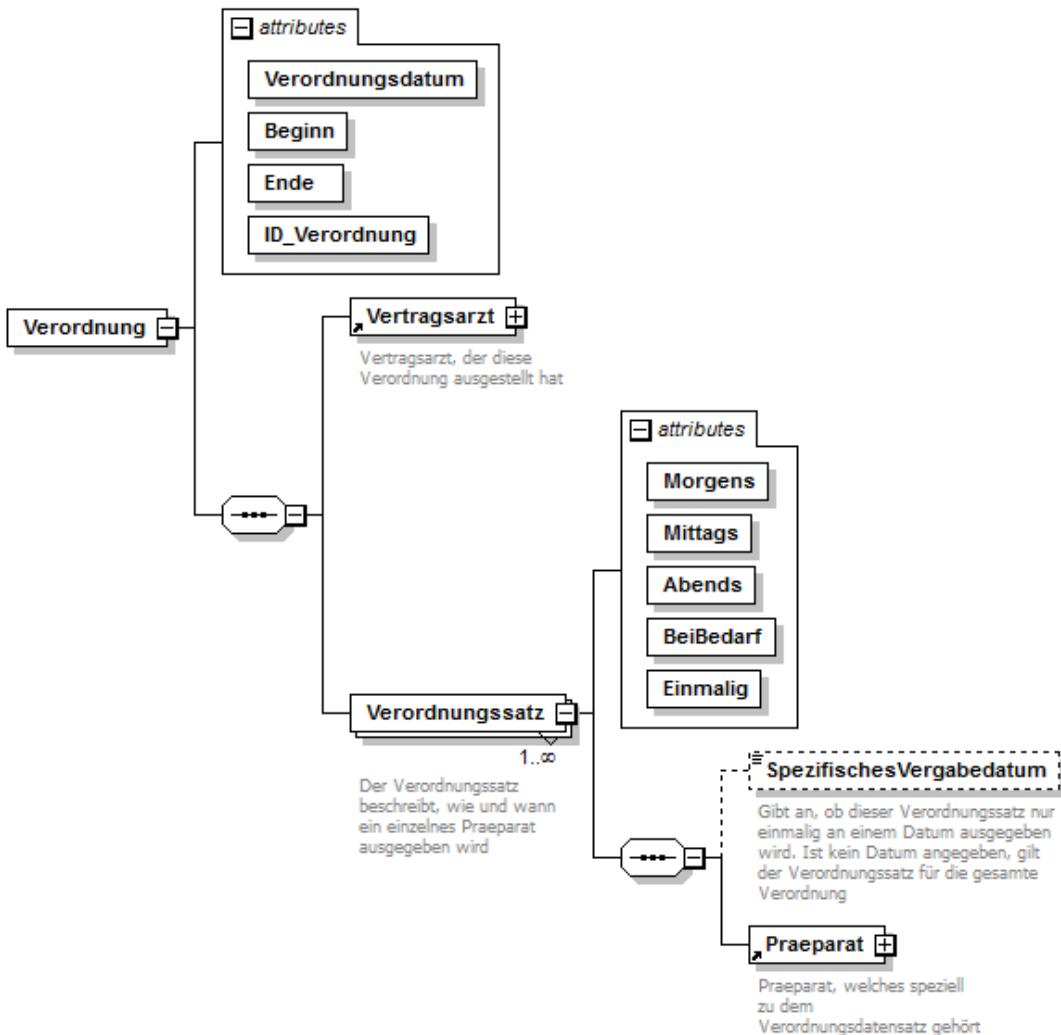
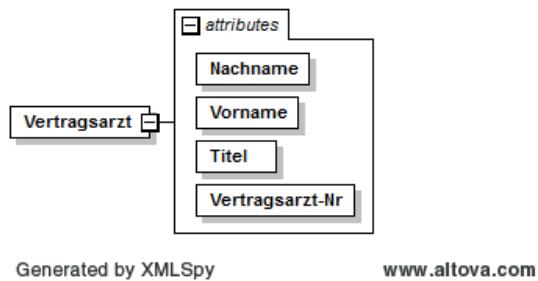
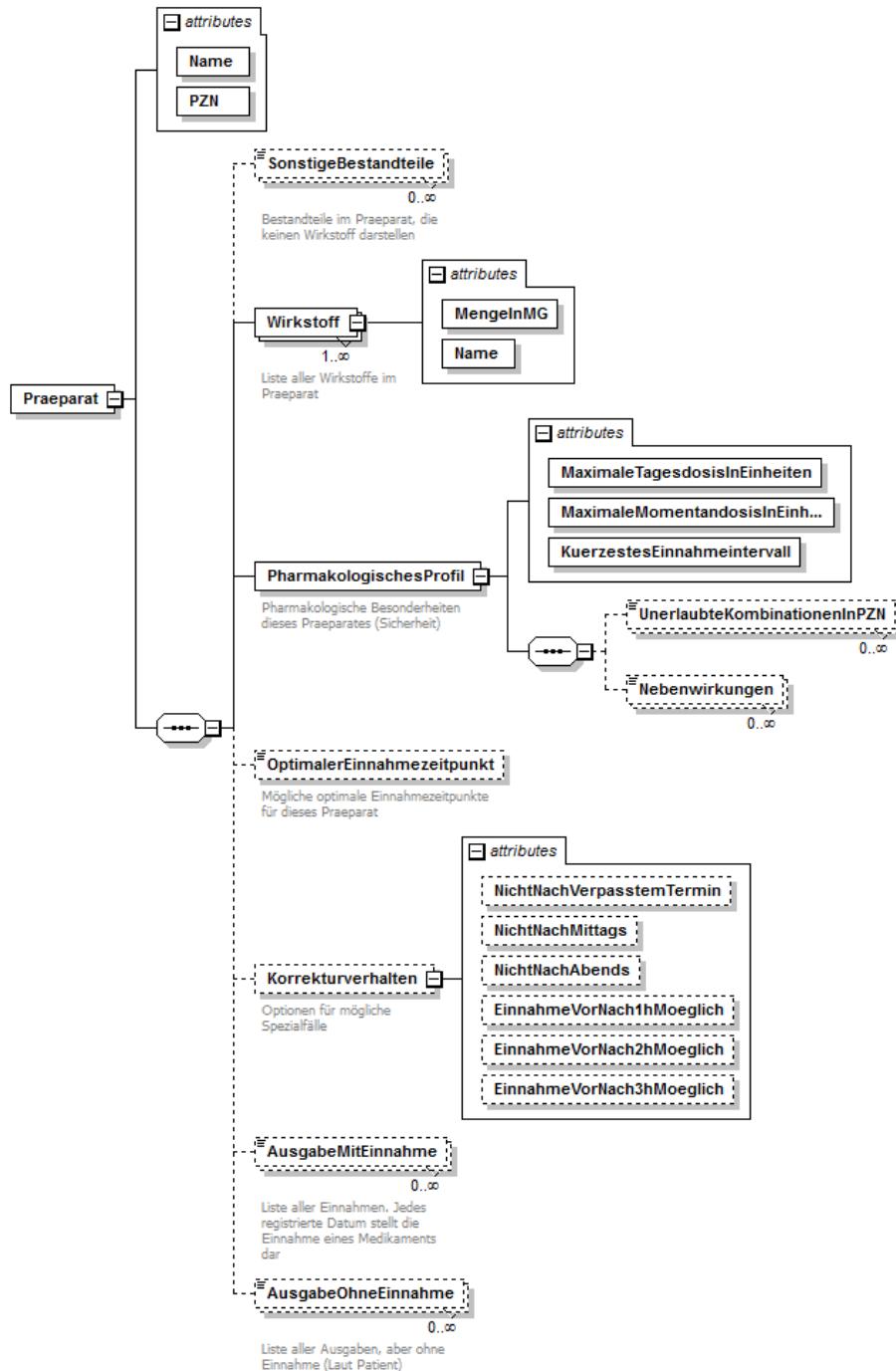


Abbildung 48: Auszug XSD-Schema Verordnung (selbst erstellt)



Generated by XMLSpy www.altova.com

Abbildung 49: Auszug XSD-Schema Vertragsarzt (selbst erstellt)



Generated by XMLSpy

www.altova.com

Abbildung 50: Auszug XSD-Schema Präparat (selbst erstellt)

4.3.2.5 User Story 5 (ANHANG 6)

Design der Kontrollsoftware (HcMd-CS)

Auf Basis der Architekturspezifikation¹¹² soll die Kontrollsoftware die USB- und XML-Schnittstelle überwachen. Erste Ideen hatten vorgesehen, dass die Kontrollsoftware außerdem eine Protokollfunktion sowie ein GUI zum Einlesen von pharmakologischen Profilen besitzen soll. Nach *SWF005*¹¹³ darf die Kontrollsoftware nicht zu komplex werden, da sie die Sicherheit des Systems garantieren muss. Aus diesem Grund wurde die Kontrollsoftware rein auf den Kontrollmechanismus reduziert. Die beiden angedachten Funktionen wurden in die Steuerosoftware verlagert. In der aktuellen Spezifikation ist die Kontrollsoftware auf dem gleichen System wie die Steuerosoftware angelegt.

Die Risiko-Analyse hat gezeigt, dass das Betriebssystem immer noch einen hohen Risikofaktor darstellt (*SWF006*¹¹³). Eine weitere Risikokontrollmaßnahme könnte deshalb die Auslagerung der Kontrollsoftware auf einen externen Controller sein. Durch die verringerte Komplexität der Kontrollsoftware ist dies jetzt möglich geworden.

Für die Übertragung zwischen Steuerosoftware und Kontrollsoftware wurden sogenannte *Named Pipes* verwendet. Named Pipes sind geschützte Datenströme zwischen Server und Client. Der Vorteil von Named Pipes besteht darin, dass auch eine Kommunikation über Netzwerk möglich ist. Somit kann die auch Kontrollsoftware auf einem externen System arbeiten, ohne dass eine komplizierte Umstellung getätigt werden muss.

Für die Überprüfung der XML-Schnittstelle werden von der Steuerosoftware die Bytegröße sowie der absolute Pfad der Datei an die Kontrollsoftware gesendet. Die Kontrollsoftware überprüft die beiden Daten und senden ein *OK* zurück, wenn sie exakt die gleichen Daten besitzt. Die Synchronisation über die XML-Schnittstelle ist auch gleichzeitig die angesprochene Synchronisation beider Softwareeinheiten. Die Kontrollsoftware und Steuerosoftware besitzen exakt die gleiche XML-Schnittstelle und den gleichen Systemkern. Die Kontrollsoftware liest somit die Daten aus der XML-Datei in seinen Systemkern.

Für die Überprüfung der USB-Schnittstelle, wird von der Steuerosoftware die PZN des auszugebenden Präparats an die Kontrollsoftware übertragen. Die Kontrollsoftware überprüft mit Hilfe der eingelesenen XML-Daten, ob das Medikament mit der angegebenen PZN ausgegeben werden darf. Erst wenn die Steuerosoftware eine

¹¹² Siehe Kapitel 4.2.1, S. 67

¹¹³ Siehe Kapitel 4.2.5, S. 75

positive Bestätigung der Kontrollsoftware erhalten hat, fährt die Steuerosoftware mit der Ausgabe fort.

Abbildung 51 zeigt die Schichten-Architektur der Kontrollsoftware. Im Windowsbetrieb besitzt die Kontrollsoftware ein minimalistisches GUI, welches lediglich die Kommunikation zwischen Steuerosoftware und Kontrollsoftware anzeigt.

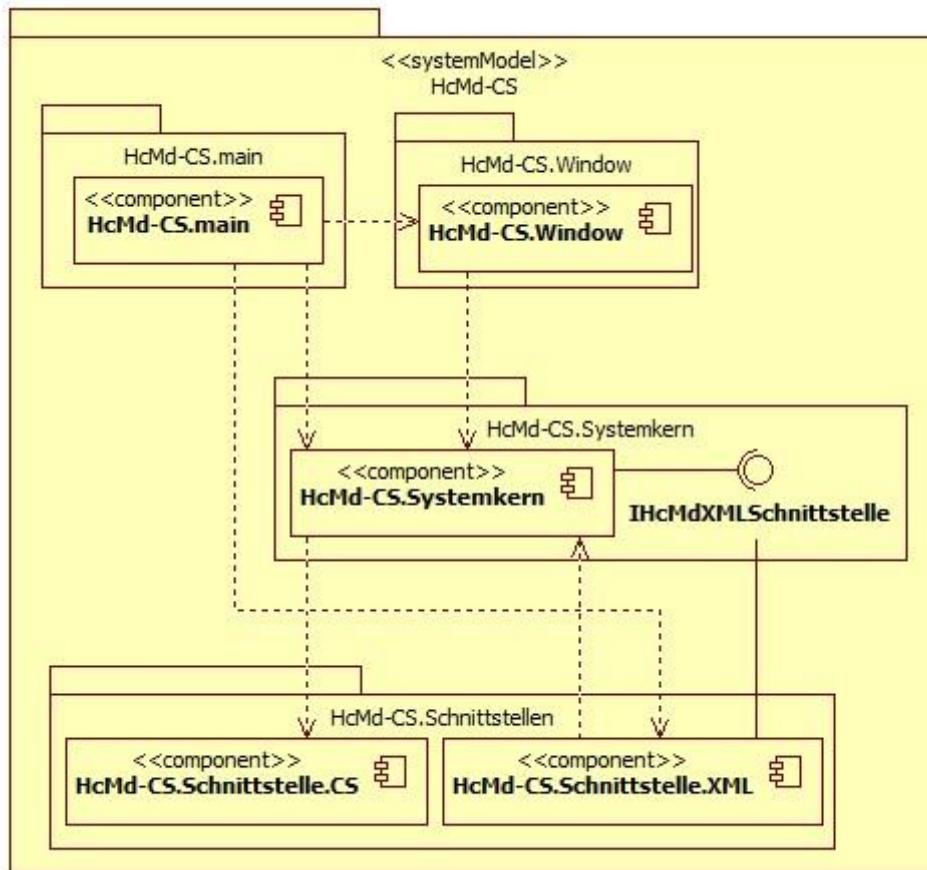


Abbildung 51: Architektur HcMd-CS (selbst erstellt)

Schnittstelle der Kontrollsoftware in der Steuerosoftware (HcMd-S)

Die Kontrollsoftware Klasse im Systemkern implementiert den *CSKommunikator* und das Event *OnNewMessage* (Abbildung 52). Die Steuerosoftware sendet die beschriebenen Anfragen über die Funktion *SendErrorMessageToCS*. Antworten der Kontrollsoftware werden wieder über ein Observer Pattern an die Kontrollsoftware Klasse weitergeleitet. Die Kontrollsoftware Klasse wartet, bis die Kontrollsoftware auf die letzte Anfrage geantwortet hat. Erst nach einer verifizierten Bestätigung, sendet die Kontrollsoftware Klasse ein *Ok* zurück und das System fährt fort. Auch bei der Kommunikation zwischen Kontrollsoftware und Steuerosoftware gilt das Prinzip der dreifachen Übertragung der Befehle mit Zeitstempel.

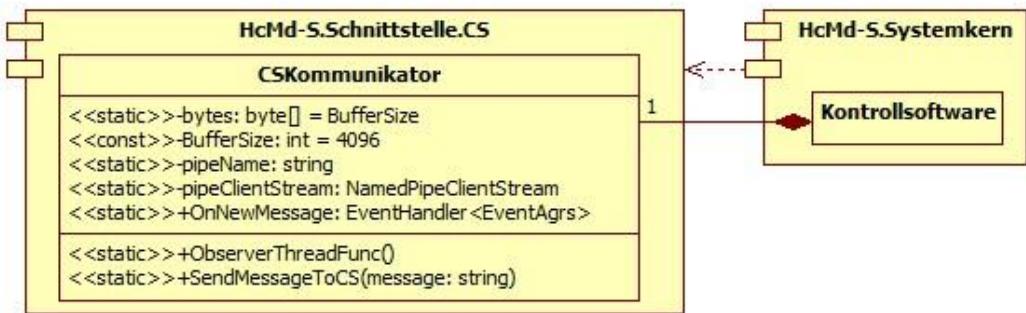


Abbildung 52: Klassenmodell Kontrollsoftware Schnittstelle (selbst erstellt)

4.3.3 Nachbearbeitung des ersten Sprints

4.3.3.1 Verifikation der Gebrauchstauglichkeit

Der Cognitive Walkthrough zählt ebenfalls zu den Usability-Inspection-Methoden¹¹⁴ und ist eine vorgeschlagene Methode der DIN EN 62366¹¹⁵ für die Verifikation der Usability. Vorteil des Cognitive Walkthrough ist, dass dieser ohne großen Aufwand (2 bis 3 Gutachter) und mit bereits guten Ergebnissen durchgeführt werden kann. Der Unterschied zu den anderen Techniken ist, dass der Cognitive Walkthrough sich weniger auf das Interface, dafür aber mehr auf die mentalen Prozesse eines hypothetischen Nutzers konzentriert. Der Cognitive Walkthrough wird in 4 Schritte unterteilt:

- Definition des Inputs
- Untersuchung der Handlungssequenzen
- Protokollierung kritischer Informationen
- Revision des Interfaces

Definition des Inputs

Die Definition des Inputs beinhaltet vier Vorbereitungen und Vorüberlegungen.

Benutzercharakteristiken: Ausführliche Benutzerprofile wurden bereits in der ersten Entwicklungsphase erstellt¹¹⁶. Diese Profile beschreiben die Charakteristiken der Benutzer und stellen die benötigten Benutzercharakteristiken für die folgenden Cognitive Walkthroughs bereit.

¹¹⁴ N. Römering. [\[RÖME2002\]](#)

¹¹⁵ Siehe Kapitel 3.1, S. 39ff

¹¹⁶ Siehe Kapitel 4.2.2, S. 69/[ANHANG](#) 4

Beispielaufgaben (Benutzer-Szenarien): Zu jedem Cognitive Walkthrough zählt ein Benutzer-Szenario. Die verwendeten Szenarien für die folgenden Cognitive Walkthroughs sind in ANHANG 9 abgelegt.

Handlungssequenzen: Die benötigten optimalen Handlungssequenzen zu den Benutzer-Szenarien sind ebenfalls in ANHANG 9 zu finden.

Interface: Für die folgenden Cognitive Walkthroughs wird das User Interface des ersten Sprints verwendet.

Untersuchung der Handlungssequenzen

Die Untersuchung der Gutachter¹¹⁷ hat gezeigt, dass der Standby-, Medikamentenausgabe-, Sonderfall- und Einnahmekontrolle-Dialog noch Defizite in der Benutzerführung aufweist. Der Informationsdialog hat als einziger Dialog im Cognitive Walkthrough gut abgeschnitten. Wichtige hervorzuhebende Mängel sind die fehlende Ermutigung, das fehlende Feedback bei Aktionen des Benutzers und teilweise noch mit Informationen überfüllte Monitore (Beispielsweise der Standby-Monitor).

