



Daniel Robinig, BSc

Entwicklung eines Web-Frontends für ein Decision Support System als Medizinprodukt

Masterarbeit

zur Erlangung des akademischen Grades

Diplom-Ingenieur

Masterstudium Biomedical Engineering

eingereicht an der

Technischen Universität Graz

Betreuer

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

Institut

Health Care Engineering

Dipl.-Ing. Dr.techn. Peter Beck

decide Clinical Software GmbH

JOANNEUM RESEARCH Forschungsgesellschaft mbH

Graz, November 2016

Deutsche Fassung:

Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008

Genehmigung des Senates am 1.12.2008

EIDESSTATTLICHE ERKLÄRUNG

AFFIDAVIT

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

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

Datum / Date

Unterschrift / Signature

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

Entwicklung eines Web-Frontends für ein Decision Support System als Medizinprodukt

Decision Support Systeme unterstützen das klinische Personal bei der Entscheidungsfindung bei diagnostischen und therapeutischen Problemen. Das GlucoTab-System dient der Therapieunterstützung von Diabetes Mellitus Typ 2 Patienten im Krankenhaus.

Im Zuge dieser Masterarbeit wird zunächst ein Überblick über Decision Support Systeme zur Therapieunterstützung gegeben. In weiterer Folge wird ein Prototyp für ein Web-Frontend als Erweiterung des bestehenden Entscheidungsunterstützungssystems erstellt. Im Rahmen dieser Entwicklung werden sämtliche Abschnitte des in der EN 62304 beschriebenen Softwareentwicklungsprozesses durchlaufen. Den Abschluss der Entwicklung des Prototyps stellt ein summativer Usability-Test, gemäß EN 62366, dar.

Bei diesem Usability-Test konnten 70,4% der Haupt-Gebrauchstauglichkeitsziele und 80,8% der Teil-Gebrauchstauglichkeitsziele erreicht werden. Deshalb kann die Gebrauchstauglichkeit als „Befriedigend“ bis „Gut“ bewertet werden.

Schlüsselwörter: Therapieunterstützung, Decision Support System, Web-Frontend, Softwareentwicklungsprozess, Usability-Test

Development of a web-frontend for a decision support system as a medical device

Decision support systems assist clinical professionals on decision-making for diagnostic and therapeutic problems. GlucoTab is such a system, with the aim of therapy support during the treatment of Diabetes Mellitus type 2 patients in hospital.

In this thesis an overview of decision support systems for therapy improvement is given. Furthermore a prototype of a web-frontend is developed as an extension of the existing GlucoTab system. Within the development the prototype has to pass through all sections of the software development process as described in EN 62304. To conclude the development of the prototype a usability-test was performed according to EN 62366.

In this usability-test 70.4% of the main usability targets, and 80.8% of the sub targets are passed. This result can be rated as “satisfactory” to “good” usability.

Keywords: therapy support, decision support system, web-frontend, software development process, usability-test

Inhaltsverzeichnis

Abbildungsverzeichnis.....	VIII
Tabellenverzeichnis.....	IX
Abkürzungsverzeichnis.....	X
1 Einleitung.....	11
1.1 Aufgabenstellung.....	12
1.2 Motivation.....	13
1.2.1 Therapie mit dem GlucoTab-System.....	14
Theoretischer Hintergrund.....	16
2 Decision Support Systeme.....	16
2.1 Technische Einteilung.....	16
2.1.1 Look-up Systeme und Dosisberechner.....	17
2.1.2 Guideline basierte Systeme.....	17
2.1.3 Bayesische Systeme.....	17
2.1.4 Systeme mit künstlicher Intelligenz.....	18
2.2 Marktübersicht über Decision Support Systeme zur Therapieunterstützung.....	18
2.2.1 TREAT STEWARD.....	20
2.2.2 EndoTool – Glucose Mangement System.....	21
2.2.3 ATHENA - CDSS.....	22
2.2.4 MediScore CARDSS.....	23
2.2.5 HTN-CDSS.....	25
2.2.6 AiDKlinik.....	26
2.2.7 Expeda OPAD – The Osteoporosis Advisor.....	27
2.2.8 GlucoTab.....	29
3 Rechtliches Umfeld bei der Entwicklung von Software als Medizinprodukt.....	30
3.1 Relevante Normen für die Entwicklung von Software als Medizinprodukt.....	33

3.2	EN ISO 13485-Qualitätsmanagementsysteme	34
3.3	EN 62304 Medizingeräte-Software – Software-Lebenszyklus-Prozesse	34
3.3.1	Software-Entwicklungslebenszyklus als V-Modell	36
3.4	EN ISO 14971 - Anwendung des Risikomanagements auf Medizinprodukte	37
3.5	EN 62366 – Anwendung der Gebrauchstauglichkeit auf Medizinprodukte	39
4	Technischer Hintergrund	41
4.1	Architektur von Webapplikationen	41
4.1.1	Model-View-Controller-Architektur	42
4.2	Entwicklung von Software mit agilen Entwicklungsmethoden	44
4.3	Verwendete Technologien	46
4.3.1	JavaServer Faces (JSF)	46
4.3.2	Primefaces	47
4.3.3	Diagram-Bibliothek Dygraphs	48
4.3.4	Apache Maven	49
5	Entwicklung des Prototyps für das Web-Frontend	50
5.1	Anforderungsanalyse	50
5.1.1	Spezifikation der Anwendung	52
5.1.2	Definition der Hauptbedienfunktionen	53
5.1.3	Definition technischer Rahmenbedingungen auf der Clientseite	55
5.2	Systemarchitektur	57
5.2.1	Gesamtsystem	57
5.2.2	Architektur des Web-Frontend-Prototyps	59
5.2.2.1	View-Komponenten	60
5.2.2.2	Model-Komponenten	61
5.2.3	Schnittstellen zum Backend	65
5.2.4	Shared-UI Bibliothek	66

5.2.4.1	AdjustTherapyController.....	66
5.2.4.2	SharedUIUtils	67
5.2.4.3	SupplementSchemeCalculator.....	67
5.3	Implementierung	68
5.3.1	Implementierung des Blutzuckerlaufdiagramms	69
5.3.2	Internationalisierung.....	70
5.4	Testen der Software	71
	Überprüfung der Gebrauchstauglichkeit	72
6	Heuristische Evaluation.....	73
6.1	Evaluierungsmethodik.....	73
6.2	Benutzerprofile.....	75
6.3	Umfang der Evaluation	75
6.4	Usabilityexperten und Evaluationsumgebungen	75
6.5	Ziele der Evaluation	76
6.6	Ergebnisse und Maßnahmen.....	76
7	Abschließender summativer Usability-Test.....	78
7.1	Vorbereitung des summativen Usability-Tests	78
7.1.1	Testpersonen und Testumgebung	78
7.1.2	Definition der Gebrauchstauglichkeitsziele	80
7.2	Durchführung.....	81
7.2.1	Bewertungsmethodik.....	81
7.3	Ergebnisse Usability-Test und Interview	82
7.3.1	Auswertung nach Gebrauchstauglichkeitszielen	82
7.3.2	Auswertung nach PROMEDIKs quantifizierter Usability	83
7.3.3	Interview der Probanden	83
7.3.4	Identifizierte Gebrauchstauglichkeitsprobleme und Gegenmaßnahmen	84

8	Diskussion	91
9	Schlussfolgerung und Ausblick.....	94
10	Literatur.....	96
11	Anhang.....	102
11.1	UML-Diagramme	102
11.2	Codeausschnitt – Kombination Dygraphs und JSF	103
11.3	Fragebogen und Aufgaben beim Usability-Test	104
11.4	Detailergebnisse heuristische Evaluation.....	112
11.5	Detailergebnisse Usability-Test	112
11.6	Berechnungsbeispiel quantitative Usability	118

Aus Gründen der besseren Lesbarkeit wurde in dieser Diplomarbeit auf eine genderkonforme Schreibweise verzichtet. Sämtliche personenbezogenen Bezeichnungen sind als geschlechtsneutral zu verstehen.