Revision des Interfaces

Änderungen am Standby-Monitor:

Im Rahmen der Revision wurden die Informationen auf dem Standby-Monitor weiter minimiert und in Popups verlegt. Auf den ersten Blick fallen die vorgenommenen Änderungen minimal aus, haben aber in Bezug auf den Gesamteindruck sehr viel Einfluss.

Um das Problem des fehlenden Feedbacks bei einem erreichten Einnahmetermin zu lösen¹¹⁸, erinnert nun ein Popup den Benutzer. Der überarbeitete Standby-Monitor ist in Abbildung 53 zu sehen - zum Vergleich der alte Standby-Monitor in Abbildung 43 auf Seite 85.

¹¹⁷ Protokolle der Gutachter zu den Szenarien sind in ANHANG 9 zu finden.

¹¹⁸ ANHANG 9, Szenario 1, Handlung 1

Guten Tag Herr Philipp

Ihr nächster Einnahmetermin ist am 08.09.2011 um 17:00 Uhr.

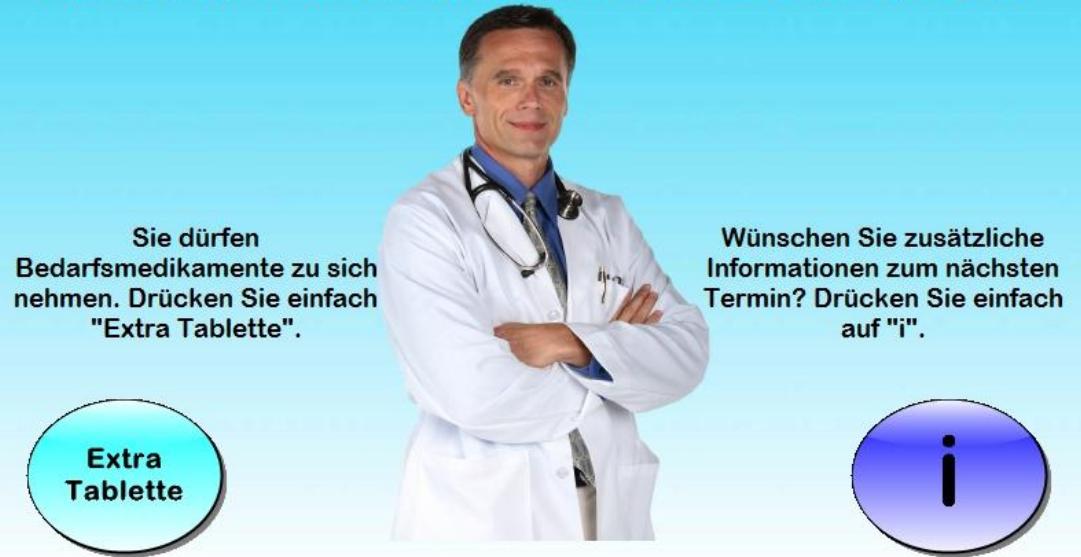


Abbildung 53: Standby-Monitor (überarbeitet) (selbst erstellt)

Änderungen am Medikamentenausgabe-Dialog:

Der Medikamentenausgabe-Dialog wurde gleichermaßen schlanker gestaltet. Hinzugekommen sind weitere Dialoge, die den Benutzer weiter im Gesamtdialog der Medikamentenausgabe unterstützen. Diese Dialoge sind über Popups gestaltet und enthalten Fragestellungen, Anweisungen und Feedback zu Aktionen des Benutzers. Besonders hervorzuheben ist das Feedback bei gewünschten Handlungen, wie beispielsweise das korrekte Einnehmen aller ausgegebenen Medikamente des Termins (Abbildung 54).

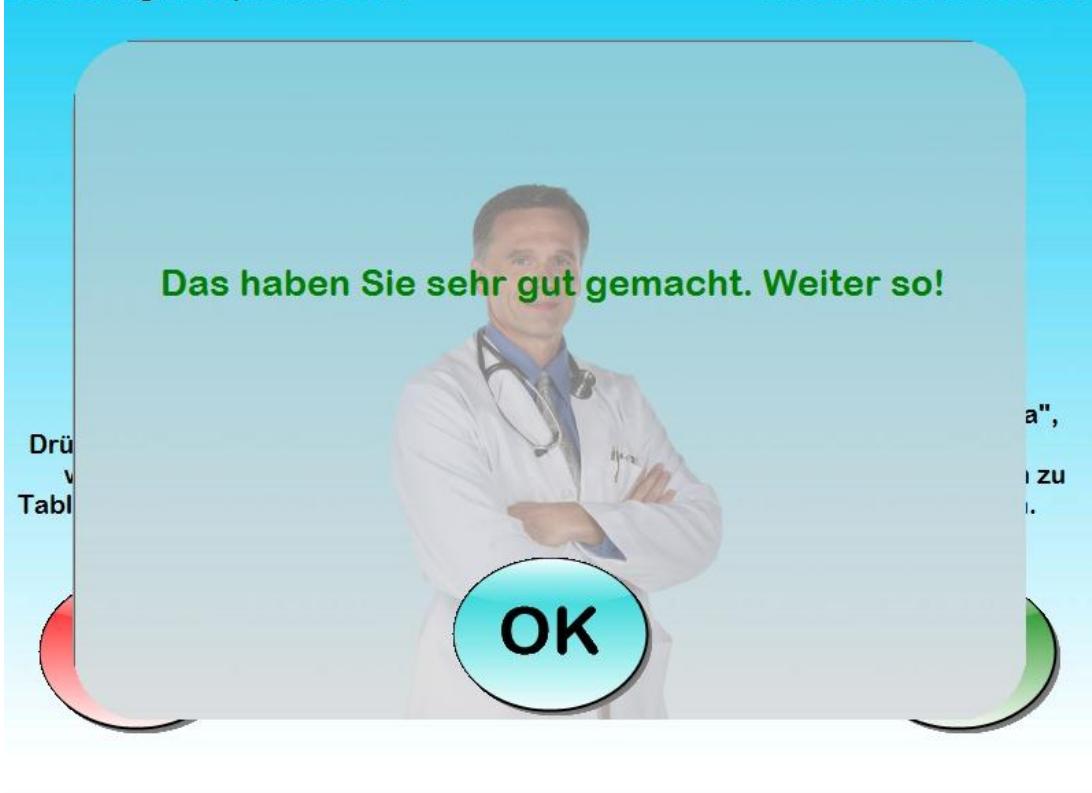


Abbildung 54: Medikamentenausgabe-Monitor Popup-Beispiel (selbst erstellt)

Änderungen am Einnahmekontrolle-Dialog und Sonderfall-Dialog:

Im Einnahmekontrolle- und Sonderfall-Dialog wurden die gleichen Änderungen, ähnlich wie in den vorherigen Dialogen, (schlankere Gestaltung und Auslagerung in Popups) durchgeführt. Folgende Auflistung zeigt die Änderungen:

- Minimierung Informationen und Neuordnung.
 - o Tablettenfrage in Popup (vor Anzeige der Tablette)
 - o Anzahl Frage in Popup (vor Anzahldialog)
- Hinzugekommene Dialoge
 - o Begrüßung in Popup
 - o Feedback nach Angabe der eingenommenen Tabletten (beispielsweise erneute Ermutigung für vollständige Einnahme)

Änderungen am Informations-Dialog:

Der Informations-Dialog hat im Cognitive Walkthrough sehr gut abgeschnitten. Aus diesem Grund wurden keine Änderungen am Dialog durchgeführt. Erst Usability-Tests können zeigen, ob noch Verbesserungen am Informations-Dialog durchgeführt werden müssen.

4.3.3.2 Risiko-Analyse und Kontrollmaßnahmen

Wie bereits in Kapitel 3.1 angesprochen, folgt nach dem Entwicklungs-Lebenszyklus-Modell nach jedem Sprint eine erneute Risiko-Analyse. Die erneute Risiko-Analyse baut auf der vorherigen Risiko-Analyse auf und wird mit sämtlichen Änderungen und Erneuerungen des aktuellen Sprints erweitert. Grob zusammengefasst haben sich folgende Punkte verändert:

- Kontrollsoftware (Aufbau vereinfacht)
- Neue Funktionen für Steuersoftware (alte Funktionen der Kontrollsoftware)
- GUI-Steuersoftware (Neubewertung auf Basis der Gebrauchstauglichkeit)

Da der Fokus dieser Arbeit nicht auf dem Risikomanagement liegt, wurde auf eine weitere Risiko-Analyse verzichtet. Es wird davon ausgegangen, dass sich das Gesamtrisiko, im Vergleich zur letzten Risiko-Analyse, im Allgemeinen verbessert hat.

5 Umsetzungsaspekte

5.1 Doppelte Sicherheit

In dem folgenden Abschnitt werden kurz ein paar Beispiele aus der Software für das Prinzip der doppelten Sicherheit behandelt. Die Beachtung der doppelten Sicherheit ist nach 93/42/EWG und MPG bei einem Medizinprodukt Pflicht¹¹⁹.

Steuersoftware und Kontrollsoftware

Wie bereits in Kapitel 4.3.2.5 beschrieben, dient die Kontrollsoftware zur Absicherung der XML- und USB-Schnittstelle. Die dadurch entstehende zweimalige Kontrolle der Befehle wird als doppelte Sicherheit angesehen. Beispielsweise wird so jeder Ausgabebefehl zweimal von zwei unabhängigen Systemen überprüft.

Abfragealgorithmus der Medikamentendosis für Bedarfsmedikamente

Erste Kontrolle: *Anzeige des Buttons, damit Bedarfsmedikamente ausgegeben werden können.*

Damit ein Bedarfsmedikament ausgegeben werden kann (Button wird angezeigt), müssen folgende Bedingungen erfüllt werden:

- Mindestens ein Verordnungssatz muss die Ausgabe zu dem aktuellen Zeitpunkt erlauben.
- Die maximale Tagesdosis des Medikaments des Verordnungssatzes darf nicht erreicht sein.
- Die maximale Momentandosis des Medikaments des Verordnungssatzes darf nicht erreicht sein.

Die folgenden Abbildungen Abbildung 55 und Abbildung 56 zeigen den Algorithmus im Detail.

¹¹⁹ Siehe Kapitel 2.2.2 und ANHANG 10: *Einhaltung der Grundsätze der integrierten Sicherheit...*

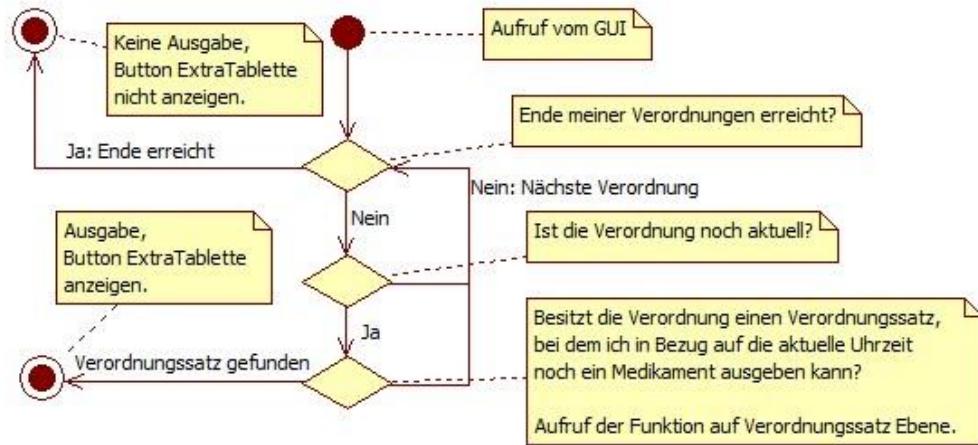


Abbildung 55: Abfragealgorithmus Medikamentendosis 1 (selbst erstellt)

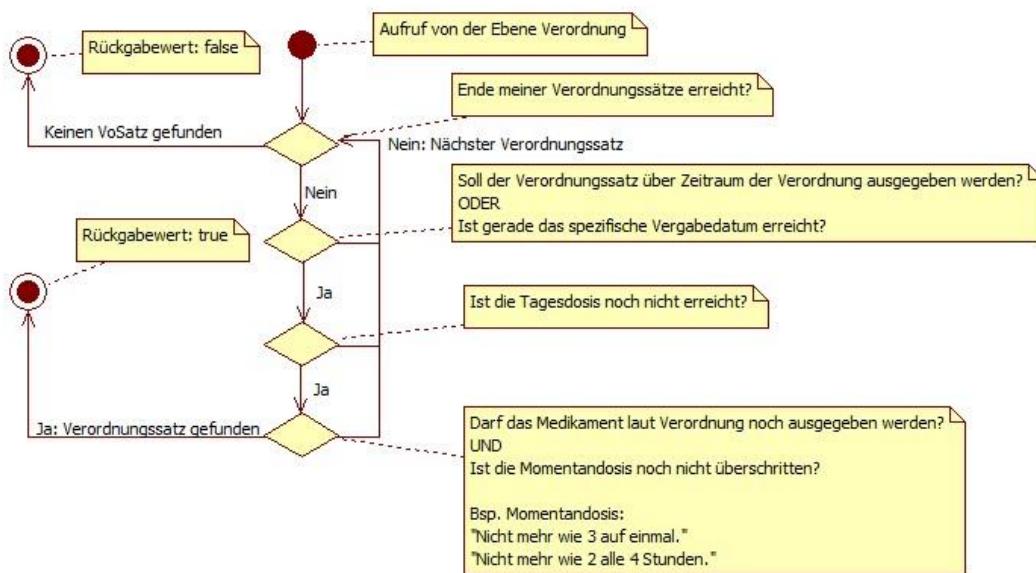


Abbildung 56: Abfragealgorithmus Medikamentendosis 2 (selbst erstellt)

Zweite Kontrolle: Kontrolle der Maximaldosis im Sonderfall Monitor.

Auf Basis der PZN des auszugebenden Bedarfsmedikaments, wird die Maximaldosis für die Ausgabe berechnet. Dazu wird folgendes durchgeführt:

- Nochmalige Durchsuchung sämtlicher Verordnungssätze, die die gesuchte PZN enthalten.
- Berechnung der Maximaldosis auf Basis der Tagesdosis, der Momentandosis mit Sperrzeit des Medikaments und der Angaben im Verordnungssatz.
- Wenn mehrere Verordnungsdatensätze existieren, die für den Zeitpunkt zutreffen, wird immer die am niedrigsten berechnete Dosis verwendet.

Sicherstellung der doppelten Sicherheit: Vor endgültiger Ausgabe der Medikamente im Medikamentenausgabe-Dialog werden die beiden vorherigen Kontrollen durch die Kontrollsoftware nochmals wiederholt. Über die doppelte Kontrolle der Kontrollsoftware wird die geforderte doppelte Sicherheit gewährleistet.¹²⁰

5.2 Defensives Design

Defensives Design ist durch die Norm DIN EN 60601-1-4¹²¹ vorgeschrieben und verlangt einen defensiven Aufbau der Architektur der Subsysteme. Bei der Softwareentwicklung ist defensives Design *und* defensive Programmierung zu beachten. Prinzipiell gilt, dass bei defensivem Design das System sämtlichen Benutzereingaben misstrauisch gegenüber steht. Bei defensiver Programmierung achtet man außerdem darauf, dass Systemressourcen von der Software *exakt* angesprochen werden¹²². Dies wurde beispielsweise durch konkrete Festlegung der Variablen in den Konstruktoren der Klassen gelöst.

Defensives Design ist eng mit der Usability des Systems verknüpft, da die Dialoge mit dem Benutzer direkt beeinflusst werden. Dies trifft besonders bei dem GUI-Design für Benutzer der Hauptgruppe 2 (Patienten) zu. Hier kann der Benutzer nur in vorgegebenen Dialogen agieren. Falscheingaben sind, bis auf einen kritischen Punkt, ausgeschlossen. Dieser kritische Punkt ist die Abfrage, ob der Patient seine Tabletten genommen hat oder nicht. Ein kurzes Beispiel:

Der Patient lässt sich am heutigen Tag bedarfsweise bereits die dritte Einheit eines schmerzlindernden Medikaments ausgeben. Die Tagesdosis des Medikaments beträgt 3 Einheiten. Der Patient nimmt das Medikament zu sich, gibt aber nun über den Einnahmekontrolle-Dialog an, dass er das ausgegebene Medikament nicht genommen hat. Gibt das System jetzt eine weitere Einheit aus, welche über der erlaubten Tagesdosis liegt oder nicht?

Bei dem angesprochenen Beispiel greift der Aspekt des defensiven Designs. Das System muss immer davon ausgehen, dass der Benutzer *lügt* und zählt von daher sämtliche ausgegebenen Einheiten als eingenommen. Der folgende Codeausschnitt zeigt den Zählalgorithmus für die Tagesdosis:

¹²⁰ Die komplette Funktionsweise der Kontrollsoftware ist in 4.3.2.5, S. 92 zu sehen.

¹²¹ Siehe Kapitel 3.3, S. 45: Anforderung an die Architektur

¹²² E. Gunnerson. „C#“. [\[GUNN2000\]](#), Kapitel 33

```

public int GetTagesdosis(DateTime tag)
{
    List<DateTime> gezaehlteTagesdosis =
        ausgabenMitEinnahme.FindAll(delegate(DateTime zeitpunkt)
        { return tag.Date == zeitpunkt.Date; });

    gezaehlteTagesdosis.AddRange(ausgabenOhneEinnahme.FindAll(delegate(DateTime zeitpunkt)
        { return tag.Date == zeitpunkt.Date; }));

    return gezaehlteTagesdosis.Count;
}

```

Hierbei werden zuerst alle ausgegebenen und die als eingenommenen deklarierten Medikamente gezählt. Als zweiten Schritt werden alle ausgegebenen und nicht eingenommenen Medikamente hinzugezählt. Die Gesamtmenge aus beiden Zählungen wird zurückgegeben. Über eine IF-Abfrage, welche die ausgegebene Dosis mit der erlaubten Tagesdosis vergleicht, wird geprüft, ob das Medikament weiter ausgegeben werden darf¹²³. Das System gibt auch bei kritischen Medikamenten, wie beispielsweise Betablockern, keine weiteren Einheiten aus - auch wenn der Patient bei Nicht-Einnahme, einem höheren Risiko ausgesetzt wird (Schlaganfallrisiko). Hier könnte zukünftig ein Warnsystem etabliert werden, welches einen zuständigen Arzt oder das zuständige Pflegepersonal darüber informiert, dass der Patient ein kritisches Medikament nicht genommen hat.

¹²³ Siehe Abfragealgorithmus der Medikamentendosis für Bedarfsmedikamente in Kapitel 5.1, S. 99

5.3 Usability des Systems

Nach der Durchführung von Usability-Tests, die am Ende jeder Entwicklungsphase stehen, sieht C. Backhaus¹²⁴ eine umfangreiche Ermittlung der Gebrauchstauglichkeit¹²⁵ vor. C. Backhaus geht dabei immer von empirischen Usability-Tests¹²⁶ aus. Die ermittelten Defizite aus den Tests werden anhand der Prozessrelevanz weiter kategorisiert. Je nachdem welcher Test als Basis dient, wird eine deskriptive oder quantitative Bewertung der Gebrauchstauglichkeit durchgeführt. Vor einer quantitativen Bewertung werden die Ergebnisse der Tests nochmals quantifiziert. Ein Usability-Test nach C. Backhaus, welcher als Basis für die Bewertung der Gebrauchstauglichkeit dienen kann, ist aufgrund des Aufwands in dieser Arbeit nicht durchführbar. In weiteren Arbeiten könnten Usability-Tests mit dem System durchgeführt werden.

Nur der durchgeführte Cognitive Walkthrough in Kapitel 4.3.3. kann momentan als Anhaltspunkt für die Usability des Systems gelten, wobei zu beachten ist, dass die ermittelten Defizite aus dem CW in die in dieser Arbeit mitgelieferte Version des Systems bereits eingearbeitet sind. Nimmt man die positiven Antworten der Gutachter und rechnet sie gegen die maximal mögliche Anzahl an positiven Antworten, stellt man fest, dass das System bereits vor der Revision des Interfaces¹²⁷ zu 75% von den Gutachtern als *positiv* bewertet wurde. Wie gesagt, dient dieser Wert nur als Anhaltspunkt, in welchem Bereich sich das System wahrscheinlich befindet. Erst ausführliche Usability-Tests mit der gewünschten Zielgruppe können sichere Ergebnisse über die Gebrauchstauglichkeit des Systems liefern.

¹²⁴ C. Backhaus. [\[BACK2009\]](#)

¹²⁵ Siehe Kapitel 3.1, Abbildung 16, S. 40

¹²⁶ Nach N. Römelin [\[RÖME2002\]](#)

¹²⁷ Siehe Kapitel 4.3.3.1, Punkt: *Revision des Interfaces* und [ANHANG 9, Zusammenfassung](#)

6 Diskussion

Die Idee der automatischen Blisterung ist nicht neu. Es existieren bereits diverse Firmen¹²⁸, die eine maschinelle Blisterung in Tüten anbieten. Dieses Angebot gilt jedoch ausschließlich für Pflegeeinrichtungen und Krankenhäuser. Im häuslichen Umfeld wäre der HomeCare Medidisp bisher einmalig. Durch zunehmende Vernetzung (Telemedizin) und Nutzung immer intelligenterer Systeme ist der HomeCare-Bereich ein Zukunftsmarkt. Deshalb halte ich die Verfolgung der Idee eines elektronischen Medikamentendispensers im HomeCare-Bereich für durchaus sinnvoll.

Die gesetzliche Basis ist weitgehend über die Richtlinie 93/42/EWG geregelt und das MPG ist klar festgelegt, jedoch enthalten diese keine konkreten Anforderungen für Produkte im HomeCare-Bereich. Auch kürzlich vorgenommene Änderungen bezüglich der Software haben die gesetzlichen Grundlagen im HomeCare-Bereich nicht verbessert. Schwierig ist daher die Risikoklassifizierung des Produktes gewesen. Für den HomeCare Medidisp wurde letztendlich die Risikoklasse IIa ausgewählt¹²⁹. Das MPG sowie die RL 93/42/EWG sollte für den HomeCare-Bereich und auch für die Telemedizin deutlich ausgeweitet werden.

Die in dieser Arbeit verwendeten Normen¹³⁰ decken ausführlich sämtliche technischen und methodischen Bereiche des HomeCare Medidisp ab, jedoch enthält die - für die Software eher uninteressante - DIN EN 60601-1 eine diskussionswürdige Anforderung. Die Ableitstromgrenze von 0.5 mA¹³¹ ist für die Hardware der Steuersoftware ein wesentlicher Kostenpunkt. In der Anforderungsanalyse wurde kurzerhand entschieden, dass die verwendeten PC-Systeme konform zur DIN EN 60601-1 sein müssen. Solche PC-Systeme sind in der Regel aber wesentlich teurer als normale PC-Systeme. Es stellt sich daher die Frage, ob die 0.5 mA Grenze im HomeCare-Bereich wirklich angebracht ist. Laut Norm wurde die Ableitstromgrenze zum Zweck der Störungsminderung benachbarter Medizinprodukte eingeführt, jedoch ist das *einige* betroffene Medizinprodukt, das wahrscheinlich in der Nähe des PC-Systems stehen wird, der Hardwarepart des HomeCare Medidisp. Im HomCare-Bereich muss man außerdem davon ausgehen, dass Interferenzen durch Ableitströme anderer *Nicht*-Medizinprodukte erzeugt werden. Somit muss der Hardwarepart des HomeCare Medidisp grundsätzlich gegen solche Interferenzen geschützt sein. Das macht wiederum den Gebrauch von richtlinienkonformen PC-Systemen für den Softwarepart hinfällig. Die Schwierigkeit

¹²⁸ Beispielsweise: Automatisches Blisterungssystem von HD Medi

¹²⁹ Siehe Kapitel 2.2.2, S. 14

¹³⁰ Siehe Normen- und Richtlinienregister

¹³¹ Siehe Kapitel 2.3.1, S. 18/Kapitel 4.1, S. 58

der Auswahl der PC-Systeme zeigt, dass Gesetz und auch Normen noch nicht hinreichend für den HomeCare-Bereich ergänzt wurden.

Die vorgeschlagene Entwicklungsmethodik aus der DIN EN 60601-1-4 ließe sich sehr gut auf die geforderte Entwicklungssituation anpassen. Die Aufteilung ab Phase 3 in Softwareentwicklung und Hardwareentwicklung gab genügend Raum für eigene passende Entwicklungsmethoden der einzelnen Komponenten. Die ersten Phasen - mit Anforderungsanalyse und dem ersten groben Entwurf - gaben dem späteren Scrum genügend Grundlagen für eine sehr schnelle und fundierte Entwicklung des Softwareparts. Diese Vorgehensweise hat viele Vorteile, wie beispielsweise Anpassbarkeit, Flexibilität oder gute Grundlagenbildung für Komponentenentwicklung. Von daher zählt diese Methodik nun zu meiner favorisierten Entwicklungsmethodik.

Während der Entwicklung des Softwareparts sind trotz der fundierten Grundlage neue Probleme und Fragen aufgetaucht. Die größte Veränderung war die Neuordnung der Funktionen von Steuersoftware und Kontrollsoftware. Der Kontrollsoftware wurden sämtliche Funktionalitäten, bis auf die Kontrollfunktion, entzogen. Die entzogenen Funktionalitäten wurden der Steuersoftware hinzugefügt. Durch die verringerte Funktionalität der Kontrollsoftware ist deren Komplexität massiv gesunken. Dies hat schließlich die Frage aufgeworfen, ob die Kontrollsoftware auf einem Controller ausgelagert werden sollte¹³² oder ob sie auf dem gleichen Betriebssystem mit der Steuersoftware verbleiben soll. Meiner Meinung nach muss die Kontrollsoftware nun ausgelagert werden. Besonders, da eine Auslagerung das Risiko des *gleichzeitigen* Ausfalls von Steuer- und Kontrollsoftware erheblich verringert. Der Controller könnte in dem Hardwarepart des HomeCare Medidisp angesiedelt sein. Die USB-Schnittstelle müsste lediglich mit den entsprechenden Befehlen erweitert werden.

Ein weiterer Aspekt der Arbeit war die Usability der GUI-Schnittstelle für Benutzer der Hauptgruppe 2 (Patienten) der Steuersoftware. Die schnelle iterative Vorgehensweise mit mehreren Sprints und der anschließenden Sprint-Nachbearbeitung¹³³ erfüllen die geforderten Anforderungen an die Methodik aus den Normen bestens. Die in der Arbeit angewandten Usability-Methoden (Usability-Heuristiken¹³⁴/Cognitive Walkthrough¹³⁴) haben die Usability der GUI-Schnittstelle des Systems deutlich verbessert. Mit diesen beiden angewandten Methoden erfüllt man weitreichend die Anforderungen an den Entwicklungsprozess nach DIN EN 62366. Weitere Methoden, wie beispielsweise der klassische Usability-Test, hätten den Rahmen dieser Arbeit überschritten. Ausführliche Usability-Tests sind aber in

¹³² Neuordnung der Funktionalität der Kontrollsoftware, Kapitel 4.3.2.5

¹³³ Kapitel 4.3.3, S. 94

¹³⁴ Kapitel 4.3.2.2, S. 86/Kapitel 4.3.3.1, S. 94

einer richtlinienkonformen Entwicklung in den späteren Phasen Pflicht. Eine gute Ergänzung der Vorgehensweise nach der DIN EN 62366 bildet die vorgeschlagene Methodik von C. Backhaus¹³⁵. Diese Methodik enthält zwar keine iterative Vorgehensweise zur Verbesserung der Usability, zeigt aber einen guten Weg zur dessen Ermittlung. Besonders die Prozessanalyse¹³⁶ ist eine gute Ergänzung für die Anforderungsanalyse in den ersten Phasen der Entwicklung.

Diese Arbeit hat gezeigt, dass die Idee eines elektronischen Medikamentendispensers grundsätzlich umsetzbar ist. Lediglich die fehlenden gesetzlichen Festlegungen für Medizinprodukte im HomeCare-Bereich erschweren die Umsetzung. Die in dieser Arbeit entwickelte Steuersoftware wurde für Windows XP und Aufwärts mit .NET 4.0 entwickelt. Die jüngste Veröffentlichung von Microsofts Windows 8¹³⁷ zeigt eine deutliche Entwicklung hin zu Tablet-PC-Systemen und Touchscreen-Anwendungen. Deshalb ist eine Ergänzung der momentanen HomeCare-Medidisp-Idee mit Wireless USB zukunftsweisend. Da die Steuersoftware als Touchscreen-Anwendung entwickelt wurde, ist deren Anwendung auf einem Windows-8-Tablet-PC-System durchaus denkbar. In so einer Form stelle ich mir auch HomeCare-Systeme der Zukunft vor: *Als App auf einem Tablet-PC-System.*

¹³⁵ Kapitel 2.3.4, Abbildung 13, S. 33

¹³⁶ Kapitel 4.1.1

¹³⁷ M. Kremp. „Was Sie über Windows 8 wissen müssen“. [\[IQ20\]](#)

7 Zusammenfassung

Diese Arbeit beschäftigt sich mit der Idee eines elektronischen Medikamentendispensers und mit dem richtlinienkonformen Design dessen Softwarekomponente. Dazu wurde zuerst ein Querschnitt durch die rechtlichen Rahmenbedingungen gezogen. Auf Basis der Richtlinie 93/42/EWG und dem Medizinproduktegesetz wurden grundsätzliche Fragen zur Zulassung und zum Design eines solchen Produktes behandelt. Im zweiten Abschnitt behandelt die Arbeit die richtlinienkonforme Entwicklung der Softwarekomponente des elektronischen Medikamentendispensers. Um ein richtlinienkonformes Design zu gewährleisten wurden gängige harmonisierte Normen (DIN EN 60601-1, DIN EN 60601-1-4, DIN EN 62304, DIN EN 62366, DIN EN ISO 14971) als Basis verwendet. Es wurde für die Softwareentwicklung eine gemischte Methodik aus dem Entwicklungsmodell der DIN EN 60601-1-4 und dem Scrum gewählt. Fokus bei der Entwicklung der Software lag auf dem Aspekt der größtmöglichen Sicherheit, verbunden mit der größtmöglichen Gebrauchstauglichkeit für den Benutzer. Dabei hat sich gezeigt, dass die Idee eines elektronischen Medikamentendispensers grundsätzlich umsetzbar ist. Lediglich fehlende gesetzliche Festlegungen für Medizinprodukte im HomeCare-Bereich erschweren eine solche Umsetzung. Die in dieser Arbeit entwickelte Steuerosoftware wurde für Windows XP und Aufwärts mit .NET 4.0 entwickelt. Die jüngste Veröffentlichung von Microsofts Windows 8 zeigt eine deutliche Entwicklung hin zu Tablet-PC-Systemen und Touchscreen-Anwendungen. Da die Steuerosoftware als Touchscreen-Anwendung entwickelt wurde, ist deren Anwendung mit Wireless USB auf einem Windows-8-Tablet-PC-System durchaus denkbar.

8 Literaturverzeichnis

8.1 Normen- und Richtlinienregister

- [606011] DIN EN 60601-1:2003 „*Medizinische elektrische Geräte; Teil 1: Allgemeine Festlegungen für die Sicherheit einschließlich der wesentlichen Leistungsmerkmale*“
- [6060114] DIN EN 60601-1-4:2001 „*Medizinische elektrische Geräte - Teil 1-4: Allgemeine Festlegungen für die Sicherheit Ergänzungsnorm: Programmierbare elektrische medizinische Systeme*“
- [62304] DIN EN 62304: 2006 „*Medizingeräte-Software - Software-Lebenszyklus-Prozesse*“
- [62366] DIN EN 62366:2008 „*Medizinprodukte - Anwendung der Gebrauchstauglichkeit auf Medizinprodukte*“
- [14971] DIN EN ISO 14971:2005 „*Medizinprodukte - Anwendung des Risikomanagements auf Medizinprodukte*“

8.2 Internetquellen

Sämtliche Internetquellen mit dem Index *IQ* sind auf der mitgelieferten CD unter dem gleichen Index zu finden. Alle anderen hier angegebenen Internetquellen sind nur online verfügbar.

- [EKAG2010] Eckelmann AG. „*ECKELMANN AG: V-Modell und agile Softwareentwicklung nach Scrum*“. <http://www.youtube.com/watch?v=swY3EJq5ijM> , Promo-Video (nur online verfügbar), Upload YouTube: 22.12.2010, Stand: 04.06.2011
- [IQ01] BfArM (Bundesinstitut für Arzneimittel und Medizinprodukte) „*Medizinproduktegesetz (MPG)*“. Stand: 07.08.2002, Letzte Änderung: 24.10.2010. <http://www.bfarm.de> / <http://www.gesetze-im-internet.de/mpg/index.html> , 11.10.2011
- [IQ02] EU Recht. „*Richtlinie 93/42/EWG*“. <http://eur-lex.europa.eu/LexUriServ/site/de/consleg/1993/L01993L0042-20031120-de.pdf> , 11.10.2011

- [IQ03] A. Gärtner. „*Patientendatenmanagementsysteme als Software-medizinprodukt? Eine regulatorische Betrachtung*“. http://www.e-health-com.eu/fileadmin/user_upload/dateien/Downloads/Gaertner_Patientendatamanagementsysteme_als_Softwaremedizinprodukt.pdf , 01.04.2011
- [IQ04] A. Gärtner. „*IT und Medizinproduktrecht*“. <http://www.ambb.de/index.php/de/component/content/article/64-it-und-medizinproduktrecht-> , 01.04.2011
- [IQ05] Dr. G. Weisser. „*Medizinische Zweckbestimmung für UMMPerfusion (OsiriX Plugin)*“. http://ikrsrv1.medma.uni-heidelberg.de/redmine/attachments/407/Medizinische_Zweckbestimmung_V1.0.1.pdf , 19.05.2011
- [IQ06] Emergo Group. „*When is Software a medical device?*“. <http://www.emergogroup.com/resources/articles/software-as-medical-device> , 01.04.2011
- [IQ07] S. Donadel. „*Die Norm DIN EN 62366 tritt ab März in Kraft*“. <http://benutzerfreundlichkeit-usability.blogspot.com/2010/03/die-norm-din-en-62366-tritt-ab-marz-in.html> , 01.04.2011
- [IQ08] HJP Datentechnik GmbH. „*HJP Medical Line Produktvorstellung*“. <http://www.hjp.de/medical.html> , 01.04.2011
- [IQ09] Prof. Dr. J. Stettin. „*Risikoklassifizierung von Software - Umsetzung des Risikomanagements für Software nach DIN EN 62304 und IEC/TR 80002-1*“. http://www.baytobio.de/Artikel/2010/100825_Stettin_Vortrag.pdf , 15.04.2011
- [IQ10] bayoonet AG. „*ISO 14971 – Risikomanagement für Medizinprodukte und In-vitro-Diagnostika mit dem Qware Riskmanager*“. <http://www.risikomanagement-14971.de/> , 12.10.2011
- [IQ11] beuth.de. „*Einführungsbeitrag zur Norm DIN EN 62304*“. <http://www.beuth.de/langanzeige/DIN-EN-62304-VDE-0750-101-2007-03/de/96561208.html&bcrumblevel=2&SearchID=338971714> , 12.10.2011

- [IQ12] Dr. C. Lindner. „*Fremdsoftware in Medizinprodukten. Welche Anforderungen sind zu beachten?*“. http://2010.medconf.de/downloads/abstracts2010/T1_T1_V2_tuevsued.pdf , Vortrag auf MedConf 2010, Vortragsdatum 13.10.2010, Downloaddatum 15.05.2011
- [IQ13] Die Beauftragte der Bundesregierung für Informationstechnik (DBBI). „*Häufig gestellte Fragen zum V-Modell XT*“. http://www.cio.bund.de/DE/Architekturen-und-Standards/V-Modell-XT/Haeufig-gestellte-Fragen/haeufig_gestellte_fragen_node.htm , 12.10.2011
- [IQ14] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, D. Thomas. „*Manifesto for Agile Software Development*“. <http://www.agilemanifesto.org/> , 01.06.2011
- [IQ15] it-agile. „*Agile Methoden*“. <http://www.it-agile.de/agile-methoden.html> , 01.06.2011
- [IQ16] it-agile. „*Was ist agile Softwareentwicklung?*“. <http://www.it-agile.de/wasistagilesoftwareentwicklung.html> , 01.06.2011
- [IQ17] it-agile. „*Scrum*“. <http://www.it-agile.de/scrum.html> , 01.06.2011
- [IQ18] Ao.Univ.-Prof. Dr. K. Andrews. „*Andrews Allgemeine Usability-Heuristiken 2006*“. Vorlesungsmaterial, <http://courses.iicm.tugraz.at/hci/practicals/materials/de/he/heuristiken.pdf> , 29.08.2011
- [IQ19] E. Gunnerson. „*C#*“. Galileo Openbook, September 2000 <http://openbook.galileocomputing.de/csharp/> , 04.10.2011
- [IQ20] M. Kremp. „*Was Sie über Windows 8 wissen müssen*“. SPIEGEL ONLINE, 14.09.2011, <http://www.spiegel.de/netzwelt/gadgets/0,1518,786213,00.html> , 13.10.2011