Abbildungsverzeichnis

Abbildung 1: Benutzeroberfläche THREAT Steward [22]	21
Abbildung 2: Architektur des CARDSS [31]	23
Abbildung 3: Architektur von CARDSS Online[35]	25
Abbildung 4: Benutzeroberfläche des OPAD-CDSS[43]	28
Abbildung 5: Entscheidungsbaum für aktive Medizinprodukte aus MedDev 2.4. [51]	31
Abbildung 6: Modell des Software-Entwicklungslebenszyklus[49]	37
Abbildung 7: Zusammenhang des Gebrauchstauglichkeitsorientierten Entwicklungsprozesses (EN62366) und des Risikomanagementprozesses (EN 14971) [56]	40
Abbildung 8 Typische Darstellung der Elemente einer Webanwendung [58]	42
Abbildung 9: Die Model-View-Controller-Architektur	43
Abbildung 10: Der Scrum Prozess [63]	45
Abbildung 11: Früher Mock-Up Prototyp des Web-Frontends	50
Abbildung 12: Weiter fortgeschrittener Entwurf der Benutzeroberfläche	51
Abbildung 13: GlucoTab Gesamtsystem	58
Abbildung 14: Umsetzung der Model-View-Controller Architektur im Prototyp	59
Abbildung 15: Layout der Therapieanpassungsansicht	60
Abbildung 16: Übersicht über die in der Model-Komponente enthaltenen Beans	62
Abbildung 17: Darstellung des Korrekturschemas im Prototyp	67
Abbildung 18: Prototyp des BZ-Verlaufsdigramms und der Verabreichungshistorie	69
Abbildung 19: BZ-Verlaufsdigramm und Historie der Medikamentengaben	70
Abbildung 20: Aufteilung der identifizierten Programmfehler auf einzelne Komponenten des Prototyps	77
Abbildung 21: Aufteilung der Identifizierten Usabilityschwächen nach den untersuchten Heuristiken	78
Abbildung 22: Identifizierte Usabilityprobleme in Abhängigkeit der Probandenanzahl nach Nielsen [74]	79
Abbildung 23: Prozentuelle Verteilung der Bedienfehler in der PROMEDIKs Auswertung	84
Abbildung 24: Identifizierte Bedienfehler während des Usability-Tests	85
Abbildung 25: Hervorhebung des schwer zu identifizierenden Messfrequenzparameters	86

Abbildung 26: Unterschiedliche Darstellung der Insuline in der eigenen und der Basis/Bolus-Therapie..... 89

Abbildung 27: UML-Diagramm zur Darstellung der Abhängigkeiten der Beans im Frontend 102

Abbildung 28: UML-Diagramm des AdjustTherapyHandlers 102

Abbildung 29: UML-Diagramm von SharedUIUtils 102

Abbildung 30: UML-Diagramm des SupplementSchemeCalculators 103

Tabellenverzeichnis

Tabelle 1: Vergleich von CDSS zur Therapieunterstützung..... 19

Tabelle 2: Spezifikation des Web-Frontend-Prototyps 52

Tabelle 3: Bei der Evaluation verwendete Heuristiken, angelehnt an [73] 74

Tabelle 4: Soziodemographische Daten der Probanden der heuristischen Evaluation..... 76

Tabelle 5: Anteil der gefundenen Usabilityprobleme in Abhängigkeit von der Probandenanzahl in Prozent [75] 80

Tabelle 6: Soziodemographische Daten der Probanden..... 80

Tabelle 7: Drei-Stufen-Bewertungsverfahren als Bewertungsmaßstab für Usability-Evaluationen nach Backhaus [72] 82

Tabelle 8: Auswertung der Gebrauchstauglichkeitsziele 83

Tabelle 9: Definition der Aufgaben und Durchführungsschritte für den Benutzertest 111

Tabelle 10: Auswertung der Gebrauchstauglichkeit nach PROMEDIKS..... 112

Tabelle 11: Detaillierergebnisse bei der Auswertung nach Gebrauchstauglichkeitszielen 115

Abkürzungsverzeichnis

AJAX	Asynchronous JavaScript and XML
ATC	Anatomisch-Therapeutisch-Chemisches Klassifikationssystem
CDSS	Entscheidungsunterstützungssystem (Clinical-decision-support-system)
CSS	Cascading Style Sheets
DM	Diabetes Mellitus
EON	Protokoll-basierte Architektur für Entscheidungsunterstützungssysteme
FDA	Food and Drug Administration
JSF	JavaServer Faces
JSP	JavaServer Pages
KIS	Krankenhausinformationssystem
MDD	Medical Device Directive (Medizinprodukterichtlinie)
MDR	Medical Device Regulation (Medizinprodukteverordnung)
MVC	Model-View-Controller Architektur
NM	Netzwerk
OAD	Orale Antidiabetika
PHP	PHP: Hypertext Processor
QI	Qualitätsverbesserung (Quality Improvement)
QMS	Qualitätsmanagementsystem
SOAP	Simple Object Access Protocol
SOUP	Software of Unknown Provenance
UML	Unified Modeling Language

1 Einleitung

Diabetes Mellitus (DM) ist eine der häufigsten Stoffwechselerkrankungen der Welt. Sie wird durch Störungen der Insulinproduktion, der Wirkung des Insulins, oder einer Kombination der Beiden ausgelöst. Diese Störungen führen zu einem erhöhten Blutzuckerspiegel (Hyperglykämie) [1]. Im Jahr 2013 litten weltweit ca. 382 Millionen Menschen unter Diabetes, für das Jahr 2035 wird die Anzahl an Diabeteserkrankten auf 592 Millionen prognostiziert [2]. In Österreich wurde die Anzahl der Diabetiker im Jahr 2013 auf rund 573.000 bis 645.000 Menschen geschätzt [3]. Die häufigsten Formen von DM werden in zwei Hauptgruppen eingeteilt: DM Typ 1 und DM Typ 2.

Patienten mit DM Typ 1 ist es nicht möglich den Blutzuckerspiegel (BZ) selbst zu steuern, da die insulinproduzierenden β -Zellen in der Bauchspeicheldrüse durch eine Autoimmunerkrankung zerstört wurden. Weitere Ursachen für eine DM Typ 1 Erkrankung stellen Erkrankungen der Bauchspeicheldrüse oder ein Verlust der Bauchspeicheldrüse dar. Diese Patienten müssen Insulin immer über Insulinpräparate zuführen. [3]

90 – 95 % aller Diabetespatienten sind an DM Typ 2 erkrankt [1][3][4]. Bei DM Typ 2 Patienten ist die Wirkung des Insulins im Körper eingeschränkt, es herrscht eine sogenannte Insulinresistenz vor, welche zu einem erhöhten Insulinbedarf zur Senkung des BZ führt. Als Hauptursache für DM Typ 2 ist Übergewicht der Betroffenen zu nennen. Die erste Therapiemaßnahme für DM Typ 2 ist eine Änderung der Lebensweise zur Reduktion des Körpergewichts und des Körperfettanteiles. In weiterer Folge wird die Gabe von Metformin empfohlen und ab einem gewissen Erkrankungsgrad muss auch hier Insulin über Insulinpräparate zugeführt werden. [1] Bei der medikamentösen Behandlung von DM Typ 2 Patienten im Krankenhaus, empfehlen klinische Leitlinien bevorzugt die Verwendung von Insulin zur Regulierung des Blutzuckerspiegels.

Diabetestherapie ist ein komplexes Thema, mit Einflussfaktoren aus der Lebensweise und den aktuellen medizinischen Parametern des Patienten. Durch diese Komplexität kommt es zu Medikationsfehlern und einer daraus resultierenden Gefährdung des Patienten. Beim „National Diabetes Inpatient Audit 2015“ wurde in 38,3 % der Medikationen von Diabetikern im Krankenhaus, zumindest ein diabetesbezogener Medikationsfehler in den vorherigen sieben Tagen gefunden [5]. In [6] wurden Fehler in der manuellen Insulin-Dosis Berechnung

als größte Quelle von Medikationsfehlern identifiziert. Durch die Verwendung von computergestützten Entscheidungsunterstützungssystemen (CDSS) können gerade diese Berechnungsfehler ausgeschlossen werden und in weiterer Folge komplexere Berechnungsalgorithmen zur Berücksichtigung von mehreren Einflussfaktoren verwendet werden. Aktuell zielen unterschiedliche CDSS auf die Verbesserung des Therapieprozesses für Diabeteserkrankte im Krankenhaus (*Endotool*, Monarch Medical Technologies, Charlotte NC, USA; *eGlycemic Management System*, Glytec Systems, Waltham MA, USA; *GlucoTab*, Joanneum Research, Graz, Österreich) ab.

Das GlucoTab-System ist ein Stand-Alone CDSS zur Unterstützung des klinischen Personals beim Workflow der Behandlung von DM Typ 2 Patienten während des Krankenhausaufenthaltes. Zum Zeitpunkt der Masterarbeit ist das System am Landeskrankenhaus-Universitätsklinikum Graz in Verwendung. In den Kapiteln 1.2.1 und 2.2.8 wird das System ausführlicher beschrieben.

Ärzte verbringen im klinischen Alltag ungefähr doppelt so viel Zeit am Schreibtisch wie beim Patienten [7]. Da das GlucoTab-System nur über eine Benutzeroberfläche am Tablet verfügt, würde eine Benutzerschnittstelle für den PC die Verwendung des Systems am Schreibtischarbeitsplatz des Arztes wesentlich vereinfachen.

1.1 Aufgabenstellung

Im Rahmen dieser Masterarbeit soll ein Prototyp eines Web-Frontends als Erweiterung des bestehenden GlucoTab®-Systems (Joanneum Research, Graz, Österreich) nach den Qualitätsstandards für Softwaremedizinprodukte (IEC 62366, IEC 62304, ISO EN 14971) soweit entwickelt werden, dass die Durchführung eines summativen Usability-Tests möglich ist. Dieser Usability-Test stellt die Validierung und den Abschluss des praktischen Teils der Masterarbeit dar. Der Prototyp soll die folgenden Hauptfunktionen der GlucoTab®-Software bieten: Zum einen soll das klinische Personal bei der Therapieplanung von DM Typ-2 Patienten unterstützt werden, indem die klinische Dokumentation erleichtert wird und BZ-Messwerte sowie die Ernährung und diabetesrelevante Medikation visualisiert werden. Zum anderen soll die Software eine automatisierte Empfehlung für die Insulintagesdosis sowie eine Empfehlung der Verteilung der Insulintagesdosis auf jede einzelne Insulingabe liefern.