8.3 Literatur

- [BACK2009] C. Backhaus. „*Usability Engineering in der Medizintechnik: Grundlagen – Methoden – Beispiele*“. Springer, Berlin, 2009.
- [CJOH2011] C. Johner, M. Hölzer-Klüpfel, S. Wittorf. „*Basiswissen Medizinische Software: Aus- und Weiterbildung zum Certified Professional for Medical Software*“. Dpunkt Verlag, Auflage 1, 28.03.2011
- [CRHO1999] T. Creter, J. Hofmann. „*Software als Medizinprodukt*“. Technical Report, Eurocat, 1999
- [HARB2001] C. Harböck. „*Richtlinienkonforme Softwareentwicklung von Medizinprodukten am Beispiel eines internetbasierten CTG Monitors mit LabVIEW*“. Diplomarbeit, 15.04.2011
- [HIDC2001] USB.org. „*USB – Device Class Definition for Human Interface Devices (HID)*“. Release 1.11, 27.06.2001
- [HYDE1999] J. Hyde. „*USB Design by Example – A Practical Guide to Building I/O Devices*“. New York, 1999, Wiley.
- [GLOG2008] B. Gloer. „*Scrum – Produkte zuverlässig und schnell entwickeln*“. Carl Hanser Verlag, München, 2008.
- [JAEG2008] Dr. B. Jaeger. „*Änderungen bei der Medizinproduktrichtlinie 93/42/EWG*“. EDI-Journal 2/2008, S. 40
- [MEIE2005] Jens. O. Meiert. „*Usability-Heuristiken*“. 25. April 2006. ISSN 1614-3124, Ausgabe 17
- [PHDC2007] USB.org. „*USB – Device Class Definition for Personal Healthcare Devices*“. Release 1.0, 08.11.2007
- [RÖME2002] N. Römeling. „*Usability im World Wide Web. Redesign eines Webauftritts mit Hilfe von Usability Testing*“. Diplomarbeit, 09.09.2002, Augsburg.
- [UHLE2000] J. Uhlemann, T. Kollbeck, R. Freyer. „*Sicherheitsrelevante Aspekte der internationalen Normen für Infusionspumpen*“. Biomedizinische Technik/Biomedical Engineering. Band 45, Ergänzungsband 1, Seiten 298–299, 2000
- [USB20] USB.org. „*Universal Serial Bus Specification Revision 2.0*“. Revision 2.0, 27.04.2000

9 Anhänge

9.1 Anhang 1

Nicht invasive Produkte

- Regel 1. Alle nicht invasiven Produkte Klasse I, es sei denn Regel 2-4 treffen zu
- Regel 2. Durchleitung/Aufbewahrung von Blut und anderen Körperflüssigkeiten: Klasse IIa
- Regel 3. Veränderung von Blut/Körperflüssigkeiten
 - Perfundierung: Klasse IIb
 - Filtration/Zentrifugierung/Wärme-/Gasaustausch: Klasse IIa
- Regel 4. Kontakt mit verletzter Haut
 - Leichte Wunden: Klasse I
 - Schwere Wunden: Klasse IIb
 - Desinfektion: Klasse IIa

Invasive Produkte

- Regel 5. In Zusammenhang mit Körperöffnungen/kein Anschluss an aktives Produkt
 - vorübergehende Anwendung: Klasse I
 - kurzzeitige Anwendung: Klasse IIa
 - Klasse I, wenn Mund/Rachen/Gehörgang bis Trommelfell
 - langzeitige Anwendung: Klasse IIb
 - Klasse IIa, wenn Mund/Rachen/Gehörgang bis Trommelfell
- Regel 6. Bei chirurgisch-invasiver und vorübergehender Anwendung: Klasse IIa
 - Klasse III, wenn Anwendung Herz/Herz-Kreislaufsystem
 - Klasse I, wenn wiederverwendbare Werkzeuge
 - Klasse IIb, wenn Abgabe von Strahlung/vollständige Resorption (Fäden)/Verabreichung von Arzneimitteln

Regel 7. Bei chirurgisch-invasiver und kurzzeitiger Anwendung:
Klasse IIa

- Klasse III, wenn Anwendung Herz/Herz-Kreislaufsystem/Nervensystem/vollständige Resorption (Fäden)
- Klasse IIb, wenn Abgabe von Strahlung/Verabreichung von Arzneimitteln

Regel 8. Implantate, sowie langzeitig angewandte chirurgisch-invasive Produkte: Klasse IIb

- Klasse IIa, wenn Zahnimplantat
- Klasse III, wenn Herz/Herz-Kreislaufsystem/Nervensystem/vollständige Resorption (Fäden)/Verabreichung von Arzneimitteln

Aktive Produkte

Regel 9. Aktive therapeutische Produkte, die zur Abgabe oder zum Austausch von Energie bestimmt sind: Klasse IIa

- in Sonderfällen (hohe Energiedichte, etc.): Klasse IIb

Regel 10. Aktive diagnostische Produkte: Klasse IIa

- Klasse IIb, wenn Kontrolle von Vitalfunktionen/Abgabe von ionisierter Strahlung

Regel 11. Aktive Produkte zur Abgabe/Entfernung von Körperflüssigkeiten/Stoffen/Arzneimitteln (z.B. Infusionspumpen): Klasse IIa

- in Sonderfällen: Klasse IIb

Regel 12. Restliche aktive Produkte: Klasse I

Sonderregeln

Regel 13. Wenn ein Produkt als Bestandteil einen Wirkstoff, der nach 65/65/EWG ein Arzneimittel ist, besitzt: Klasse III

- wenn Produkt als Bestandteil ein Derivat aus menschl. Blut besitzt: Klasse III

Regel 14. Empfängnisverhütung: Klasse IIb

- Implantat oder invasiv langzeitig: Klasse III

Regel 15. Desinfektion/Reinigung: Klasse IIa

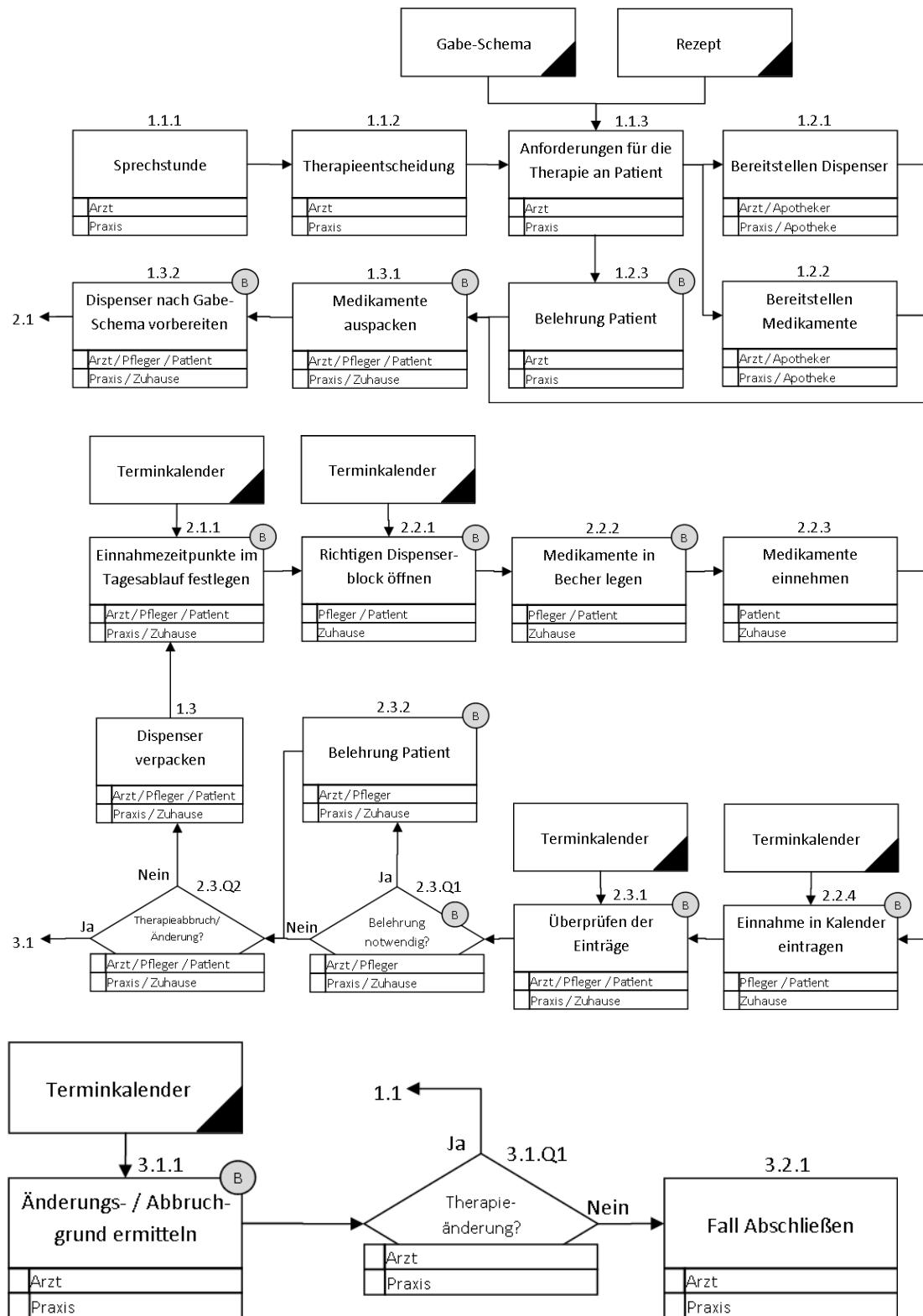
- Kontaktlinsen: Klasse IIb

Regel 16. Nicht aktive Produkte für Röntgenaufzeichnung: Klasse IIa

Regel 17. Produkte, welche durch tierisches Gewebe hergestellt wurden: Klasse III

Regel 18. Blutbeutel: Klasse IIb

9.2 Anhang 2



Prozessanalyse nach C. Backhaus [BACK2009] (Detailansichten selbst erstellt)

9.3 Anhang 3

Die Kommunikation zwischen den Modulen (Steuer-/Hardwaremodul) wird mit Hilfe von USB bewerkstelligt. Unser Steuermodul mit Windows als Betriebssystem übernimmt die Host-Rolle und das Hardwaremodul die Client(Device)-Rolle.

9.3.1 USB-Kommunikation

Sobald Host und Device mittels USB-Kabel verbunden sind, beginnt ein standardisierter Anmeldeprozess zwischen Device und Host. In Abbildung A3 1 ist ein einfacher Aufbau zwischen Device – USB – Host zu sehen. Ein USB-Kabel besitzt 4 Adern. D+, D- sind die Signaladern und VCC, Gnd die Stromversorgungsadern. Solange kein USB Device angeschlossen ist, sind durch den doppelten Vorwiderstand beim Host Controller die Signaladern D+ und D- auf low. Der Host befindet sich momentan im Modus *unattached*. Sobald ein USB Device angeschlossen wird, verändert sich die Spannung bei D+ oder D-. Der Host Controller registriert den Spannungsunterschied und meldet, dass ein USB-Kabel angeschlossen wurde. Je nachdem, ob D+ (12 Mbps) oder D- (1.5 Mbps) auf high sind, passt der Host Controller die USB-Geschwindigkeit an. Danach schaltet der Host in den Modus *attached*. Das Device schaltet vom *unattached*-Modus zum *powered*-Modus. Nun beginnt der Host mit dem sogenannten Handshake.

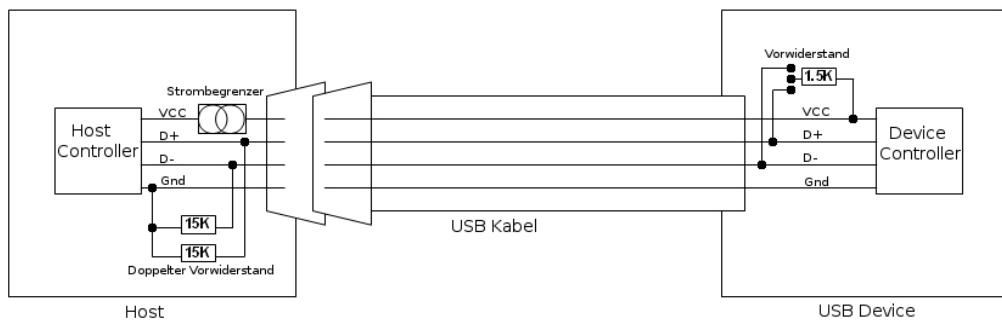


Abbildung A3 1: Simpler USB-Aufbau [[HYDE1999](#)]

Handshake

Im folgenden Abschnitt wird in kurzen Schritten der Handshake zwischen Host und USB Device erläutert. Laut J. Hyde¹³⁸ und der USB 2.0 Spezifikation¹³⁹ beginnen Host und Device mit dem Handshake folgendermaßen:

¹³⁸ J. Hyde. "USB Design by Example – A Practical Guide to Building I/O Devices". [[HYDE1999](#)]

¹³⁹ USB.org. "Universal Serial Bus Specification Revision 2.0". [[USB20](#)]

1. **Get_Port_Status:** Host entdeckt neues USB Device. Das Device befindet jetzt im *attached*-Modus.
2. **Clear_Port_Feature(C_PORT_CONNECTION):** Löscht den Flag im STATUS_CHANGE-Register. Das Device befindet sich jetzt im *powered*-Modus.
3. **Set_Port_Feature(PORT_RESET):** Host sendet ein *reset* zum Device. Danach setzt der Host in seinen Registern die Bits RESET_CHANGE und PORT_ENABLE und aktiviert den Port für die Datenübertragung.
4. **Get_Port_Status:** Reset-Vorgang abgeschlossen. Das Device befindet sich jetzt im *default*-Modus.
5. **Clear_Port_Feature(C_PORT_RESET):** Löscht den Flag im STATUS_CHANGE-Register

Das Device ist nun bereit bestimmte Nachrichten vom Host zu erhalten und zu senden. Der Host sendet momentan noch an die default-Adresse des Devices (Control Endpoint: End0). Jetzt erfolgt die Identifikation des Devices. Der Host sendet mehrere standardisierte Anfragen an das Device.

1. **Get_Device_Descriptor:** Das USB Device sendet auf die Anfrage hin seinen Device Descriptor.
2. **Set_Adress:** Der Host hat eine devicespezifische Adresse festgelegt. Das Device selbst schaltet sich nun in den *addressed*-Modus.
3. **Get_Device_Descriptor:** Der Host sendet die obige Anfrage erneut an die neu zugewiesene Adresse und vergleicht beide Deskriptoren. Sind die Deskriptoren unterschiedlich, sendet der Host Controller einen Error.
4. **Get_Configuration_Descriptor:** Das USB Device sendet auf die Anfrage hin sämtliche restlichen Deskriptoren.
5. **Select_Device_Driver:** Auf Basis der erhaltenen Deskriptoren ordnet der Host PC dem USB Device einen passenden Treiber zu.
6. **Set_Configuration:** Host sendet an das Device das Ende des Handshakes. Das USB Device schaltet in den *configured*-Modus und ist nun betriebsbereit.

9.3.2 USB-Deskriptoren

Deskriptoren stellen ein standardisiertes Konfigurationsgerüst für USB Devices dar, womit der Host Controller die angeschlossenen USB Devices über einen Handshake identifizieren kann. Im Handshake wird zuerst der Device Descriptor übertragen, dann die Configuration-, Interface- und Endpoint-Deskriptoren. Jeder Device Descriptor kann beliebig viele Configuration-Deskriptoren enthalten. Das gleiche gilt für Interface-Deskriptoren und Endpoint-Deskriptoren. Somit ergibt sich folgendes Beispiel für eine Device-Konfiguration (Abbildung A3 2).

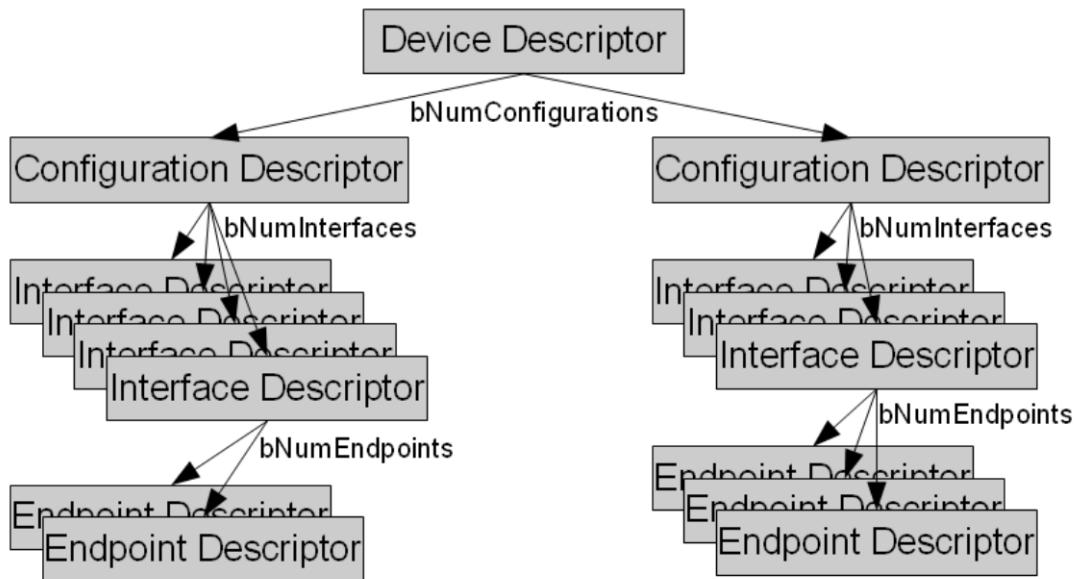


Abbildung A3 2: Beispielhafter Descriptor-Aufbau (selbst erstellt)

Device Descriptor

Tabellen A3 1 bis 4 zeigen beispielhaft den Aufbau eines Device, Configuration, Interface und Endpoint Descriptor für ein HID Device für USB-2.0.-HID-Geräte besitzen zusätzlich zu den Standard-Deskriptoren noch spezielle HID- und Report-Deskriptoren, auf die ich in dieser Arbeit nicht näher eingehen werde.

| Feldname | Offset/Größe | Bsp. Value | Beschreibung |
|-----------------|--------------|------------|---|
| bLength | 0/1 | 0x12 | Länge des Deskriptors in bytes |
| bDescriptorType | 1/1 | 0x01 | Descriptor Type |
| bcdUSB | 2/2 | 0x200 | USB-Specification-Release-Nummer |
| bDeviceClass | 4/1 | 0x00 | USB Device Class Code (00h = DCC im Interface Descriptor) |
| bDeviceSubClass | 5/1 | 0x00 | USB Device Sub Class Code |
| bDeviceProtocol | 6/1 | 0x00 | USB Device Protocol Code |
| bMaxPacketSize0 | 7/1 | 0x08 | Maximale Paketgröße für Endpunkt 0 |
| idVendor | 8/2 | 0xFFFF | USB.org registrierte Vendor ID |
| idProduct | 10/2 | 0x0001 | Product ID, zugeordnet vom Hersteller |
| bcdDevice | 12/2 | 0x0100 | Device-Release-Nummer |

| | | | |
|--------------------|------|------|--|
| iManufacturer | 14/1 | 0x00 | Index String Array für Beschreibung Hersteller |
| iProduct | 15/1 | 0x00 | Index zum String Array für die Beschreibung des Produkts |
| iSerialNumber | 16/1 | 0x00 | Index zum String Array für die Beschreibung der Seriennummer des Devices |
| bNumConfigurations | 17/1 | 0x01 | Anzahl der Configuration-Deskriptoren |

Tabelle A3 1: Beispiel Device Descriptor (selbst erstellt)

Configuration Descriptor

| Feldname | Offset/Größe | Bsp. Value | Beschreibung |
|---------------------|--------------|------------|--|
| bLength | 0/1 | 0x09 | Länge des Deskriptors in bytes |
| bDescriptorType | 1/1 | 0x02 | Descriptor Type |
| wTotalLength | 2/2 | 0x0019 | Totale Länge aller Interface- und Endpoint-Deskriptoren, inklusive Configuration Descriptor |
| bNumInterfaces | 4/1 | 0x01 | Anzahl der Interface-Deskriptoren für diesen Configuration Descriptor |
| bConfigurationValue | 5/1 | 0x01 | Argument, um die Konfiguration auszuwählen |
| iConfiguration | 6/1 | 0x00 | Index zum String Array für die Beschreibung der Konfiguration |
| bmAttributes | 7/1 | 10100000 | Konfigurationen (Bit 7 = Bus Powered, Bit 6 = Self Powered, Bit 5 = Remote Wakeup, Bit 4...0 = Reserved) |
| MaxPower | 8/1 | 0x32 | Maximaler Stromverbrauch des Devices mit dieser Konfiguration (1 = 2mA, 32 = 64mA) |

Tabelle A3 2: Beispiel Configuration Descriptor (selbst erstellt)

Interface Descriptor (Bsp. Keyboard)

| Feldname | Offset/Größe | Bsp. Value | Beschreibung |
|--------------------|--------------|------------|---|
| bLength | 0/1 | 0x09 | Länge des Deskriptors in bytes |
| bDescriptorType | 1/1 | 0x04 | Descriptor Type |
| bInterfaceNumber | 2/1 | 0x00 | Index des Interfaces im Interface Array, angefangen bei 0 |
| bAlternateSetting | 3/1 | 0x01 | Identifier für alternative Settings |
| bNumEndpoints | 4/1 | 0x01 | Anzahl der Endpoint-Deskriptoren, ausgeschlossen Endpoint 0 (Wenn der Wert 0 ist, wird ausschließlich Endpoint 0 verwendet) |
| bInterfaceClass | 5/1 | 0x03 | USB Device Class Code (03h = HID) |
| bInterfaceSubClass | 6/1 | 0x01 | Subclass Code (0 = keine Subclass, 1 = Boot Interface subclass) |
| bInterfaceProtocol | 7/1 | 0x01 | Protocol Code (0 = keins, 1 = Keyboard, 2 = Maus) |
| iInterface | 8/1 | 0x00 | Index zum String Array für die Beschreibung des Interfaces |

Tabelle A3 3: Beispiel Interface Descriptor (selbst erstellt)

Endpoint Descriptor (Bsp. Keyboard)

Ein USB Device darf maximal nur 31 Endpoints besitzen. Den Control Endpoint (End0) und je 15 In- und Out-Endpoints

| Feldname | Offset/Größe | Bsp. Value | Beschreibung |
|-----------------|--------------|------------|---|
| bLength | 0/1 | 0x07 | Länge des Deskriptors in bytes |
| bDescriptorType | 1/1 | 0x05 | Descriptor Type |
| bEndpointAdress | 2/1 | 10000001 | Adressendefinierung für den Endpoint (Bit 0...3 = Endpoint Nummer, Bit 4...6 = Reserviert, Bit 7 = Richtung 0 – OUT / 1 – IN) |
| bbmAttributes | 3/1 | 00000011 | Weitere Konfigurationen (Bit 2...7 = Reserviert, Bit 0...1 Transfer type, 00 Control, 01 Isochronous, 10 Bulk, 11 Interrupt) |
| wMaxPacketSize | 4/2 | 0x0008 | Maximale Paketgröße für diesen Endpoint |
| bInterval | 6/1 | 0x0A | Abrufintervall in Millisekunden |

Tabelle A3 4: Beispiel Endpoint Descriptor (selbst erstellt)

9.3.3 USB-Transfermodi

Wie in Tabelle A3 4 zu sehen ist, unterstützt USB über die Endpoints verschiedene Transfermodi. Den Control, Isochron, Bulk und Interrupt Transfer. Diese Transfermodi unterscheiden sich in der Übertragungsgeschwindigkeit und der möglichen Datenmenge, die übertragen werden kann.

Der Control Transfer bildet eine Ausnahme. Dieser ist für die Konfiguration des USB Devices nötig und benötigt einen Endpunkt der In- sowie Out-Operationen durchführen kann. Der Endpunkt 0 verwendet beispielsweise diesen Transfer-Modus.

Der isochrone Transfermodus ist für Daten geeignet, die eine garantierte Datenrate benötigen, wie z.B. Audiostreams. Dafür wird in einem bestimmten Abfrageintervall eine bestimmte Datenmenge an den Endpunkt gesendet. Die Übertragungsrate berechnet sich aus dem Produkt des Abfrageintervalls und der Größe des Datenbuffers.

Der Interrupt -Transfermodus ist für kleine Datenmengen, die sporadisch eintreffen, gedacht. Auch hier wird ein Abfrageintervall verwendet, um den Buffer regelmäßig zu überprüfen.

Im Bulk-Transfermodus sind hingegen große Datenmengen übertragbar. Bulk Transfers werden niedriger priorisiert als Isochrone oder Interrupt Transfers und werden von jeweiligen Controller erst dann durchgeführt, wenn keine Isochrone oder Interrupt Transfers mehr offen sind. Bulk Transfers werden durch die Hardware bis zu dreimal wiederholt.

9.4 Anhang 4

Hauptgruppe 1: Profil 1 (Hausarzt)

| | |
|--|--|
| Dr. Monika Philipp Lebensmotto: Ich muss immer auf dem neuesten Stand sein. | Alter: 55 Funktion: Hausarzt, Spezialisierungen: Innere Medizin, Lymphologie Ausbildungsniveau: Medizinstudium, Promotion, FA innere Medizin, besucht häufig Fachkongresse Lehrgänge Sprachlicher/kultureller Hintergrund: deutsch Eigenheiten: ungeduldig, lernt nur langsam den Umgang mit neuer Software, statisch Geistige, physische, demographische Besonderheiten: Brillenträger Familiäres: verheiratet, 1 Kind Sonstiges: Workaholic, arbeitet ~12h am Tag Arbeitsumfeld: Arztpraxis Produktkenntnisse: kennt die Vorgehensweise bei einem normalen Dispenser Häufigste Tätigkeit mit Produkt: Eingabe des Verordnungsdatensatzes/von pharmakologischen Profilen Wichtigste Tätigkeit mit Produkt: Eingabe des Verordnungsdatensatzes Seltenste Tätigkeit mit Produkt: Medikamentenausgabe Typische Hindernisse beim Umgang mit dem Produkt: Lernfähigkeit bei der Eingabe der Verordnungsdatensätze |
|--|--|

Hauptgruppe 1: Profil 2 (Servicekraft/Hersteller)

| | |
|--|--|
| Herr Meyer Lebensmotto: Ganz ruhig, das wird schon. | Alter: 30 Funktion: Techniker Ausbildungsniveau: Master Medizininformatik, gelegentlich auf Fachmessen, Spezialisierung für das Produkt Sprachlicher/kultureller Hintergrund: deutsch Eigenheiten: ruhig, professionell Geistige, physische, demographische Besonderheiten: Brillenträger Familiäres: ledig Sonstiges: - Arbeitsumfeld: beim Benutzer (Außendienst) Produktkenntnis: komplettes Wissen über das Produkt Häufigste Tätigkeit mit Produkt: Eingabe von pharmakologischen Profilen, Reparaturen Wichtigste Tätigkeit mit Produkt: Reparaturen Seltenste Tätigkeit mit Produkt: Medikamentenausgabe Typische Hindernisse beim Umgang mit dem Produkt: - |
|--|--|

Hauptgruppe 2: Profil 1 (Kognitiv nicht eingeschränkter Patient)

| | |
|--|---|
| <p>Frau Müller</p> <p>Lebensmotto: Ich brauche nichts mehr lernen.</p> | <p>Alter: 75 Funktion: Patient Ausbildungsniveau: mittlerer Abschluss, Ausbildung als Einzelhandelskauffrau Sprachlicher/kultureller Hintergrund: deutsch Eigenheiten: eigensinnig, statisch Geistige, physische, demographische Besonderheiten: - Familiäres: verwitwet Sonstiges: Oft im Garten Arbeitsumfeld: Rentnerin Produktkenntnisse: keine Häufigste Tätigkeit mit Produkt: Medikamentenausgabe Wichtigste Tätigkeit mit Produkt: Medikamentenausgabe Seltenste Tätigkeit mit Produkt: - Typische Hindernisse beim Umgang mit dem Produkt: Lernfähigkeit beschränkt, leidet an Altersschwäche</p> |
|--|---|

Hauptgruppe 2: Profil 2 (Kognitiv eingeschränkter Patient)

| | |
|--|--|
| <p>Herr Müller</p> <p>Lebensmotto: Einfach muss es sein.</p> | <p>Alter: 60 Funktion: Patient Ausbildungsniveau: mittlerer Abschluss, Ausbildung als Chemiker und Polizeibeamter Sprachlicher/kultureller Hintergrund: deutsch Eigenheiten: cholerisch, phlegmatisch Geistige, physische, demographische Besonderheiten: Brillenträger Familiäres: verheiratet, 3 Kinder Sonstiges: sitzt viel vor dem Computer Arbeitsumfeld: Dienstzimmer Polizei, Arbeit am Schreibtisch Produktkenntnisse: keine Häufigste Tätigkeit mit Produkt: Medikamentenausgabe Wichtigste Tätigkeit mit Produkt: Medikamentenausgabe Seltenste Tätigkeit mit Produkt: - Typische Hindernisse beim Umgang mit dem Produkt: -</p> |
|--|--|