In diesem Prototyp sollen die von Ärzten verwendeten Funktionen wie die Therapieanpassung und Medikamentenverordnung implementiert werden, die die Pflege betreffenden Funktionen werden nicht umgesetzt da bei der Anwendung dieser die Mobilität des Tablets von Vorteil ist. Die Tablet-gebundene Benutzeroberfläche des GlucoTab®-Systems soll für die Anwendung am Computer übertragen und angepasst werden. Dies führt zur besseren Integration des Systems in den klinischen Workflow, da das System direkt über das Krankenhausinformationssystem aufgerufen und ohne zusätzliches Gerät direkt am Computerarbeitsplatz im Krankenhaus verwendet werden soll. Während der Entwicklung des Web-Frontends wird der gesamte Entwicklungsprozess medizinischer Software durchlaufen. Dieser Entwicklungsprozess soll mit den einzelnen Schritten der Planung, Implementierung und des Testens dokumentiert werden.

Weiters soll das rechtliche Umfeld und der normative Rahmen für die Entwicklung von Software als Medizinprodukt mit besonderem Augenmerk auf die Schlüsselnormen für die Zertifizierung beleuchtet werden. Abschließend soll durch eine Literatur- und Marktrecherche die aktuelle Marktsituation für klinische Decision Support Systeme zur Therapieunterstützung erfasst werden. Hierbei sollen nicht nur Softwaresysteme für die Unterstützung bei der Therapie von DM Patienten betrachtet werden, sondern ein breiterer Überblick über unterschiedliche Systeme zur Therapieunterstützung geschaffen werden.

1.2 Motivation

Ursprünglich war es geplant ein Webinterface für die Tätigkeiten des Arztes in der zu Beginn der Masterarbeit aktuellen Softwareversion 3.0 des GlucoTab-Systems zu erstellen. Diese Version umfasste das Workflow-Management für die Basis/Bolus Therapie, nicht jedoch die Möglichkeit der Therapiedokumentation für individuelle Therapieformen.

Im Rahmen der Weiterentwicklung der Android-Benutzerschnittstelle des GlucoTab-Systems, näherte sich jedoch die Darstellung der Basis/Bolus-Therapie immer näher an die geplante Darstellung der eigenen Therapie an. Die einzelnen Therapieformen werden in 1.2.1 näher erläutert. Dies sollte dem Benutzer eine konsistente Darstellung der Therapiedokumentation, unabhängig von der gewählten Therapieart, gewährleisten. Dadurch sind nur geringe Unterschiede bei der Bedienung des Systems durch das klinische Personal bei unterschiedlicher Therapiewahl gegeben. Durch den Vergleich der Benutzerschnittstelle mit

Systemen zur Medikamentenverordnung wurde letztendlich eine virtuelle Fieberkurve für die Tätigkeit der Medikamentenverordnung gewählt. Diese Art der Darstellung ähnelt der aktuell gängigen Insulintherapie-Dokumentation auf Papier, und sollte somit die Umstellung vom Papier zur digitalen Oberfläche erleichtern.

Durch die starke Annäherung der Benutzerschnittstelle bei der eigenen und der Basis/Bolus-Therapieform wurde entschieden, dass auch das im Rahmen dieser Masterarbeit zu entwickelnde System beide Therapieformen und alle den Arzt betreffenden Features der nächsten Version 4.0 umfassen sollte. Dies stellte eine maßgebliche Änderung der Zielsetzung des Prototyps dar.

Durch die parallele Entwicklung der eigenen Therapieform für beide Benutzerschnittstellen, soll es möglich sein die Änderungsanfragen der Benutzer, im Rahmen der wöchentlichen Treffen mit dem klinischen Personal der Medizinischen Universität Graz, in beide Systeme einfließen zu lassen. Die Definition der Anforderungen an die Benutzerschnittstellen wird in Kapitel 5.1.1 detaillierter ausgeführt.

1.2.1 Therapie mit dem GlucoTab-System

Die Therapie mit dem GlucoTab-System basiert auf einer Guideline zur Diabetesbehandlung. Durch die Umsetzung der Benutzeroberfläche auf einem Android-Tablet kann das System vom klinischen Personal direkt am Point-of-Care verwendet werden. Dadurch kann, gerade bei Aufgaben des Pflegepersonals, zusätzlich zur Dosisberechnung und Therapieplanung der Workflow der Therapie direkt am Patienten unterstützt werden. Grundsätzlich werden vom System zwei Therapiearten unterstützt:

Basis/Bolus-Therapie: Hier wird die Kombination von einem langwirksamen Basal-Insulin mit einem kurzwirksamen Bolus-Insulin zur Einstellung des BZ verwendet. Für die Berechnung der Insulin-Dosen müssen vier BZ-Messungen durchgeführt werden. Das Pflegepersonal wird über Alarme und Erinnerungen bei der Durchführung dieser Messungen unterstützt. Täglich sind vier Gaben des Bolus-Insulins und eine des Basal-Insulins vorgesehen, auch hier stellt das System die Workflowunterstützung bereit. Bei jeder Gabe des Bolus-Insulins wird basierend auf der aktuellen Blutzuckermessung eine Neuberechnung der zu verabreichenden Dosis durchgeführt. Deshalb ist vor jeder Bolus-Gabe eine BZ-Messung essentiell. Mittags wird die

Basal-Insulin Gabe durchgeführt. Die Tagesdosis wird durch den Algorithmus täglich vor Mittag angepasst. Bei der Dosisberechnung werden folgende Parameter in Betracht gezogen:

Alter, Gewicht und Kreatininwert: Bei der Berechnung der Initialen Tagesdosis

Insulinkorrekturschema: Bei jeder Bolus-Insulingabe, zur Berücksichtigung von Insulinresistenzen des Patienten

Eingenommene Mahlzeit: Diese Information wird zur Anpassung der Insulindosis bei der Verabreichung herangezogen

BZ-Messerwert, vorherige Tagesdosis: Bei jeder Berechnung einer neuen Tagesdosis, außer der Berechnung der initialen Tagesdosis, bilden die BZ-Messerwerte und die vorherige Tagesdosis die Basis. Beide Werte werden auch bei der Anpassung der Insulindosis bei jeder Insulingabe herangezogen.

Eigene Therapie: Mit der eigenen Therapie ist es dem Benutzer möglich, jede beliebige DM Typ 2-Therapie zu dokumentieren. So kann zum Beispiel die Vortherapie des Patienten weitergeführt werden. Ähnlich wie in der Basis/Bolus-Therapie wird der Benutzer an die Medikamentengabe und an geplante Blutzuckermessungen erinnert. Bei der Verabreichung eines Bolus-Insulins findet eine Unterstützung in Form einer Dosisanpassung statt. In diese Dosisberechnung werden sowohl das verordnete **Korrekturschema**, eine **eingenommene Mahlzeit** und andere **verabreichte Bolus-Insuline** miteinbezogen.

Theoretischer Hintergrund

2 Decision Support Systeme

Klinische Decision Support Systeme (CDSS) stellen Ärzten, Therapeuten und anderen Personen im Gesundheitswesen auf jeden Patienten maßgeschneiderte Informationen zur Entscheidungsunterstützung bei diagnostischen oder therapeutischen Problemen zur Verfügung. Ziel dieser Systeme ist es, die unterstützte medizinische Handlung sicherer, effizienter und ökonomischer zu machen [8].

Grundsätzlich gibt es mehrere Möglichkeiten der Kategorisierung von CDSS. Einerseits basierend auf ihrem Bestimmungszweck [9], wie zur Prävention, zur Unterstützung der Diagnose, der akuten Therapie oder bei der Nachbehandlung sowie zur Verbesserung der Gesundheitsökonomie. Manche CDSS werden auch bei der Ausbildung des klinischen Personals eingesetzt, ein Beispiel dafür wäre *DXplain* (Massachusetts General Hospital Laboratory of Computer Science, Boston, USA).

Andererseits kann man sie basierend auf dem technischen Hintergrund der Entscheidungsunterstützung einteilen. Auch hier gibt es unterschiedlichste Möglichkeiten der Einteilung, alleine dadurch bedingt ob man die Systeme nach der Funktion ihres Frontends oder nach der technischen Lösung des Backends kategorisiert. In [10] werden die Funktionen der CDSS-Frontends in die folgenden Kategorien eingeteilt:

- a) Medikamentendosisberechnung
- b) Point-of-Care Alarme, Erinnerungen
- c) Darstellung Relevanter Zusatzinformationen
- d) Experten Systeme
- e) Workflow-Unterstützung

Diese Kategorisierung der Funktionen wird in Tabelle 1 wiederverwendet.