9.5 Anhang 5

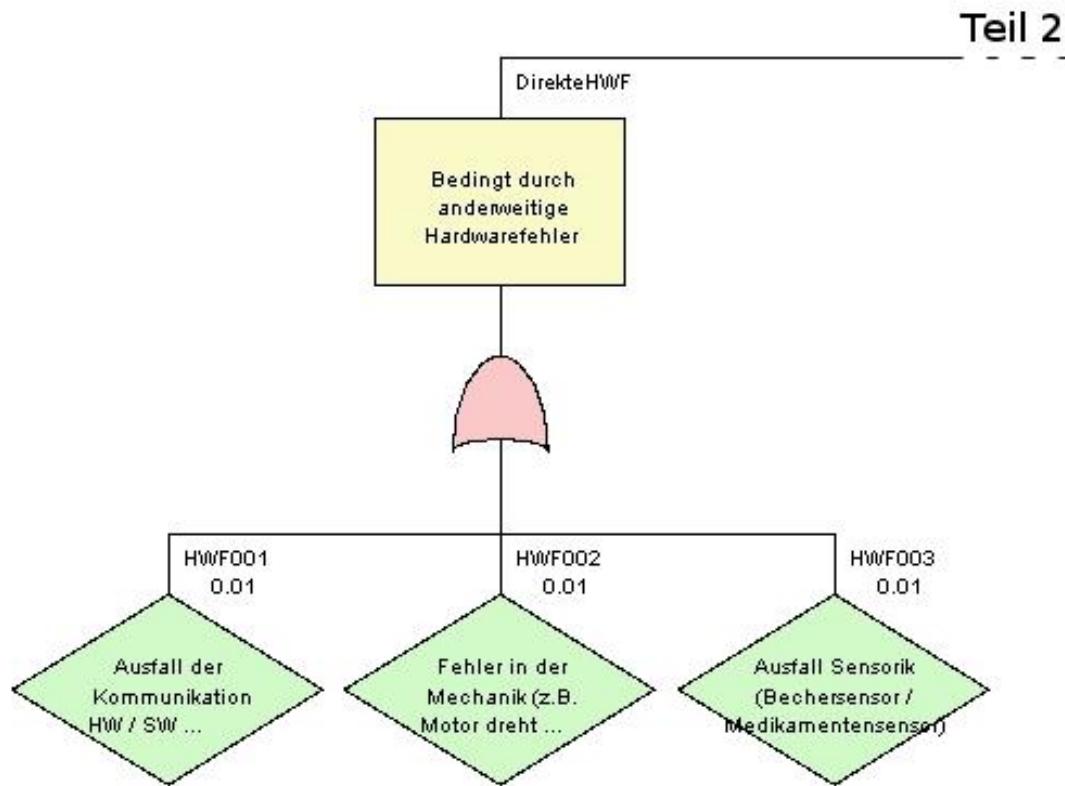


Abbildung A5 1: Fehlerbaum Teil 1

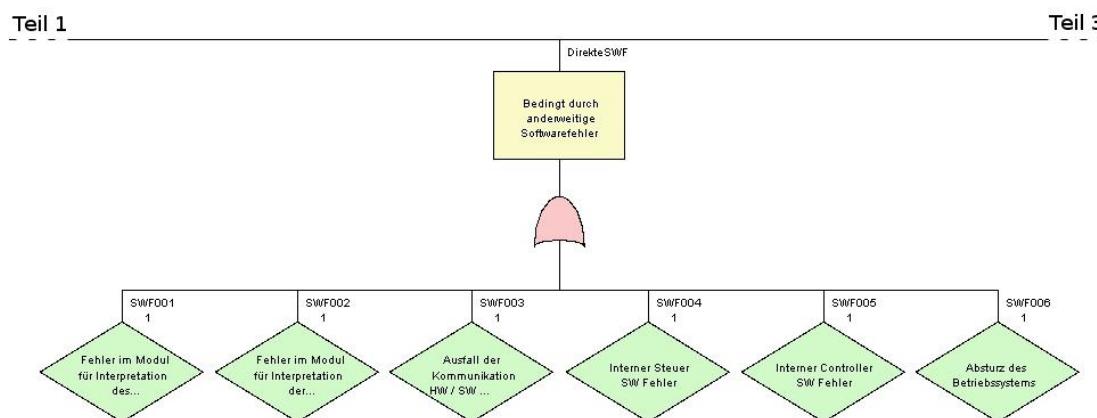


Abbildung A5 2: Fehlerbaum Teil 2

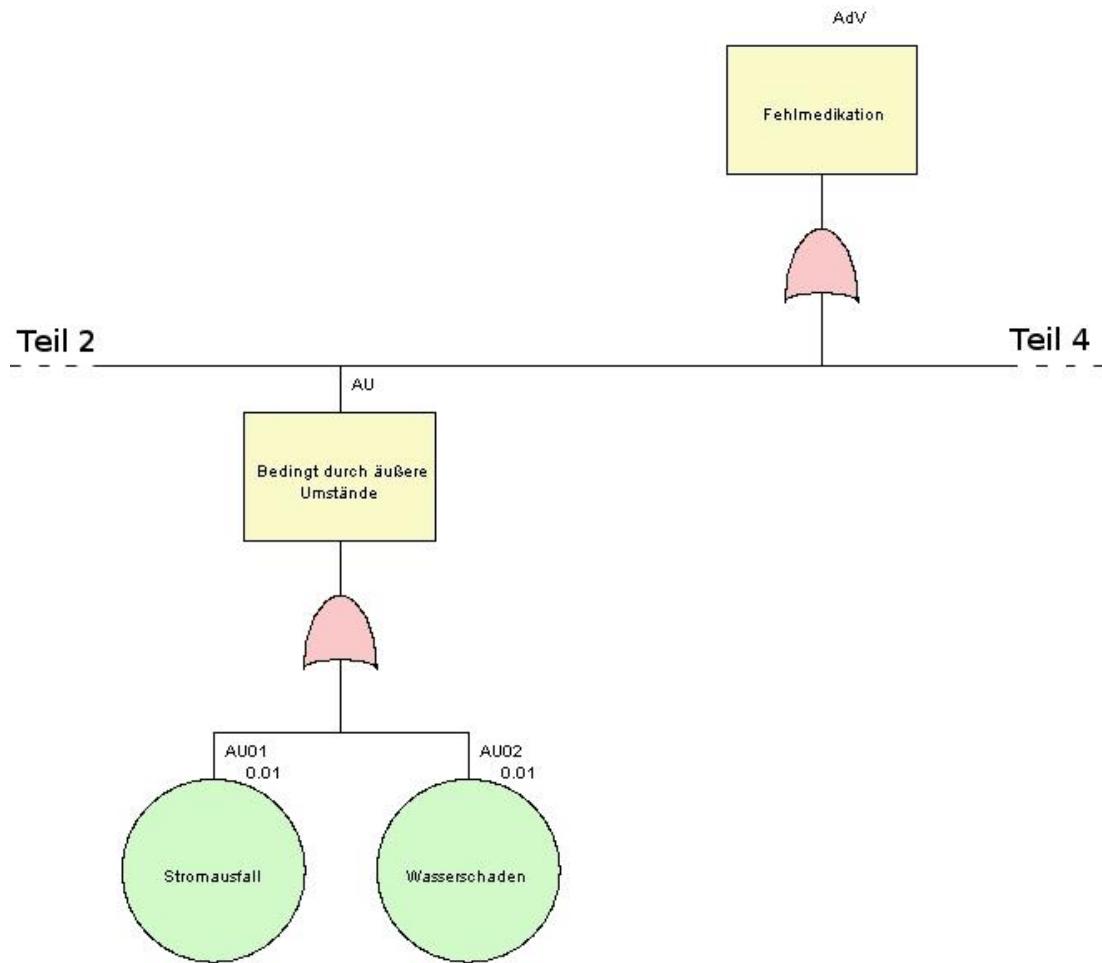


Abbildung A5 3: Fehlerbaum Teil 3

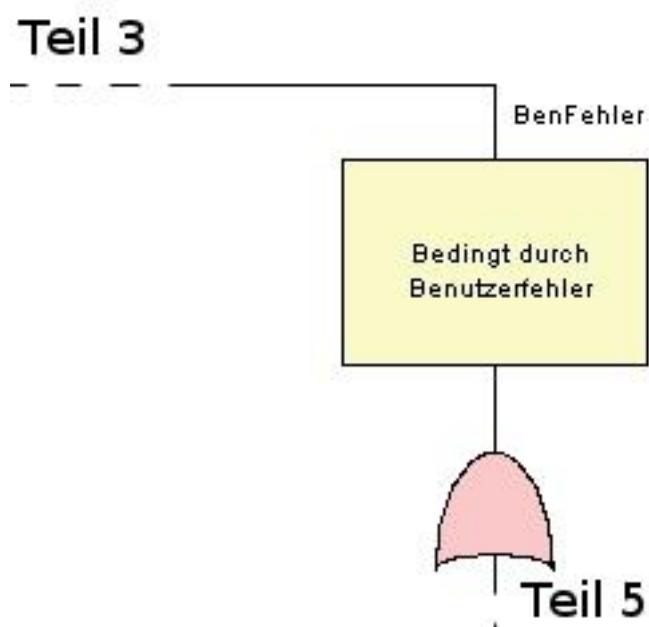


Abbildung A5 4: Fehlerbaum Teil 4

Teil 4

Teil 6

Teil 8

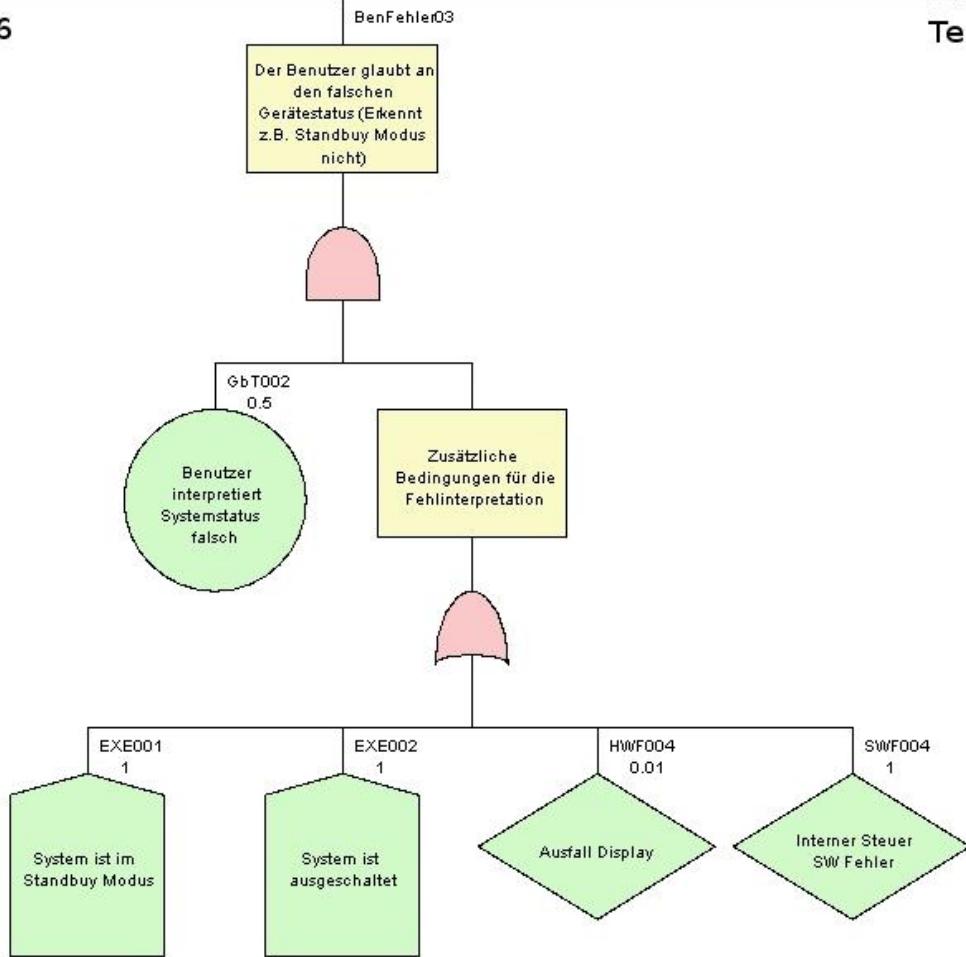


Abbildung A5 5: Fehlerbaum Teil 5

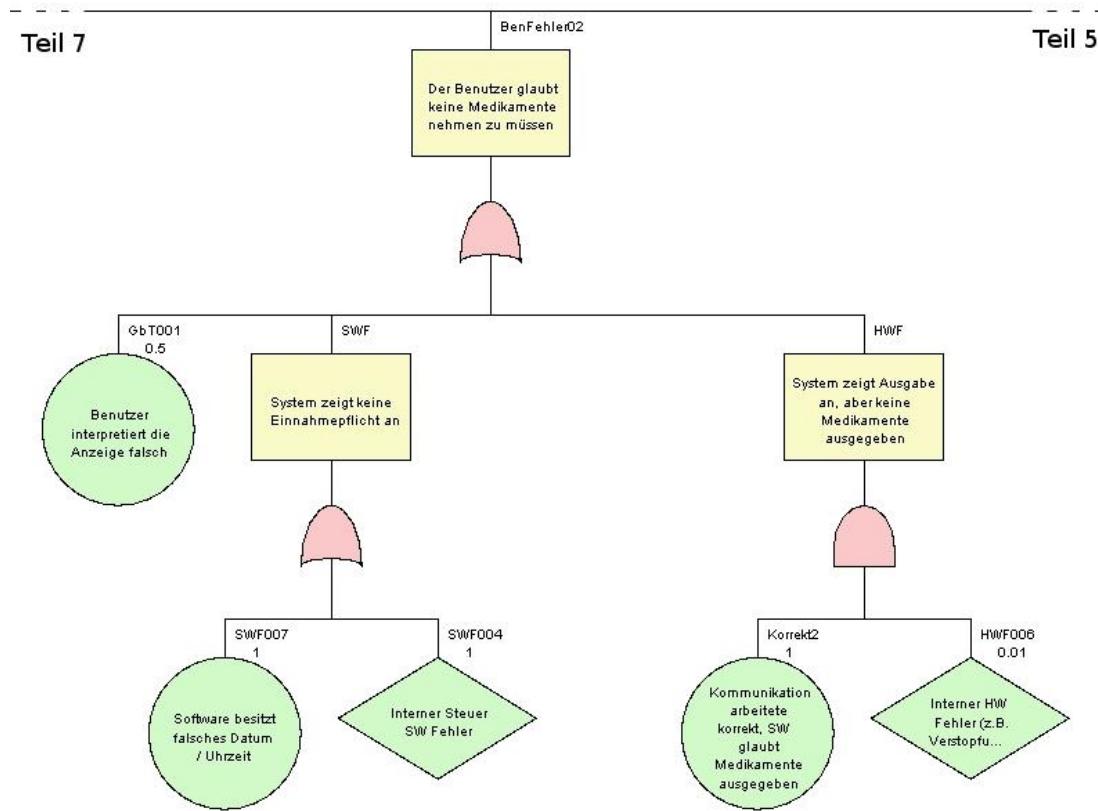


Abbildung A5 6: Fehlerbaum Teil 6

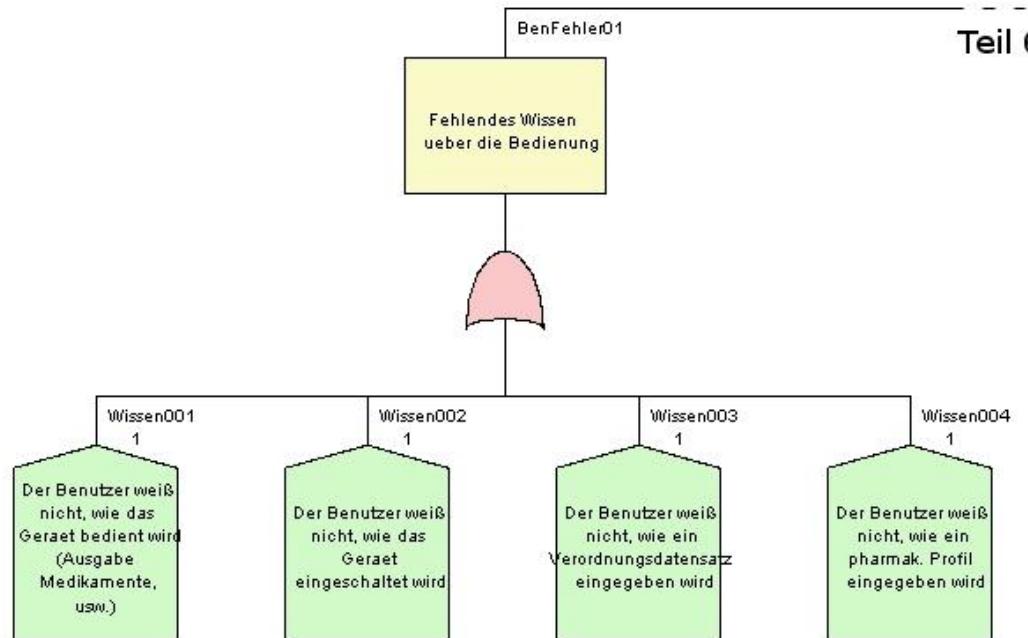


Abbildung A5 7: Fehlerbaum Teil 7

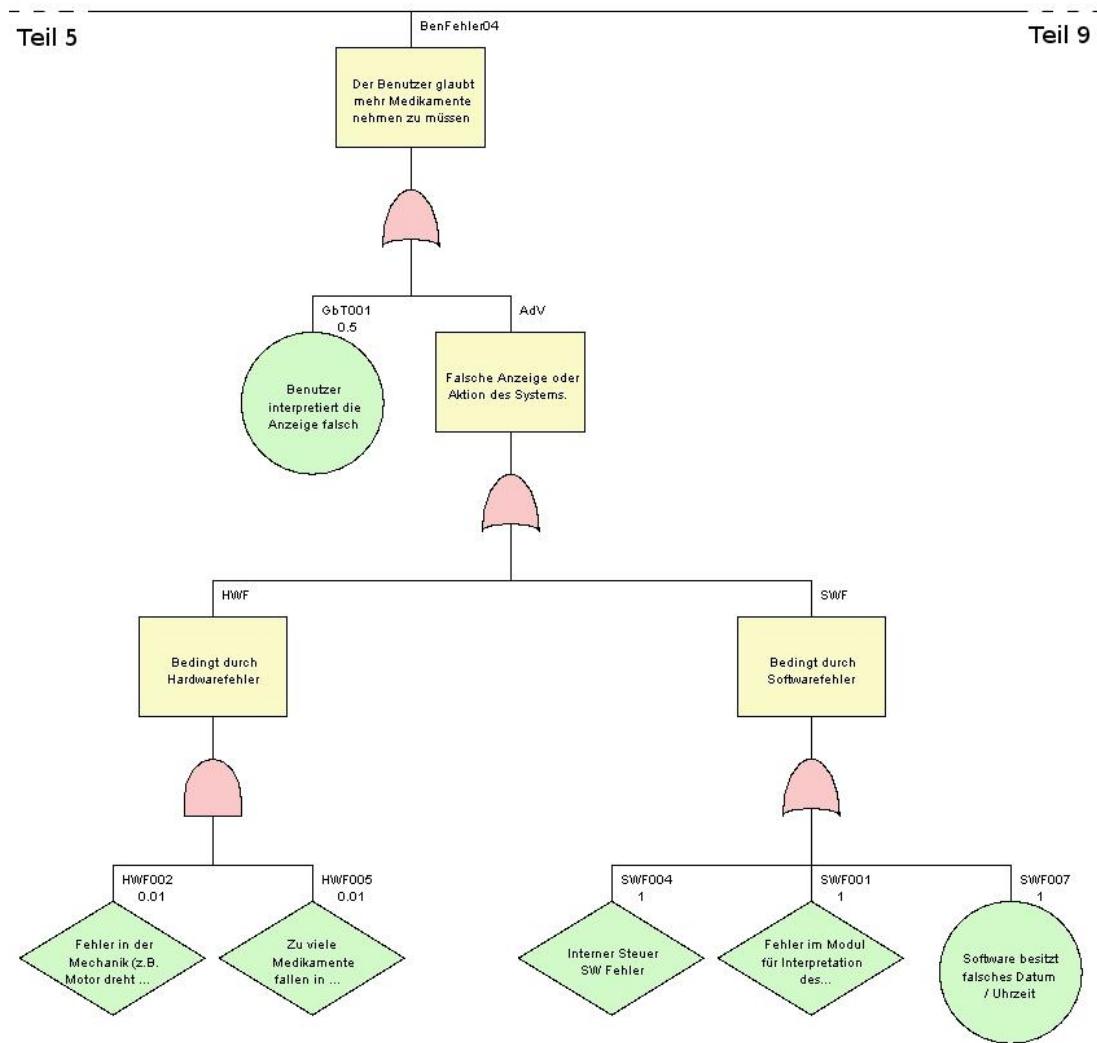


Abbildung A5 8: Fehlerbaum Teil 8

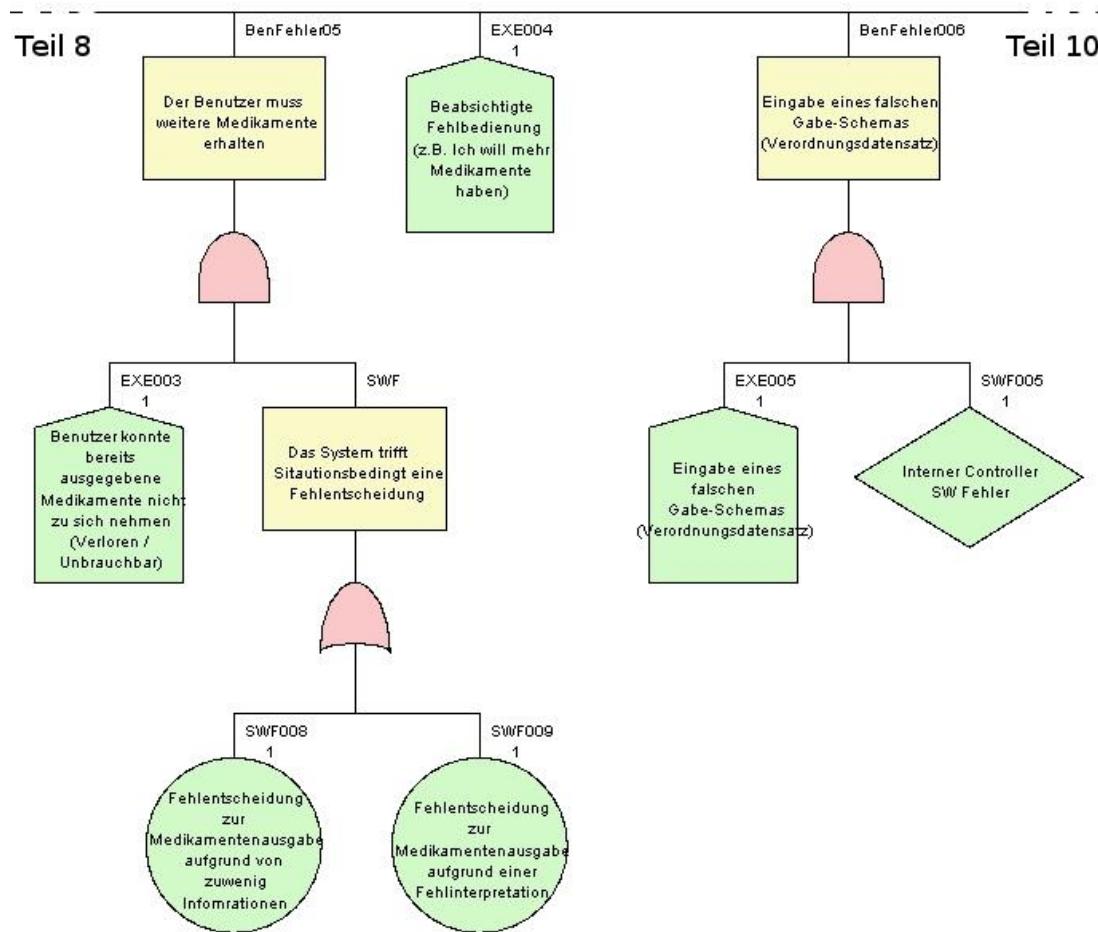


Abbildung A5 9: Fehlerbaum Teil 9

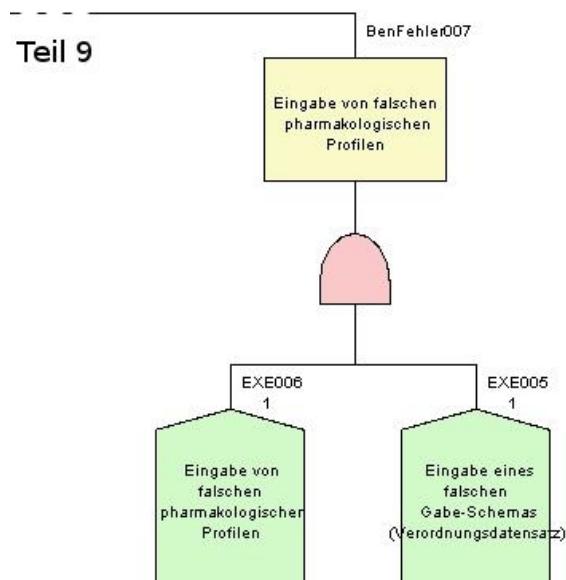


Abbildung A5 10: Fehlerbaum Teil 10

9.6 Anhang 6

9.6.1 Änderungshistorie

| Datum | Bearbeiter/-in | Änderung |
|------------|-----------------|------------------------------|
| 23.06.2011 | Torsten Philipp | Initiale Version |
| 23.06.2011 | Torsten Philipp | User Story 1 – 8 hinzugefügt |

9.6.2 Backlog-Übersicht

| ID. | Titel | Status | Priorität | Release / Sprint |
|------|---|-----------|-----------|------------------|
| US 1 | Entwicklung Simulationsumgebung | In Arbeit | 1 | R.1 / S.1 |
| US 2 | Entwicklung eines ergonomischen GUI | In Arbeit | 1 | R.1 / S.1 |
| US 3 | Entwicklung eines richtlinienkonformen USB-Moduls | In Arbeit | 1 | R.1 / S.1 |
| US 4 | Entwicklung eines richtlinienkonformen XML-Moduls | In Arbeit | 1 | R.1 / S.1 |
| US 5 | Entwicklung einer unabhängigen Kontrollsoftware | In Arbeit | 1 | R.1 / S.1 |
| US 6 | Umsetzung eines interaktiven Terminkalenders | Offen | 2 | R.1 / S.2 |
| US 7 | Umsetzung eines Internetmoduls für Onlinebearbeitung von Verordnungsdatensätzen und pharmak. Profilen | Offen | 3 | R.2 / S.1 |
| US 8 | Integration von Wireless USB in das USB-Modul | Offen | 3 | R.2 / S.2 |

9.6.3 Inhalt User Stories

| | | |
|--------|-------------------|----|
| 9.6.4 | User Story 1..... | 19 |
| 9.6.5 | User Story 2..... | 20 |
| 9.6.6 | User Story 3..... | 20 |
| 9.6.7 | User Story 4..... | 21 |
| 9.6.8 | User Story 5..... | 22 |
| 9.6.9 | User Story 6..... | 23 |
| 9.6.10 | User Story 7..... | 23 |
| 9.6.11 | User Story 8..... | 24 |

9.6.4 User Story 1

Als Entwickler,
möchte ich eine Simulationsumgebung besitzen,
damit ich die komplexen Vorgänge des Systems und die fertiggestellten Module während der Entwicklung darstellen und testen kann.

9.6.4.1 Erste Akzeptanzkriterien

Wenn die Simulationsumgebung fertiggestellt ist,
dann müssen mindestens die Module für GUI, XML und USB, sowie die Kontrollsoftware einsatzbereit sein.

9.6.4.2 Bemerkungen/Notizen

Die Simulationsumgebung muss enthalten:

- Bereich für die Darstellung der Softwareoberfläche (GUI)
- Simulation der Hardware
- Darstellung der Kommunikation mit der Hardware
- Darstellung der Kommunikation zwischen Kontroll- und Steueroftware
- Möglichkeit zur Simulation verschiedener Szenarien (z.B. Ausfall USB)

9.6.4.3 Story Points/Priorität

Geschätzte Story Points: 24

Priorität: 1

9.6.5 User Story 2

Als Benutzer der Hauptgruppe 1 und 2,
möchte ich eine leicht verständliche Benutzeroberfläche der Software haben,
damit ich Erinnerungen zu Einnahmeterminen erhalte und über Dialoge auf Systemfunktionen und Sonderfälle reagieren kann.

9.6.5.1 Erste Akzeptanzkriterien

Wenn das GUI fertiggestellt ist,
dann muss es in die Simulationsumgebung eingebunden werden
und ausgiebig getestet werden, damit es den Anforderungen entspricht (Kapitel 4.1).

9.6.5.2 Bemerkungen/Notizen

Das grundlegende GUI beinhaltet:

- ein Erinnerungssystem
- Dialog zur Medikamentenausgabe (Kommunikation mit HW-Modul nötig)
- Verschiedene Sichten für Benutzer der Hauptgruppe 1 und 2
 - o Medikamentenausgabe (Hauptgruppe 2)
 - o Eingabe Verordnungsdatensätze und pharmakologische Profile (Vorbereitung der Medikamententherapie) (Hauptgruppe 1)

9.6.5.3 Story Points/Priorität

Geschätzte Story Points: 16

Priorität: 1

9.6.6 User Story 3

Als Benutzer der Hauptgruppe 1 und 2,
möchte ich über das USB-Modul der Software (HcMd-S) mit dem Hardwaremodul kommunizieren können,
damit ich die Hardware bedienen kann (z.B. Medikamentenausgabe).

9.6.6.1 Erste Akzeptanzkriterien

Wenn das USB-Modul fertiggestellt ist,
dann muss es in die Simulationsumgebung eingebunden werden
und ausgiebig getestet werden, damit es den Anforderungen entspricht (Kapitel 4.1).

9.6.6.2 Bemerkungen/Notizen

Das Modul beinhaltet:

- Eingliederung der PHDC Klasse
- Kommunikation: SW -> PHDC -> Hardware und umgekehrt
- Möglichkeit der Überwachung durch Kontrollsoftware

9.6.6.3 Story Points/Priorität

Geschätzte Story Points: 16

Priorität: 1

9.6.7 User Story 4

Als Benutzer der Hauptgruppe 1 und 2,
möchte ich über das XML-Modul der Software (HcMd-S) meine Gabe-Schemata
eingeben beziehungsweise dargestellt haben,
damit ich die Medikamentenausgabe, Erinnerungsfunktion und Eingabe des Gabe-
Schemas durchführen kann.

9.6.7.1 Erste Akzeptanzkriterien

Wenn das XML-Modul fertiggestellt ist,
dann muss es in die Simulationsumgebung eingebunden
und ausgiebig getestet werden, damit es den Anforderungen entspricht (Kapitel
4.1).

9.6.7.2 Bemerkungen/Notizen

Das Modul beinhaltet:

- eine erforderliche Schema-Datei für Verordnungsdatensätze und pharmakologische Profile
- ein defensiv sicheres einlesen/abspeichern der XML-Daten
- Möglichkeit der Überwachung durch Kontrollsoftware
- Schnittstelle für das GUI zum Einlesen und auslesen der XML-Daten (Sicht für Hauptgruppe 1)

9.6.7.3 Story Points/Priorität

Geschätzte Story Points: 16

Priorität: 1

9.6.8 User Story 5

Als Benutzer der Hauptgruppe 1 und 2
möchte ich über eine Kontrollsoftware zusätzliche Sicherheit und Protokolle zur Überprüfung erhalten,
damit ich, neben der Sicherheit, als Benutzer der Hauptgruppe 1 Analysen tätigen kann und gegebenenfalls Anpassungen am Gabe-Schema vornehmen kann.

9.6.8.1 Erste Akzeptanzkriterien

Wenn die Kontrollsoftware fertiggestellt ist,
dann muss sie in die Simulationsumgebung eingebunden
und ausgiebig getestet werden, damit sie den Anforderungen entspricht (Kapitel 4.1).

9.6.8.2 Bemerkungen/Notizen

Laut Architektspezifikation und Anforderungen muss die Kontrollsoftware folgendes leisten:

- Protokollierung und Ausgabe der aufgenommenen Daten als Protokoll
 - o Ausgabezeitpunkte der Medikamente, Erfassung der ausgegebenen Medikamente, besondere Benutzerwünsche¹⁴⁰
- Überwachung der XML- und USB-Schnittstelle
- Überwachung auf Kontrolle der Korrektheit der Gabe-Schemata (Integrität Verordnungsdatensätze, z.B. Kontrolle vor Überdosis)
- Überwachung der besonderen Benutzerwünsche (siehe Protokollierung)
- notfalls Eingreifen bei Entdeckung eines Fehlers und Stoppen des aktuellen Vorgangs
- unabhängig von der Steuersoftware

9.6.8.3 Story Points/Priorität

Geschätzte Story Points: 24

Priorität: 1

¹⁴⁰ Beispielsweise: Benutzer wollte mehr Medikamente des Präparats x am xx.xx.2011 um 19.00 Uhr. Entscheidung: Zusätzliche Ausgabe abgelehnt. Grund: Überdosis von Präparat x entdeckt.

9.6.9 User Story 6

Als Benutzer der Hauptgruppe 2
möchte ich über einen integrierten Terminkalender verfügen,
damit ich meine Termine, speziell zur Medikamententherapie, über das System verwalten kann.

9.6.9.1 Erste Akzeptanzkriterien

Wenn die grundlegenden Funktionen des Erinnerungssystems und die Medikamentenausgabe-Dialoge vorhanden sind,
dann kann ein interaktiver Terminkalender integriert werden.

9.6.9.2 Bemerkungen/Notizen

Der Terminkalender ist nicht *Must Have*.

9.6.9.3 Story Points/Priorität

Geschätzte Story Points: 8

Priorität: 2

9.6.10 User Story 7

Als Benutzer der Hauptgruppe 1
möchte ich über das Internet die Verordnungsdatensätze und pharmakologischen Profile des Systems verwalten können,
damit ich die Medikamententherapie wesentlich schneller anpassen kann.

9.6.10.1 Erste Akzeptanzkriterien

Wenn der erste Release vorhanden ist,
dann kann eine Erweiterung auf das Internet angedacht sein.

9.6.10.2 Bemerkungen/Notizen

Gilt für den nächsten Release.

9.6.10.3 Story Points/Priorität

Geschätzte Story Points: 24

Priorität: 3

9.6.11 User Story 8

Als Benutzer der Hauptgruppe 2
möchte ich die Steuersoftware getrennt von der Hardwarestation bedienen können
(ungebunden: z.B. Verbindung über Wireless USB und Anwendung ist auf iPod),
damit ich nicht immer zur Hardware gehen muss, um zu kontrollieren, ob ein
Einnahmetermin ansteht.

9.6.11.1 Erste Akzeptanzkriterien

Wenn der erste Release vorhanden ist,
dann kann eine Erweiterung des USB-Moduls und der Software auf iPods angedacht
sein.

9.6.11.2 Bemerkungen/Notizen

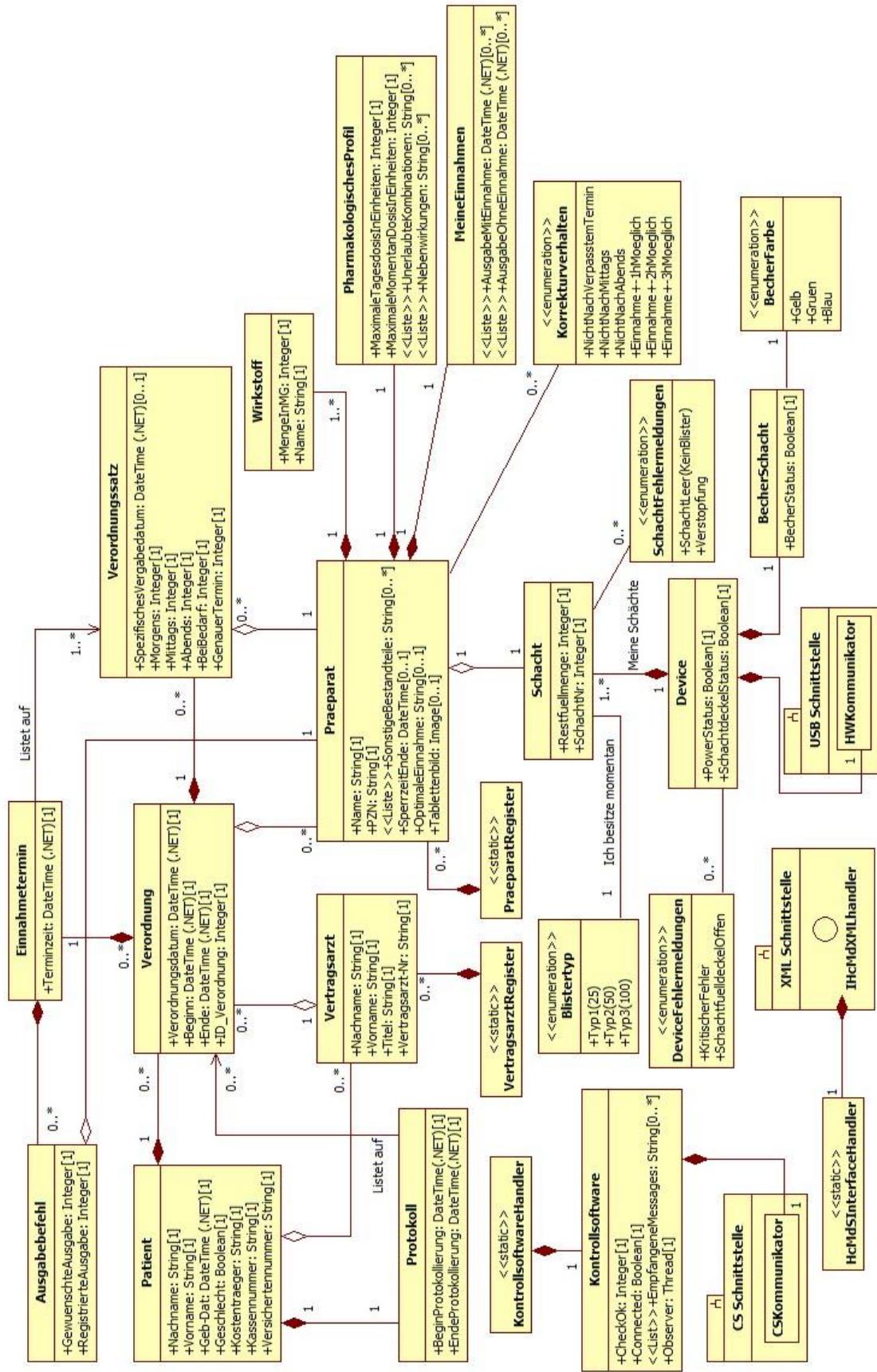
Gilt für den nächsten Release.

9.6.11.3 Story Points/Priorität

Geschätzte Story Points: 48

Priorität: 3

9.7 Anhang 7



9.8 Anhang 8



9.9 Anhang 9

9.9.1 Szenario 1

Aufgabe

Ausgabe der Tabletten zu einem Termin und Einnahme: Der Patient wird vom System aufgefordert, seine Tabletten zu nehmen, welche der Patient daraufhin durch den Medikamentenausgabe-Dialog zu sich nimmt.

Optimale Handlungssequenz

1. Der Patient erkennt, dass Tabletten genommen werden sollen und fängt mit dem Medikamentenausgabe-Dialog an.
2. Der Patient legt den richtigen Becher in das System.
3. Der Patient nimmt die Tabletten zu sich. (wie das System es verlangt)
4. Der Patient bestätigt die Tabletteneinnahme (oder Nicht-Einnahme).

9.9.1.1 Protokollierung Gutachter 1

Handlung 1

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - o Wahrscheinlich ja. Das System blinkt rot auf und gibt einen Ton von sich. Der Benutzer sollte darauf aufmerksam werden.
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - o Wahrscheinlich nein. Das System unterscheidet sich bei dem Tablettenausgabe-Monitor kaum vom Standby-Monitor. Der Benutzer könnte übersehen, dass ein Termin erreicht ist.
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - o Wahrscheinlich nein. Siehe vorherige Frage. Die ähnlichen Monitore könnten für Verwirrung sorgen, sodass der Benutzer nicht mit der Medikamentenausgabe beginnt.
- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - o Wahrscheinlich nein. Gleicher Grund wie bei den vorausgegangenen zwei Fragen. Ich gehe davon aus, dass der Benutzer die Medikamentenausgabe schwer erkennt.

Handlung 2

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - o Wahrscheinlich nein. Der Benutzer wird wahrscheinlich anfangen, einen Becher in das System zu legen. Wenn er keine Vorkenntnisse hat, wird er nicht sofort wissen, was zu tun ist. Erst nach dem Lesen der Aufforderungen (umständlich für die Benutzergruppe).
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - o Wahrscheinlich nein. Er wird aber sehr schwer erkennen, welchen Becher er in das System legen soll. Die Aufforderung dazu ist schwer zu sehen.
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - o Wahrscheinlich ja. Er wird wahrscheinlich alle Becher durchgehen und irgendwann zu seinem Ziel kommen.
- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - o Wahrscheinlich nein. Das System gibt Tabletten aus. Der Benutzer hat aber kein Feedback, wann die Tablettenausgabe fertig ist. Er könnte den Becher zu früh herausnehmen.

Handlung 3

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - o Wahrscheinlich ja. Der Benutzer könnte besondere Einnahmebedingungen übersehen, da sie im GUI untergehen. Wahrscheinlich wird der Benutzer die Tabletten so zu sich nehmen, wie er es gewohnt ist.
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - o Wahrscheinlich ja. Die Aufforderung zur Einnahme ist schlecht zu sehen (gleich wie erste Frage).
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - o Wahrscheinlich nein. Das System sollte den Benutzer ermutigen, die Tabletten zu nehmen (Ermutigung fehlt).
- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - o Wahrscheinlich nein. Das GUI verändert sich mit der Frage der Einnahme. Wieder schwer zu erkennen. Es kann leicht vom Benutzer vergessen werden. Das System sollte darauf aufmerksam machen, dass noch etwas zu erledigen ist.

Handlung 4

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - o Wahrscheinlich ja.
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - o Wahrscheinlich nein. Letzte Frage von Handlung 3 (System soll darauf aufmerksam machen, dass die Einnahmekontrolle aktiv ist)
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - o Wahrscheinlich ja.
- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - o Wahrscheinlich nein. Das System gibt kein Feedback, wenn der Benutzer mit der Eingabe fertig ist. Wahrscheinlich wird er übersehen, dass er den Dialog abgeschlossen hat.

9.9.1.2 Protokollierung Gutachter 2

Handlung 1

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - o Wahrscheinlich ja. Alles ist klar formuliert.
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - o Wahrscheinlich ja.
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - o Wahrscheinlich ja. Hier stimmen die Anweisungen und sie führen klar zur Aktion hin.
- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - o Wahrscheinlich ja.

Handlung 2

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - o Wahrscheinlich ja.
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - o Wahrscheinlich ja. Allein dadurch, dass das Gerät beim richtigen Becher dann die Tabletten hineingibt.
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - o Wahrscheinlich nein.

- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - Wahrscheinlich ja. Da die Tabletten ausgegeben worden sind.

Handlung 3

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - Wahrscheinlich ja.
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - Wahrscheinlich nein. Die Aufforderung ist schlecht zu sehen.
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - Wahrscheinlich ja.
- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - Wahrscheinlich ja.

Handlung 4

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - Wahrscheinlich ja.
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - Wahrscheinlich ja.
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - Wahrscheinlich nein.
- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - Wahrscheinlich nein. Das System gibt kein Feedback aus.

9.9.2 Szenario 2

Aufgabe

Ausgabe von Bedarfsmedikamenten und Einnahme: Der Patient möchte Bedarfsmedikamente ausgegeben haben (beispielsweise bei Kopfschmerzen) und zu sich nehmen. Dazu gibt der Patient die gewünschte Anzahl ein, lässt sich die Tabletten ausgeben und nimmt sie zu sich.

Optimale Handlungssequenz

1. Der Patient drückt auf den Button für Bedarfsmedikamente (Extra Tablette)
2. Der Patient wählt das/die gewünschte/n Medikament/e und deren Anzahl aus.
3. Der Patient erkennt, dass seine Bedarfsmedikamente genommen werden können.
4. Der Patient legt den richtigen Becher in das System.
5. Der Patient nimmt die Tabletten zu sich (wie das System es verlangt)
6. Der Patient bestätigt die Tabletteneinnahme (oder Nicht-Einnahme).

9.9.2.1 Protokollierung Gutachter 1

Handlung 1

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - o Wahrscheinlich ja.
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - o Wahrscheinlich ja.
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - o Wahrscheinlich ja.
- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - o Wahrscheinlich ja. Es kann sein, dass der Benutzer kurzzeitig verwirrt ist, da sich das GUI kaum ändert.

Handlung 2

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - o Wahrscheinlich ja.
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - o Wahrscheinlich ja.
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - o Wahrscheinlich ja.
- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - o Wahrscheinlich nein. Der Benutzer weiß nicht, wo er gerade ist (wie viele Tabletten er noch eingeben muss).

Handlung 3 (ähnlich Szenario 1, Handlung 1)

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - o Wahrscheinlich ja. Der Benutzer ist mit dem vorherigen Tablettendialog durch, er ist aber wahrscheinlich verwirrt und weiß zuerst nicht, was er machen soll.
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - o Wahrscheinlich ja. Aber erst nach einiger Zeit.
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - o Wahrscheinlich ja. Er wird anfangen Becher in das System zu legen.
- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - o Wahrscheinlich ja.

Handlung 4 (ähnlich Szenario 1, Handlung 2)

- Wiederholender Ablauf. Siehe Antworten aus **Szenario 1, Handlung 2**.

Handlung 5 (ähnlich Szenario 1, Handlung 3)

- Wiederholender Ablauf. Siehe Antworten aus **Szenario 1, Handlung 3**.

Handlung 6 (ähnlich Szenario 1, Handlung 4)

- Wiederholender Ablauf. Siehe Antworten aus **Szenario 1, Handlung 4**.

9.9.2.2 Protokollierung Gutachter 2

Handlung 1

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - o Wahrscheinlich nein. Der Benutzer weiß wahrscheinlich nicht, was Bedarfsmedikamente sind.
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - o Wahrscheinlich ja. Da die Anweisungen wieder klar formuliert sind.
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - o Wahrscheinlich ja.
- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - o Wahrscheinlich ja. Das Bild und die Schriftzüge ändern sich klar.