2.1 Technische Einteilung

Alther und Reddy teilen CDSS in zwei Gruppen ein [11]. In Wissensbasierte (Knowledge Based) und nicht-Wissensbasierte (Nonknowledge-Based) CDSS. Während die Erstgenannten auf klinischen Regeln, evidenzbasiertem Wissen und Berechnungsalgorithmen sowie

vordefinierten Entscheidungsbäumen basieren, greifen Zweitere auf durch maschinelles Lernen erstellte künstliche neuronale Netze und evolutionäre Algorithmen zurück. [11]

Die unterschiedlichen Architekturen der Backends der CDSS lassen sich jedoch in mehr als zwei Gruppen einteilen. Gerade die in den letzten Jahrzehnten stetig gestiegene Rechenleistung von Computersystemen ermöglicht die Entwicklung von CDSS mit immer komplexeren Berechnungsmodulen und Algorithmen. So lassen sich CDSS Systeme, mit steigender Komplexität, in folgende Kategorien einteilen [12]:

2.1.1 Look-up Systeme und Dosisberechner

Dies sind einfache Systeme, welche leicht zu validieren sind. Sie können bei Patienten, welche gut und stabil auf die Behandlung ansprechen, verwendet werden. Sie benötigen wenig Rechenleistung und können auch rein auf mobilen Geräten betrieben werden. An der Stanford University, Stanford, USA, wurde ein CDSS System zur Optimierung der Verwendung von Bluttransfusionen entwickelt [13]. Durch die Einführung von Benachrichtigungen an den Benutzer, bei Verordnung einer unnötigen Transfusion, konnte eine Reduktion der verwendeten Bluttransfusionen um 24% erzielt werden. Weiters könnte man das in 2.2.2 behandelte EndoTool® – Glucose Management Systems (Monarch Medical Technologies, Charlotte, USA) als Dosisberechner bezeichnen.

2.1.2 Guideline basierte Systeme

In Situationen, in denen mehrere unterschiedliche Informationen in die Entscheidungsunterstützung einfließen und es zu unterschiedlichen Therapiemöglichkeiten kommen kann, werden häufig Guideline basierte Systeme eingesetzt. Diese Systeme setzen komplexere Entscheidungsbäume und Regeln um als Look-up Systeme. Die Ergebnisse sind jedoch noch validierbar und deren Ermittlung benötigt wenig Rechenleistung. In diese Kategorie lassen sich das in [14] beschriebene System zur Behandlung von Morbus Crohn als auch der Algorithmus des GlucoTab Systems [15][16], das HTN-DSS und das ATHENA-System einordnen. Die drei letztgenannten werden in der Marktübersicht in Kapitel 2.2 genauer behandelt.

2.1.3 Bayesische Systeme

Bayesische Systeme sind sehr berechnungsintensiv. Häufig werden sie in Cloud basierten Systemen betrieben. Sie benötigen eine kürzere Lernphase als Systeme mit künstlicher Intelligenz und stellen individuellere Therapieansätze als Dosisberechner und logikbasierte

Systeme zur Verfügung. Die Entscheidungsunterstützung des CDSS TREAT STEWARD (TREAT Systems, Aalborg, Dänemark) basiert auf einem bayesischen Netzwerk [17].

2.1.4 Systeme mit künstlicher Intelligenz

Sie sind berechnungsintensiver als die zuvor genannten Systeme und basieren meist auf künstlichen neuronalen Netzwerken. Da das Ergebnis dieser Systeme auf dem Training des Systems basiert, ist es nicht möglich diese Systeme zu validieren. Sie erscheinen dem Benutzer als Blackbox, dies führt dazu, dass der Einsatz dieser Systeme, trotz ihrer Verfügbarkeit, schlecht akzeptiert ist [18].

Es werden große Eingabedatenmengen benötigt um solche Systeme zu trainieren. Die Entscheidungsvorschläge dieser Systeme müssen streng auf Plausibilität geprüft werden. Diese Systeme werden für komplexe Probleme eingesetzt. Bei der Erstellung des CDSS OPAD - The Osteoporosis Advisor (Expeda, Reykjavik, Island) wurde ein neuronales Netz verwendet, dieses System wird in 2.2.7 beleuchtet.

2.2 Marktübersicht über Decision Support Systeme zur Therapieunterstützung

In diesem Kapitel wird versucht einen Überblick über einige aktuell kommerziell verfügbare CDSS zu geben. Da es sich bei dem GlucoTab-System um ein Therapieunterstützungssystem für die Diabetestherapie handelt wird an dieser Stelle ein Fokus auf andere therapieunterstützende CDSS gelegt. Hierbei soll ein breiter Überblick über Systeme, für die Verwendung bei unterschiedlichen medizinischen Indikationen, und mit verschiedenen technischen Lösungen gegeben werden. Die Systeme werden in Tabelle 1, unter Anlehnung an die in [19] definierten Merkmale, zum Vergleich von CDSS, miteinander verglichen und in weiterer Folge kurz beschrieben.

Tabelle 1: Vergleich von CDSS zur Therapieunterstützung

CDSS-System	TREAT	EndoTool	ATHENA	CARDSS	HTN-CDSS	AiDKlinik	OPAD	Glucotab
Medizinische Indikation	Bakterielle Infektion	Erkrankung Zuckerstoffwechsel	Hypertonie, chronische Schmerzen	Rehabilitation nach Herzinfarkt	Hypertonie	Allgemeine Medikation	Osteoporose	DM Typ 2
Benutzer	Arzt	Arzt, Pflege	Arzt	Arzt	Arzt	Arzt	Arzt	Arzt, Pflege
Frontend	KIS-Integration, WebApplikation	KIS-Integration Stand Alone	KIS-Integration	KIS-Integration	Stand Alone	KIS-Integration, Stand Alone	Stand Alone	(KIS-Integration) Stand Alone
Funktion (vgl. S 15).	d, c,	a, e	a, c, d	c, d,	a, c, d,	c,d	c,d,e	a, b, e
Backend	Bayesisches Netzwerk	Ca. 60. Algorithmen	Guideline	Guideline	Guideline	Algorithmus Guideline	Guideline neuronal. NW	Algorithmus Guideline
Status	Kommerziell	Kommerziell 180+ KA	Klinische Studien	Kommerziell 40+ KA	Open Source	kommerziell	kommerziell	Kommerziell, Pilot in 3 KA
Unterstützung	Antibiotikawahl, Kostenanalyse	Dosisberechnung	Medikation Therapieplanung	Rehabilitationsplanung	Medikation, Lebensstil	Medikamenten-Interaktionen, Kostenanalyse	Behandlung, Lebensstil	Dosisberechnung Workflow
Nachgewiesener Nutzen	~	+	~	+	+	+	~	+
Kommerzieller Nutzen	+	k.A.	k.A.	k.A.	Behandlung - Nutzen +	+	k.A.	k.A.

k.A. ... Keine Angaben, + ... Nachgewiesener Nutzen gegenüber Papierbasierter/konventioneller Therapie, ~...eingehaltene Best -Practice, jedoch kein Vergleich mit konventioneller Therapie/ keine Verbesserung gegenüber konventioneller Therapie

2.2.1 TREAT STEWARD

TREAT STEWARD dient zur Unterstützung der Antibiotikabehandlung von Patienten mit bakteriellen Infektionen im Krankenhaus. Das Ziel ist es sowohl den Einsatz von Antibiotika, in Hinblick auf die steigende Bedrohung der Arzneimittelresistenzen, zu optimieren, als auch die Krankenhausaufenthalte der Patienten zu verkürzen. Den Arzneimittelresistenzen soll durch die Verabreichung einer speziell auf die infektiösen Mikroorganismen angepassten Medikation und die Verringerung der Gabe von Breitbandantibiotika entgegengewirkt werden. Insgesamt soll es durch diese Effekte zu einer Kostenersparnis durch die Verwendung von TREAT STEWARD kommen.