Handlung 2

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - o Wahrscheinlich ja.
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - o Wahrscheinlich ja.
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - o Wahrscheinlich nein. Die Führung ist verwirrend.
- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - o Wahrscheinlich ja.

Handlung 3

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - o Wahrscheinlich ja.
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - o Wahrscheinlich ja.
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - o Wahrscheinlich ja.
- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - o Wahrscheinlich nein.

Handlung 4 (ähnlich Szenario 1, Handlung 2)

- Wiederholender Ablauf. Siehe Antworten aus **Szenario 1, Handlung 2**.

Handlung 5 (ähnlich Szenario 1, Handlung 3)

- Wiederholender Ablauf. Siehe Antworten aus **Szenario 1, Handlung 3**.

Handlung 6 (ähnlich Szenario 1, Handlung 4)

- Wiederholender Ablauf. Siehe Antworten aus **Szenario 1, Handlung 4**.

9.9.3 Szenario 3

Aufgabe

Nachfragen von Informationen zu Termin und Medikament: Der Patient möchte zum nächsten Termin und zu gerade ausgewählten Medikamenten/Bedarfsmedikamenten zusätzliche Informationen erhalten (mindestens zwei Medikamente beim Termin).

Optimale Handlungssequenz

1. Der Patient erkennt den Informationsbutton und wählt den Button.
2. Der Patient erkennt, dass Detailinformationen zum angezeigten Medikament existieren und wählt den „i“-Button.
3. Nach erhaltener Information kehrt der Patient über den „Zurück“-Button zum Ausgangsdialog zurück.
4. Der Patient wählt den „Weiter“-Button, um zum nächsten Medikament zu kommen.
5. Der Patient wählt den „Ende“-Button, um zurück zum Ausgangsdialog zu gelangen.

9.9.3.1 Protokollierung Gutachter 1

Handlung 1

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - o Wahrscheinlich ja. Ein Info-Button (i) ist allgemein bekannt.
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - o Wahrscheinlich ja. Ein Info-Button (i) ist allgemein bekannt.
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - o Wahrscheinlich ja. Ein Info-Button (i) ist allgemein bekannt.

- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - Wahrscheinlich ja.

Handlung 2

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - Wahrscheinlich ja. Jedoch ist der weitere Info-Button verwirrend. (Er ist zwar anders platziert und es stehen Informationen über dem Button, aber er sieht exakt gleich aus.)
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - Wahrscheinlich ja. Jedoch ist der weitere Info-Button verwirrend. (Er ist zwar anders platziert und es stehen Informationen über dem Button, aber er sieht exakt gleich aus.)
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - Wahrscheinlich nein. Jedoch ist der weitere Info-Button verwirrend. (Er ist zwar anders platziert und es stehen Informationen über dem Button, aber er sieht exakt gleich aus.)
- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - Wahrscheinlich ja.

Handlung 3

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - Wahrscheinlich ja.
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - Wahrscheinlich ja.
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - Wahrscheinlich ja.
- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - Wahrscheinlich ja.

Handlung 4

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - o Wahrscheinlich ja.
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - o Wahrscheinlich ja.
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - o Wahrscheinlich ja.
- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - o Wahrscheinlich ja.

Handlung 5

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - o Wahrscheinlich ja.
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - o Wahrscheinlich ja.
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - o Wahrscheinlich ja.
- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - o Wahrscheinlich ja.

9.9.3.2 Protokollierung Gutachter 2

Handlung 1

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - o Wahrscheinlich ja. Der Informationsbutton ist klar gekennzeichnet.
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - o Wahrscheinlich ja.
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - o Wahrscheinlich ja.
- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - o Wahrscheinlich ja. Da die Anzeige umspringt.

Handlung 2

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - o Wahrscheinlich ja.
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - o Wahrscheinlich ja.
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - o Wahrscheinlich ja.
- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - o Wahrscheinlich ja.

Handlung 3

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - o Wahrscheinlich ja, wenn er zurück will.
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - o Wahrscheinlich ja.
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - o Wahrscheinlich ja.
- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - o Wahrscheinlich ja. Er wird merken, dass er wieder da ist, wo er schon einmal war.

Handlung 4

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - o Wahrscheinlich ja.
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - o Wahrscheinlich ja.
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - o Wahrscheinlich ja.
- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - o Wahrscheinlich ja. Er wird erkennen, dass es sich nun um ein anderes Medikament handelt.

Handlung 5

- Wird der Benutzer versuchen, den richtigen Effekt zu erzielen?
 - o Wahrscheinlich ja.
- Wird der Benutzer erkennen, dass die korrekte Aktion zur Verfügung steht?
 - o Wahrscheinlich ja.
- Wird der Benutzer eine Verbindung herstellen zwischen der korrekten Aktion und dem gewünschten Effekt?
 - o Wahrscheinlich ja.
- Wenn die korrekte Aktion ausgeführt worden ist: Wird der Benutzer den Fortschritt erkennen?
 - o Wahrscheinlich ja. Da er wieder zur Startseite zurückgelangt.

9.9.4 Zusammenfassung

Es sind maximal vier positive Antworten auf die Fragen pro Handlung möglich. Die folgende Tabelle zeigt die positiven Antworten der beiden Gutachter zu den Handlungen.

| Szenario / Handlung | Gutachter 1 | Gutachter 2 |
|----------------------|-------------|-------------|
| S1 / H1 | 1 | 4 |
| S1 / H2 | 1 | 3 |
| S1 / H3 | 2 | 3 |
| S1 / H4 | 2 | 2 |
| S2 / H1 | 4 | 3 |
| S2 / H2 | 3 | 3 |
| S2 / H3 | 4 | 3 |
| S2 / H4 | 1 | 3 |
| S2 / H5 | 2 | 3 |
| S2 / H6 | 2 | 2 |
| S3 / H1 | 4 | 4 |
| S3 / H2 | 3 | 4 |
| S3 / H3 | 4 | 4 |
| S3 / H4 | 4 | 4 |
| S3 / H5 | 4 | 4 |
| Positiv/Max. Möglich | 41 / 60 | 49 / 60 |

Tabelle A9 1: Zusammenfassung der Antworten der Gutachter

9.10 Anhang 10

1. Allgemeine Anforderungen

- keine Gefährdung Dritter oder Patienten. Risiken sind nur erlaubt, wenn der Nutzen vertretbar ist.
- Einhaltung der Grundsätze der integrierten Sicherheit unter Berücksichtigung des allgemein anerkannten Standes der Technik.
- bei Lösungsfindung folgende Grundsätze einhalten:
 - Beseitigung oder Minimierung der Risiken.
 - Gegebenenfalls Ergreifen angemessener Schutzmaßnahmen, einschließlich Alarmvorrichtungen gegen nicht zu beseitigende Risiken.
 - Unterrichtung der Benutzer über die Restrisiken, für die keine angemessenen Schutzmaßnahmen getroffen werden können.
- Produkt muss Medizinprodukt nach Art. 1, Abs. 2a sein.
- Sicherheit muss auch nach Verschleiß/Transport gewährleistet sein.
- Nebenwirkungen dürfen keine unvertretbaren Risiken beinhalten.

2. Anforderungen an die Konstruktion

- chem., physikal. und biologische Eigenschaften.
 - möglichst Vermeidung des Einsatzes von toxischen Stoffen.
 - Arzneimittelrichtlinien gelten, wenn Stoff des Produktes als Arzneimittel gilt.
 - Risiken durch Stoffe, die dem Produkt entweichen, müssen maximal verringert werden.
 - unter Berücksichtigung Anwendungsort muss Risiko bei unbeabsichtigtem Eindringen eines Fremdstoffes maximal verringert werden.
- Infektion und mikrobielle Kontamination
- Eigenschaften im Hinblick auf die Konstruktion und die Umgebungsbedingungen.
 - Kombination von Produkten, einschl. Anschlüsse muss sicher sein, Einschränkungen müssen in der Gebrauchsanweisung gekennzeichnet sein.

- Produkte mit Messfunktionen
 - angemessene Genauigkeitsgrenzen müssen eingehalten werden, Genauigkeitsgrenzen sind vom Hersteller anzugeben.
 - Messskalen, Bedienungs- und Anzeigeeinrichtungen müssen ergonomischen Grundsätzen entsprechen.
 - vorgeschriebene Einheiten nach 80/181/EWG sind einzuhalten.
- Schutz vor Strahlen
- Anforderungen an Produkte mit externer oder interner Energiequelle
 - programmierbare Elektroniksysteme: Wiederholbarkeit, Zuverlässigkeit und Leistung, gemäß Zweckbestimmung muss gewährleistet sein.
 - Produkte müssen mit Einrichtungen ausgestattet sein, die jegliche Störung der Mengenregelung, die eine Gefahr darstellen kann, verhindern und/oder signalisieren.
 - Produkte müssen mit Vorrichtungen ausgestattet sein, welche eine unbeabsichtigte gefährlich erhöhte Abgabe von Energie/Stoffen verhindern.
- Bereitstellung von Informationen durch den Hersteller.

9.11 Anhang 11

Ermittelte Anforderungen aus den Kapiteln 1 und 2:

- Kapitel 1.1:
 - Einsetzbar für den HomeCare-Bereich
 - Darstellung von Verordnungen in Verordnungsdatensätzen
 - Kontrollierbarkeit der Verordnungsdatensätze mit Hilfe von pharmakologischen Profilen
 - zugeschnitten auf eine ältere Personengruppe
- Kapitel 1.2:
 - Möglichkeit der Lagerung und einfacher Nachfüllbarkeit der Medikamente
 - Nachweis der Füllmenge der Medikamente
 - Nachweis der Identität der Medikamente
 - Schnittstelle HW <> SW muss gewisse Grundbefehle für die Kommunikation bereitstellen
- Kapitel 2.1:
 - Für die Zulassung gelten das deutsche Recht (MPG) und die EU-Richtlinien (z.B. 93/42/EWG)
- Kapitel 2.2:
 - Medizinische Zweckbestimmung ist erforderlich
 - Zulassungsvorgang nach 93/42/EWG ist erforderlich
 - Testen des Prototyps nur in klinischer Studie zulässig
- Kapitel 2.3:
 - Grenze der Ableitströme (DIN EN 60601-1)
 - Entwicklungsmethodik und Anforderungen an den Entwicklungsprozess (DIN EN 60601-1-4)
 - Sicherheitsklassifizierung und SOUP-Komponenten (DIN EN 62304)
 - Risikomanagement nach DIN EN ISO 14971
 - Anpassung des Entwicklungsprozesses für die Verbesserung und Minimierung der Risiken durch die Gebrauchstauglichkeit (DIN EN 62366)
 - Anforderungen an die Dokumentation (aus sämtlichen Normen)

9.12 Anhang 12

Mit Hilfe der Spezifikationen aus dem PHDC Standard¹⁴¹, ist es möglich geeignete Deskriptoren für den elektronischen Medikamentendispenser zu erstellen. Im kommenden Abschnitt sind sämtliche Deskriptoren für das Device tabellarisch abgebildet. Angefangen beim Device Descriptor, bis hin zu den PHDC-spezifischen Deskriptoren und den Endpoint-Deskriptoren. Es wird ein minimalistischer Deskriptor-Aufbau – mit einer Konfiguration und einem Interface – erstellt. Abbildung A12 1 verdeutlicht dies.

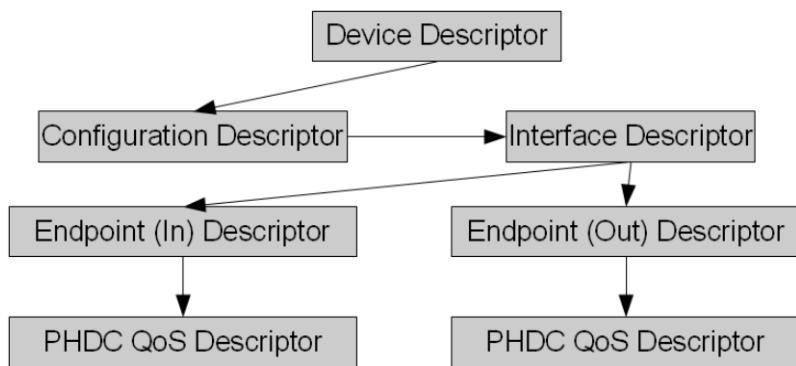


Abbildung A12 1: Descriptor-Aufbau HomeCare Medidisp (selbst erstellt)

Device Descriptor

| Feldname | Offset/Größe | Bsp. Value | Beschreibung |
|-----------------|--------------|------------|---|
| bLength | 0/1 | 0x12 | Länge des Deskriptors in bytes |
| bDescriptorType | 1/1 | 0x01 | Device Descriptor |
| bcdUSB | 2/2 | 0x200 | PHDC unterstützt nur USB 2.0 |
| bDeviceClass | 4/1 | 0x00 | 00h = DCC im Interface Descriptor |
| bDeviceSubClass | 5/1 | 0x00 | Keine Subklassen |
| bDeviceProtocol | 6/1 | 0x00 | Kein spezielles Protokoll |
| bMaxPacketSize0 | 7/1 | 0x64 | Maximale Paketgröße für ein High Speed Device |
| idVendor | 8/2 | 0xFFFF | USB.org registrierte Vendor ID |
| idProduct | 10/2 | 0x0001 | Product ID |

¹⁴¹ USB.org. [\[PHDC2007\]](#)

| | | | |
|--------------------|------|--------|--|
| bcdDevice | 12/2 | 0x0100 | Device-Release-Nummer |
| iManufacturer | 14/1 | 0x00 | Index zum String Array für die Beschreibung des Herstellers |
| iProduct | 15/1 | 0x00 | Index zum String Array für die Beschreibung des Produkts |
| iSerialNumber | 16/1 | 0x00 | Index zum String Array für die Beschreibung der Seriennummer des Devices |
| bNumConfigurations | 17/1 | 0x01 | Anzahl Configuration-Deskriptoren |

Tabelle A12 1: Device Descriptor HomeCare Medidisp (selbst erstellt)

Configuration Descriptor

| Feldname | Offset/Größe | Bsp. Value | Beschreibung |
|---------------------|--------------|---------------|---|
| bLength | 0/1 | 0x09 | Länge des Deskriptors in bytes |
| bDescriptorType | 1/1 | 0x02 | Configuration Descriptor |
| wTotalLength | 2/2 | 0x0019 | Totale Länge aller Interface- und Endpoint-Deskriptoren, inklusive dieses Configuration-Deskriptors |
| bNumInterfaces | 4/1 | 0x01 | Anzahl der Interface-Deskriptoren für diesen Configuration Descriptor |
| bConfigurationValue | 5/1 | 0x01 | Argument, um die Konfiguration auszuwählen |
| iConfiguration | 6/1 | 0x00 | Index zum String Array für die Beschreibung der Konfiguration |
| bmAttributes | 7/1 | 01000000 | Konfigurationen: Self Powered/Kein Remote Wakeup |
| MaxPower | 8/1 | 0x0 | Maximaler Stromverbrauch des Devices mit dieser Konfiguration (1 = 2mA, 32 = 64mA) |

Tabelle A12 2: Configuration Descriptor HomeCare Medidisp (selbst erstellt)

Interface Descriptor

| Feldname | Offset/Größe | Bsp. Value | Beschreibung |
|--------------------|--------------|------------|--|
| bLength | 0/1 | 0x09 | Länge des Deskriptors in bytes |
| bDescriptorType | 1/1 | 0x04 | Interface Descriptor |
| bInterfaceNumber | 2/1 | 0x00 | Index des Interfaces im Interface Array, angefangen bei 0 |
| bAlternateSetting | 3/1 | 0x00 | Identifier für alternative Settings |
| bNumEndpoints | 4/1 | 0x02 | Anzahl der Endpoint-Deskriptoren = 2 |
| bInterfaceClass | 5/1 | 0x0F | USB Device Class Code für PHDC (0Fh) |
| bInterfaceSubClass | 6/1 | 0x00 | Keine Subklasse |
| bInterfaceProtocol | 7/1 | 0x00 | Kein Protokoll |
| iInterface | 8/1 | 0x00 | Index zum String Array für die Beschreibung des Interfaces |

Tabelle A12 3: Interface Descriptor HomeCare Medidisp (selbst erstellt)

PHDC Class Function Descriptor

| Feldname | Offset/Größe | Bsp. Value | Beschreibung |
|-----------------|--------------|------------|--|
| bLength | 0/1 | 0x04 | Länge des Deskriptors in bytes |
| bDescriptorType | 1/1 | 0x20 | PHDC Class Function Descriptor (20h) |
| bPHDCDataCode | 2/1 | 0x01 | Vendor definierter Standard (01h) |
| bmCapability | 3/1 | 00000000 | Meta-Data Message Preamble feature nicht implementiert |

Tabelle A12 4: PHDC Class Function Descriptor für den HomeCare Medidisp (selbst erstellt)

Endpoint- Deskriptoren

Jeweils für In/Out Endpoint. Die Deskriptoren unterscheiden sich nur anhand der Adresse und der Übertragungsrichtung. Es wird eine maximale Paketgröße von 2 MB festgelegt. Diese Größe deckt alle Befehlsketten bei weitem ab. Eventuell könnte man später die Größe noch verringern.

| Feldname | Offset/Größe | Bsp. Value | Beschreibung |
|-----------------|--------------|----------------------|---|
| bLength | 0/1 | 0x07 | Länge des Deskriptors in bytes |
| bDescriptorType | 1/1 | 0x05 | Endpoint Descriptor |
| bEndpointAdress | 2/1 | 10000001 00000010 | Adresse 1, Endpoint IN Adresse 2, Endpoint OUT |
| bbmAttributes | 3/1 | 00000010 | PHDC vorgeschrieben: BULK-Transfertyp |
| wMaxPacketSize | 4/2 | 0x2048 | Maximale Paketgröße für diesen Endpoint |
| bInterval | 6/1 | 0x64 | Abrufintervall in Millisekunden (100ms) |

Tabelle A12 5: Endpoint-Deskriptoren des HomeCare Medidisp (selbst erstellt)

QoS-Deskriptoren

Jeweils für In/Out Endpoint die gleichen Deskriptoren. Es wurde die Konfiguration *Medium/Best* gewählt, da über diese Konfiguration eine Kommunikation in beide Richtungen möglich ist. Die festgelegte Abrufgeschwindigkeit des Endpoints von 100ms fällt in den vorgeschriebenen Rahmen der. Für Steuerbefehle und Antworten ist ein Intervall von 100ms völlig ausreichend.

| Feldname | Offset/Größe | Bsp. Value | Beschreibung |
|---------------------|--------------|------------|----------------------------------|
| bLength | 0/1 | 0x04 | Länge des Deskriptors in bytes |
| bDescriptorType | 1/1 | 0x21 | PHDC QoS Descriptor (21h) |
| bQoSEncodingVersion | 2/1 | 0x01 | Version 1 QoS Encoding |
| bbmAttributes | 3/1 | 00001000 | Konfiguration <i>Medium/Best</i> |

Tabelle A12 6: QoS-Deskriptoren für den HomeCare Medidisp (selbst erstellt)