In der Forschungsarbeit von Leibovici et. al. [17] wird zunächst ein allgemein gehaltenes bayesisches Netzwerk zur Behandlung von bakteriellen Infektionen eingeführt, um dann ein auf Infektionen des Urinaltraktes spezialisiertes Netzwerk vorzustellen. Basierend auf dieser Arbeit entstand das CDSS TREAT STEWARD. Als Eingabeparameter dienen sowohl demographische Parameter des Patienten, allgemeine Vitalwerte also auch auf die Infektion bezogene Laborwerte. Ein Überblick über die Eingabewerte in der Benutzeroberfläche befindet sich in Abbildung 1. Neben dem Fokus auf die effizienteste medizinische Behandlung der häufigsten bakteriellen Infektionen wird auch der wirtschaftliche Faktor der Behandlung in die Entscheidungsunterstützung miteinbezogen. In diese Kostenberechnung werden neben den Kosten der Medikamente und der Behandlung auch die Kosten von zu erwartenden Nebenwirkungen, sowie die Kosten der Induktion einer Antibiotikaresistenz miteinbezogen. Diese Kostenberechnung der Resistenz bezieht sowohl die Folgekosten für den behandelten Patienten, als auch die Kosten die durch die zukünftige Behandlung von Patienten mit Infektionen des resistenten Erregerstammes entstehen mit ein.[20]

2013 wurden an 15 Abteilungen in drei Krankenanstalten, in Deutschland, Italien und Israel, zwei Studien, sowohl eine prospektive als auch eine retrospektive durchgeführt. Die Behandlung von 1683 Patienten in Israel (860 CDSS, 823 Kontrollgruppe), wurde in Hinblick auf die 180-Tage-Überlebensrate betrachtet. Diese Studien konnten eine geringe Verbesserung des Behandlungsergebnisses zeigen (eine Verbesserung der 180 Tage Überlebensrate von 71% auf 74% in der prospektiven Studie (keine Statistische Signifikanz, $p=0,2$), und von 71% auf 77% in der retrospektiven ($p=0,04$)). Diese wenn auch nur geringe

Verbesserung wird auf die verbesserte Dokumentation des Krankheits- und Behandlungsverlaufes zurückgeführt. [21]

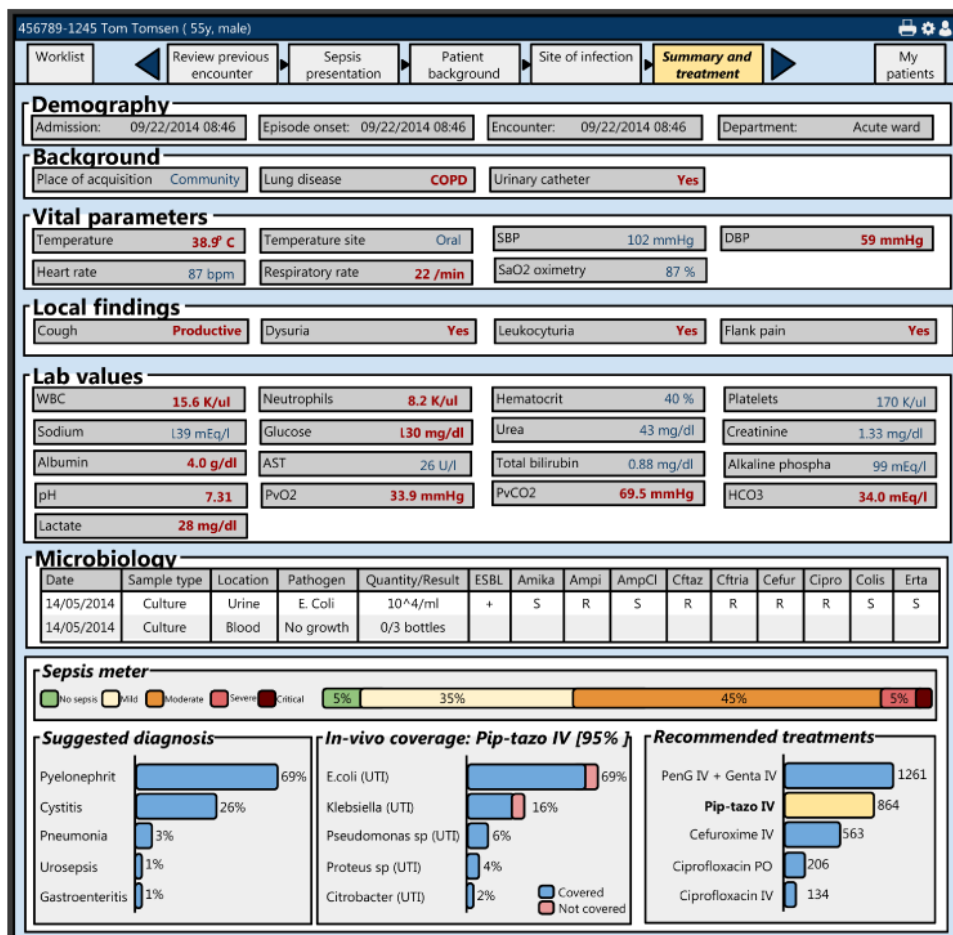


Abbildung 1: Benutzeroberfläche THREAT Steward [22]

Insgesamt wurde in acht Studien im Zeitraum von 2005 bis 2014 eine durchschnittliche Kostenersparnis von 36% durch Verwendung des Systems ermittelt [22].

2.2.2 EndoTool – Glucose Mangement System

Die Entwicklung von EndoTool startete im Jahr 2003 durch P. Bruggess. Das System nützt unterschiedliche Algorithmen zur Dosisberechnung für die subkutane und intravenöse Verabreichung von Insulin, hauptsächlich bei Patienten auf Intensivstationen. Ziel ist die Reduzierung der Medikationsfehler sowohl bei DM Typ 1, DM Typ 2 als auch anderen Erkrankungen des Zuckerstoffwechsels. Aktuell werden von Monarch Medical Technologies, Charlotte, USA drei unterschiedliche Varianten des CDSS angeboten welche, in den USA, alle als FDA Class II Medizinprodukte zertifiziert sind. Auf der Webseite des Herstellers wird mit dem Einsatz in über 180 Spitälern in den USA geworben. [23]

Bei zwei voneinander unabhängigen, auf Intensivstationen durchgeführten, Studien [24], [25] zur Effizienz des CDSS konnte die verbesserte Einstellung des BZ-Spiegels gezeigt werden:

Bei einer randomisierten Kontrollstudie mit 300 Patienten (141 mit CDSS, 159 mit papierbasierter Dosisberechnung) konnten die Patienten mit CDSS länger in BZ-Zielbereich gehalten werden (70,4% [SD, 15,2%] vs. 61,6% [SD, 17,9%], $P < 0,001$). Ebenso konnte die Varianz der gemessenen BZ-Werte geringer gehalten werden (35,5 [SD, 18,3] mg/dL vs. 42,3 [SD, 21,2] mg/dL, $P < 0,001$). [24]

Im Zeitraum von Jänner 2012 bis Oktober 2013 wurde eine Kohortenstudie, mit 759 Patienten, durchgeführt. Im Rahmen dieser Studie wurde beobachtet dass der mittlere BZ-Spiegel bei der Gruppe mit Einsatz des CDSS niedriger ist als bei der ohne (141,5 mg/dl vs. 159,9 mg/dl, $p < 0,0001$). Auch die Variabilität der gemessenen Blutzuckerwerte ist bei der Gruppe mit CDSS geringer. Die Wahrscheinlichkeit einer leichten Hypoglykämie war jedoch bei der CDSS-Gruppe höher als bei der Kontrollgruppe (5,65 hypoglykämische Messwerte/100 Personen-Protokoll Tage vs. 3,43/100 Personen-Protokoll Tage). [25]

Insgesamt erscheint dieser Dosisberechner als ausgereiftes Produkt, mit Einsatz in zahlreichen Krankenhäusern und nachgewiesenem Nutzen. Das System kann in mehrere KIS integriert werden. Vertrieben wird das System von Monarch Medical Technologies in den USA.[23]

2.2.3 ATHENA - CDSS

ATHENA (**A**ssessment and **T**reatment in **H**ealthcare: **E**vidence-based **A**utomation) wurde 2002 als Guideline-basiertes CDSS für die Behandlung arterieller Hypertonie entwickelt. Das System sollte die Kontrolle des Blutdrucks verbessern und bei der Wahl der Medikation von Begleiterkrankungen unterstützen. ATHENA-DSS greift auf die Patientendaten der elektronischen Gesundheitsakte zu und beurteilt diese Informationen anhand der in der Wissensbasis umgesetzten klinischen Leitlinien durch Verwendung des EON – Guideline-Interpreters. Dieser Interpretierer erstellt patientenspezifische Therapieempfehlungen, welche an der graphischen Benutzeroberfläche der elektronischen Gesundheitsakte angezeigt werden. Dieses System kann durch die Anpassung der Wissensbasis auf lokale Anforderungen und Problemstellungen angepasst werden. [26]

Um die Sicherheit und Effizienz der Behandlung von chronischen Schmerzen mit opiat-basierten Schmerzmitteln zu verbessern wurde beginnend im Jahr 2004 basierend auf dem

ATHENA DSS ein CDSS für diesen Anwendungsfall entwickelt. [27][28] In einer Studie mit ursprünglich 44 Probanden, von denen letztendlich 16 befragt wurden wurde das System von allen Probanden als informativ und entscheidungsunterstützend bezeichnet. Zehn der Probanden sahen den Einsatz des Systems als potentiell risikovermindernd an. [29]

Von Jänner 2014 bis Jänner 2017 wird eine Studie durch das San Francisco Veterans Administration Medical Center durchgeführt, um die Wirksamkeit, Akzeptanz und Gebrauchstauglichkeit des Systems zu untersuchen.[30]

2.2.4 MediScore CARDSS

Dieses Guideline-basierte CDSS wurde am *Department of Medical Informatics am Academic Medical Center* der Universität Amsterdam entwickelt und wird von Itémedicale, Tiel, Niederlande vertrieben. Im Jahr 2008 wurde es an über 40 ambulanten Rehabilitationskliniken, für die Therapie nach Herzinfarkten sowie nach Eingriffen am Herzen, in den Niederlanden eingesetzt. [31]

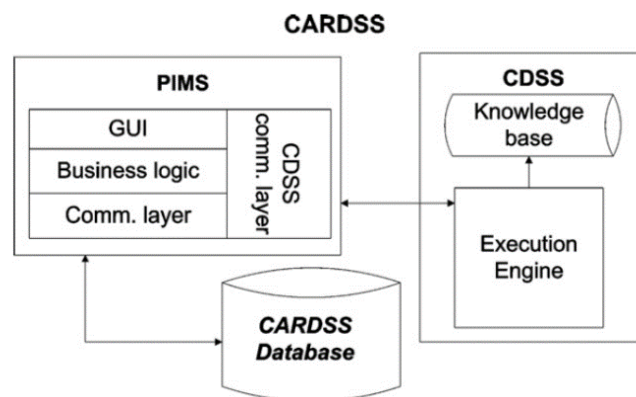


Abbildung 2: Architektur des CARDSS [31]

Für die Entwicklung des CDSS wurde das 1996, durch eine Kooperation der technischen Universität Eindhoven und der Universität Maastricht, entwickelte CDSS-Entwicklungs-Framework GASTON als Grundlage herangezogen und die fehlenden Funktionalitäten entsprechend erweitert. Das CARDSS besteht aus dem CDSS-System an sich, dem Host-Krankenhausinformationssystem, welches die grafische Benutzeroberfläche bereitstellt und der Datenbank in welcher sowohl die klinischen als auch für das CDSS relevanten Daten gespeichert werden (siehe Abbildung 2). Das System unterstützt den Benutzer:

- Bei der Festlegung der Rehabilitationsziele
- Bei der Eingabe und Speicherung der Patientendaten
- Bei der Evaluation des Rehabilitationsfortschritts
- Bei der Analyse von allgemeinen Rehabilitations-, patienten-, und Populationsdatensätzen
- Durch die Bereitstellung von Berichten und Präsentationen, wie der Zusammenfassung der Untersuchungen
- Durch die Bereitstellung von Hintergrundinformationen auf welchen die Empfehlungen des Systems begründet werden können

Ein Prototyp des CARDSS wurde 2003 in einer zweimonatigen Pilotstudie an vier unterschiedlichen Ambulatorien mit 134 Patienten durchgeführt. Während dieser Pilotstudie konnte die grundsätzliche Akzeptanz durch die Benutzer, sowie die einfache Integration des Systems in den klinischen Workflow gezeigt werden [32].

Die Daten von 2787 Patienten aus 21 Kliniken wurden im Jahr 2009 in einer Cluster-randomisierten Studie ausgewertet. Durch Verwendung des DSS die Übereinstimmung der Behandlung mit der Guideline-Empfehlung für Bewegungstherapie um 7,9% (95% Konfidenzintervall 0,1% bis 5,2%), für die Patientenaufklärung um 25,7% (95% Konfidenzintervall 15,5% bis 29,4%) und bei der Entspannungstherapie um 25,5% (25,2% bis 51,3%) verbessert werden. [33]

Im Jahr 2010 wurde vor und nach dem Redesign der Benutzeroberfläche von CARDSS ein Usability-Test mit sieben Probanden durchgeführt. Im Rahmen dieses Thinking-Aloud Tests konnten von den Probanden nach dem Redesign 76% der Hauptaufgaben zur Dateneingabe und 79% der Teilaufgaben (vor Redesign 31% der Hauptaufgaben, 76% der Teilaufgaben) erfüllen. Auch bei anderen Aufgaben konnte eine Verbesserung der Gebrauchstauglichkeit festgestellt werden, durch die geringe Sample-Size war diese Verbesserung nicht statistisch signifikant ($p = 0,7362$; $p = 0,1445$ und $p = 0,5129$).[34]

Das System wurde 2012 um die webbasierte Applikation CARDSS Online erweitert. Die entwickelte Webanwendung soll die stetige systematische und lokale Steigerung der Qualität (Quality Improvement (QI)) des Systems begünstigen. Die Steigerung der Qualität soll durch Einführung eines Plan-Do-Study-Act-Zyklus ermöglicht werden. Die Architektur des Systems

wurde wie in Abbildung 3 ersichtlich um Komponenten für die Erfassung des Feedbacks und des QI-Plans der lokalen QI-Teams erweitert. [35][36]

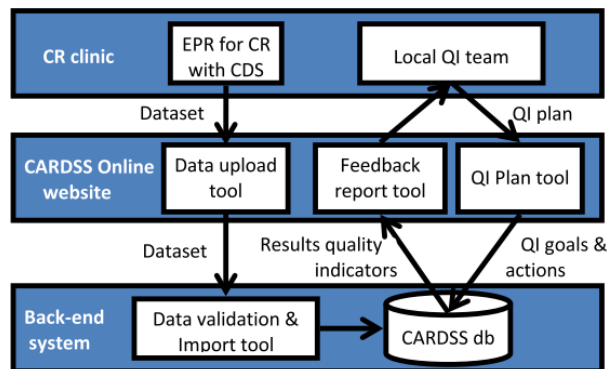


Abbildung 3: Architektur von CARDSS Online[35]

Zur Beurteilung der Verbesserung der Behandlung durch CARDSS Online wurde 2014 eine Studie mit 18 Kliniken geplant [36], die Ergebnisse dieser Studie waren im November 2016 noch nicht veröffentlicht.

2.2.5 HTN-CDSS

Zur Verbesserung der Behandlung von arterieller Hypertonie wurde in Indien, beginnend im Jahr 2011, ein CDSS entwickelt. Dieses System gibt Empfehlungen zur medikamentösen Behandlung von Hypertonie. Weiters evaluiert es, ob aus der Hypertonie eine weitere Herz-Kreislaufkrankung entstehen könnte und gibt Empfehlungen zur Verbesserung des Lebensstils der Patienten. Die Wissensgrundlage für die, vom System bereitgestellten, Empfehlungen stellen die „Indian Hypertension II“-Richtlinien aus dem Jahr 2007 dar. [37]

Die Anforderungen an das System wurden gemeinsam mit 34 Ärzten aus der Region Mahabubnagar, Indien entwickelt. Auf Basis dieser Anforderungen wurde der hauptsächlich auf IF-THEN-Anweisungen basierte Algorithmus von Data Template, Bangalore, Indien entwickelt. Ein erster Pilot-Test, bei welchem die Empfehlungen zur Behandlung von 60 Hypertonie-Patienten mit von Medizinern abgegebenen Behandlungsvorschlägen ergab eine Übereinstimmung pro Empfehlungsfunktion zwischen 83,33% und 90,16%. [37] In dieser Literaturstelle wurde weiters festgehalten, dass das System nur in der primären Basisgesundheitsversorgung eingesetzt werden sollte, da keine Warnungen für Medikamenteninteraktionen bei empfohlenen Medikamenten abgegeben werden und der einfache Algorithmus nur für die Anwendung in diesem Bereich geeignet ist. Auch die

Ähnlichkeit mit dem zu Beginn des Kapitels 2.2.3 erwähnten ATHENA System für die Behandlung von Hypertonie wird erwähnt.

Um die Effizienz und Kosteneffizienz des Systems zu untersuchen wurde eine randomisierte Kontrollstudie, in 8 Clustern mit insgesamt 1628 Patienten, durchgeführt. Davon waren 845 in der CDSS Gruppe und 783 in der Kontrollgruppe. Untersucht wurde der systolische Blutdruck zum Startpunkt der Behandlung und 12 Monate später. Als zweites Ziel wurde die Bewertung der Kosteneffizienz des Systems nach 12 Monaten herangezogen. Für die Evaluation der Kosteneffizienz wurden die während der Therapie verwendeten Ressourcen in beiden Gruppen analysiert. Beim Vergleich des systolischen Blutdrucks konnte eine durchschnittliche Senkung des Blutdrucks um -7,11 mm Hg (95% CI: -2,23 bis -13,11 mm Hg; P=0.008) bei der CDSS Gruppe gegenüber der Kontrollgruppe gezeigt werden. Beim Vergleich der Kosten der Behandlung zwischen den Gruppen wurden bei der CDSS Gruppe Kosten von 370,48\$, pro Patient pro Jahr, gegenüber 344,06\$ bei der Kontrollgruppe. Durch Einsatz des CDSS wurden die direkten Behandlungskosten also um durchschnittlich 26,42\$ pro Patient erhöht. Vergleicht man jedoch die Kosten der Behandlung, pro gesenktem mm Hg, so ergibt sich, wegen der signifikant stärkeren Verringerung des Blutdrucks in der CDSS Gruppe, ein Wert von 36,57 \$/mm Hg für CDSS- und 96,01 \$/mm Hg für die Kontrollgruppe.[38]

2.2.6 AiDKlinik

AiDKlinik stellte ursprünglich ein Medikamenteninformationssystem dar. Dieses System wurde an der Universitätsklinik Heidelberg entwickelt. Um Fehler bei der Medikamentenverordnung zu reduzieren wurde das System um die Funktionen eines elektronischen Arzneimittelverordnungssystems erweitert. Zusätzlich wurden noch entscheidungsunterstützende Funktionen für die Interaktion von rund 8000 Medikamenten untereinander, und zur Kostenoptimierung der Therapie bei der Erstellung von Entlassungsbriefen, implementiert.[39][40] Aktuell verwendet AiDKlinik die vollständige Pharmaindex Datenbank der Medizinischen Medien Informations GmbH, Neu-Isenburg, Deutschland[41].

Von Jänner bis März 2006 wurden 4846 elektronische Medikamentenverordnungen analysiert. Von diesen Verordnungen wurden 43% in weniger als 1,5 Minuten erstellt, zwei Drittel in weniger als 3 Minuten. Bei 178 Verordnungen wurden potentiell schwerwiegende

Medikamenteninteraktionen festgestellt, von welchen 135 bei der Verordnung vermieden wurden (-24,2%, $p=0.016$). Bei der Analyse von 66 Entlassungsbriefen konnte eine durchschnittliche Ersparnis von 31,05€ pro Brief durch den Vorschlag der Verwendung entsprechender Generika erzielt werden.[40]

Hecht et. al. [41] untersuchten 2015 drei CDSS für die Medikamentenverordnung zur Schmerztherapie. 97 Pharmakotherapien wurden mit AiDKlinik analysiert, die Analyse einer Therapie dauerte im Durchschnitt 1:59 Minuten, im Vergleich mit 2:32 Minuten und 1:58 Minuten bei den anderen untersuchten CDSS. In 11.7% der Therapien konnten schwerwiegende Medikamenteninteraktionen festgestellt werden. Bei der Befragung der Mediziner im Anschluss an die Untersuchung wurde AiDKlinik in Bezug auf die einfache und schnelle Medikamentensuche gelobt und die Informationen bezüglich der Medikamenteninteraktionen als sinnvoll erachtet. Als Nachteile wurden die nur kommerzielle Verfügbarkeit und die zeitintensive Eingabe der Verordnungsdaten genannt. Insgesamt wurde das System gegenüber den anderen bevorzugt.

Seit 2006 wird das System durch die Dosing GmbH, Heidelberg, Deutschland weiterentwickelt und vertrieben. Es ist als Medizinprodukt der Klasse IIa zertifiziert. AiDKlinik wird sowohl als Standalone-Lösung als auch in das ICM-KIS von Dräger, Lübeck, Deutschland, integriert vertrieben.[42]

2.2.7 Expeda OPAD – The Osteoporosis Advisor

Am Universitätsklinikum Reykjavik wurde ein Expertensystem zur Diagnose und Behandlung von Osteoporose entwickelt. Als Eingabewerte dienen klinisch relevante Parameter, aus welchen die Wahrscheinlichkeit einer durch Osteoporose ausgelösten Knochenfraktur in den nächsten zehn Jahren, Lebensstil und Behandlungsempfehlungen sowie die Empfehlung für den Zeitpunkt der nächsten Knochendichtemessung ermittelt werden. Weiters wird eine Risikogruppe bezogen auf die Frakturwahrscheinlichkeit ermittelt, sowie die Diagnose von Osteopenia, Osteoporose und manifestierter Osteoporose gestellt. Die Benutzeroberfläche von OPAD mit exemplarischen Eingabewerten und dem dazugehörigen Entscheidungsunterstützungsergebnis befindet sich in Abbildung 4. [43]

Abbildung 4: Benutzeroberfläche des OPAD-CDSS[43]

Die Entscheidungsfindung basiert auf einem Knowledge-Mapping-Ansatz. Experten wurden zu den klinisch relevanten Parametern, welche für die Empfehlung der Behandlungsmethode von Osteoporose eingesetzt werden können, befragt. Aus diesen Parametern durch Verwendung eines neuronalen Netzwerks ein Lookup-Table erstellt. Zur Erstellung der Behandlungsempfehlungen befolgt das OPAD-System sowohl internationale als auch nationale Richtlinien zur Behandlung von Osteoporose. Bei nicht von den Richtlinien abgedeckten Erkrankungsfällen vertraut das System auf die zuvor beschriebene Wissensdatenbank welche aus den Daten der Experten erstellt wurde.

Das System wurde zuerst durch die Verifikation der Behandlungsempfehlungen von 300 virtuellen Patienten getestet. Anschließend wurde ab 1. Jänner 2012 eine Studie mit 308 Patienten, 39 Männern und 269 Frauen, an der Osteoporose-Abteilung der Universitätsklinik Reykjavik durchgeführt. Die von OPAD ermittelte 10-Jahres Knochenfrakturwahrscheinlichkeit für diese Patientengruppe wurde mit der des *WHO Fracture Risk Assessment Tool* verglichen. Bei Verwendung des aktuellen Knochendichtemesswertes wurde eine Korrelation von $r = 0.988$ mit einer mittleren Abweichung von 0,7678% ($SD = 1,9946\%$) ermittelt. Bei einer Berechnung vor der Knochendichtemessung ohne Verwendung eines Messergebnisses ergab sich eine Korrelation von $r = 0,977$, mit mittlerer Abweichung von 1,8285% ($SD = 2,8\%$). Dieses Ergebnis zeigt die hohe Übereinstimmung bei der Berechnung der Knochenfrakturwahrscheinlichkeit mit der Guideline. Die ausführliche Beschreibung der Diagnose soll dem klinischen Personal die Interpretation des berechneten Wertes erleichtern und so

Behandlungsfehlern vorbeugen.[43] Das System wurde im Juli 2012 mit dem CE-Zeichen gekennzeichnet. 2016 wurde ein Patent für das System beim US-Patentamt eingereicht [44]. Vertrieben wird das System von Expeda, Reykjavik, Island [45].

2.2.8 GlucoTab

Der erste Prototyp des GlucoTab-systems wurde im Jahr 2011 im Rahmen des REACTION Projekts entwickelt[46]. Die Entwicklungsarbeit für das System wurde von Joanneum Research, Graz, Österreich und der Medizinischen Universität Graz durchgeführt. Das System dient zur Therapieunterstützung bei der Behandlung von DM Typ 2 im Krankenhaus. Eine genauere Ausführung über die Funktionen des Systems ist in Kapitel 1.2.11.2 zu finden. Grundsätzlich kann zusammengefasst werden, dass das System über eine auf einem Algorithmus basierte Dosisberechnung und eine Guideline-basierte Workflowunterstützung verfügt. Durch die Entwicklung eines Webfrontends soll die Integration des Systems in KIS ermöglicht werden. Vertrieben wird das System von decide Clinical Software GmbH, Graz, Österreich. In den folgenden beiden Studien wurde ein Nutzen des Systems untersucht:

In einer nicht-kontrollierten Interventionsstudie, mit 99 DM Typ 2 Patienten auf vier Abteilungen, wurde die Wirksamkeit, Sicherheit und Gebrauchstauglichkeit des Systems untersucht. Im Rahmen dieser Studie konnten die Patienten signifikant länger im BZ-Zielbereich eingestellt werden als in einer Best-Practice Studie (50,2% vs. 42%, $p=0,001$). Es konnte keine schwerwiegende Hypoglykämie beobachtet werden, bei 1,9% der Blutzuckermessungen konnten jedoch leichte Hypoglykämien festgestellt werden. 91% des in der Studie teilgenommenen klinischen Personals fühlen sich im Umgang mit dem System sicher, 89% fanden das System praktisch in der Anwendung. Allgemein wurde ein positives Feedback zur Gebrauchstauglichkeit abgegeben.[47]

Um den Einfluss des Systems auf die Fehlervermeidung bei der Algorithmus-basierten Insulin Dosisberechnung zu beobachten wurde an der Medizinischen Universität Graz eine Studie mit 79 Patienten durchgeführt. Die Fehlerrate bei der Dokumentation des Blutzuckerspiegels war in beiden Gruppen mit 4,9% für die Gruppe mit Papierbasierte Dosisberechnung und 4,0% bei der CDSS Gruppe sehr ähnlich. Bei der manuellen Dosisberechnung traten in 11,1% der Berechnungen Fehler auf. Die Anzahl der von Berechnungsfehlern beeinflussten BZ-Werte war in der Gruppe mit manueller Berechnung acht Mal so hoch wie in der CDSS Gruppe. [6]

3 Rechtliches Umfeld bei der Entwicklung von Software als Medizinprodukt

In diesem Kapitel soll das rechtliche und normative Umfeld für die Entwicklung von Software als Medizinprodukt beleuchtet werden. Zunächst muss festgelegt werden wann die Software als Medizinprodukt anzusehen ist. Dazu legt die Richtlinie 93/42/EWG fest:

„medizinisches Gerät‘: alle einzeln oder miteinander verbunden verwendete/n Instrumente, Apparate, Vorrichtungen, Software, Stoffe oder anderen Gegenstände samt der Zubehörteile, einschließlich der vom Hersteller speziell zur Anwendung für diagnostische und/oder therapeutische Zwecke bestimmten und für ein einwandfreies Funktionieren des medizinischen Geräts eingesetzten Software, die vom Hersteller zur Anwendung für Menschen für folgende Zwecke bestimmt sind:

- *Erkennung, Verhütung, Überwachung, Behandlung oder Linderung von Krankheiten,*
- *Erkennung, Überwachung, Behandlung, Linderung oder Kompensierung von Verletzungen oder Behinderungen,*
- *Untersuchung, Ersatz oder Veränderung des anatomischen Aufbaus oder eines physiologischen Vorgangs,*
- *Empfängnisregelung,*

und deren bestimmungsgemäße Hauptwirkung im oder am menschlichen Körper weder durch pharmakologische oder immunologische Mittel noch metabolisch erreicht wird, deren Wirkungsweise aber durch solche Mittel unterstützt werden kann.“ [48]

Die Norm EN 62304 [49] definiert Medizinprodukte-Software als Software-System, welches zur Integration in ein anderes Medizinprodukt entwickelt wurde, oder als Software welche für den unabhängigen Gebrauch als Medizinprodukt bestimmt wurde. Medizinprodukte-Software fällt unter dieselben Richtlinien wie andere Medizinprodukte und muss die darin genannten grundlegenden Anforderungen erfüllen. Um die Sicherheit und Wirksamkeit des Medizinproduktes sicherzustellen ist es notwendig festzuhalten, dass die Software ihren Bestimmungszweck ohne das hervorrufen von inakzeptablen Risiken erfüllt [49]. Das

Europäische Komitee veröffentlichte drei EU Richtlinien welche den rechtlichen Rahmen für die Entwicklung von Medizinprodukten festlegen.

- 93/42/EWG Allgemeine Medizinprodukte
- 90/385/EWG Aktive implantierbare Medizinprodukte
- 98/79/EG Medizinprodukte zur In-vitro-Diagnose

Diese EU-Richtlinien müssen innerhalb der EU von den Mitgliedsstaaten in der nationalen Gesetzgebung umgesetzt werden. Dies erfolgt in Österreich durch das Medizinproduktegesetz (MPG BGBl Nr. 657). Betrachtet man die Direktiven mit dem Fokus auf Software als Medizinprodukt so gilt eigenständige Software immer als aktives Medizinprodukt, wenn ihr vom Hersteller eine medizinische Zweckbestimmung zugewiesen wurde [50]. Zur Klassifizierung von Medizinprodukten hat die Europäische Kommission zwei Guideline-Dokumente ausgegeben, die MedDev 2.4 für die Klassifizierung von allen Medizinprodukten und die MedDev 2.1-6 für die Klassifizierung von Software im Speziellen. Sollte die Software ein anderes Medizinprodukt beeinflussen oder steuern ist sie in der gleichen Klasse einzuordnen. Da Software als aktives Medizinprodukt anzusehen ist, ist der Entscheidungsbaum für aktive Medizinprodukte anzuwenden (Siehe Abbildung 5).

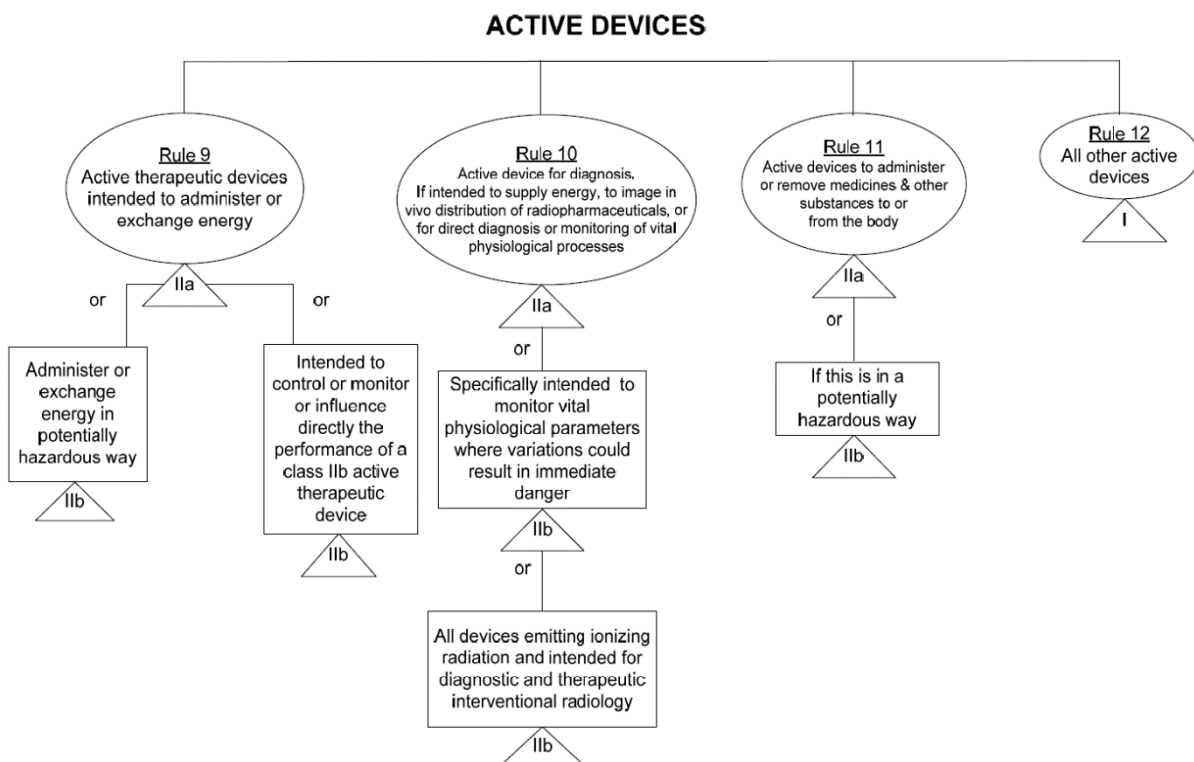


Abbildung 5: Entscheidungsbaum für aktive Medizinprodukte aus MedDev 2.4. [51]

Wendet man diese Regeln auf das GlucoTab®-System an um die Medizinproduktklasse zu ermitteln, so treffen die Regeln 9 bis 11 nicht auf das System zu, da es weder zur direkten Energiezufuhr noch zur direkten Medikamenten-, bzw. Substanzen-gabe vorgesehen ist. Auch dient das System nicht zur Diagnose einer Erkrankung, es darf nur bei Patienten mit diagnostizierter DM Typ 2 verwendet werden. Daraus folgt, dass die Regel 12 angewandt werden muss und das System ein Medizinprodukt der Klasse I ist. Da das System auch kein anderes Medizinprodukt steuert und es mit keinem anderen Medizinprodukt verbunden wird bleibt die Einteilung in diese Klasse bestehen. Im am 15. Juni 2016 bestätigten Vorschlag für die, die Medizinproduktedirektive ablösende, Medizinprodukteverordnung gibt es folgende neue Klassifizierungsregel:

“Rule 10a

Software intended to provide information which is used to take decisions with diagnosis or therapeutic purposes, is in class IIa, except if such decisions have an impact that may directly or indirectly cause:

- the death or an irreversible deterioration of the state of health, in which case it is in class III;*
- a serious deterioration of the state of health or a surgical intervention, in which case it is in class IIb.*

Software intended to monitor physiological processes is in class IIa, except if it is intended for monitoring of vital physiological parameters, where the nature of variations is such that it could result in immediate danger to the patient, in which case it is in class IIb.”[52]

Bei Beurteilung der Medizinproduktklassifizierung nach dieser Regel wäre das GlucoTab-System ein Medizinprodukt der Klasse III. Da laut EN 62304 für das GlucoTab-System die Sicherheitsklasse C anzuwenden ist, wird die Dokumentation zum aktuellen Zeitpunkt mit vollem Umfang erstellt.

3.1 Relevante Normen für die Entwicklung von Software als Medizinprodukt

Die grundlegenden Anforderungen an Medizinprodukte, welche Software enthalten, sind im Anhang 1 der Richtlinie 93/42/EWG unter Abschnitt 12 wie folgend definiert:

*Bei Produkten, die Software enthalten oder bei denen es sich um medizinische Software an sich handelt, muss die Software entsprechend dem Stand der Technik validiert werden, wobei die Grundsätze des **Software-Lebenszyklus**, des **Risikomanagements**, der **Validierung** und **Verifizierung** zu berücksichtigen sind. [50]*

In Artikel 5 der 93/42/EWG gehen die Mitgliedstaaten davon aus, dass die oben genannten grundlegenden Anforderungen erfüllt sind wenn das Medizinprodukt anhand der harmonisierten Normen entwickelt wurde. Die EN 62304 beschreibt die geforderten Software-Lebenszyklusmodelle. Sie fordert weiter die Einrichtung eines Risikomanagementsystems nach EN ISO 14971 und empfiehlt ein Qualitätsmanagementsystem nach EN ISO 13485. Da die grundlegenden Anforderungen an alle Medizinprodukte zusätzlich noch die Gebrauchstauglichkeit umfassen wird dieser Normenkatalog noch um die Normen EN 60601-1 und um die EN 62366, welche die EN 60601-1 in weiterer Folge ersetzt. Die folgenden Normen werden also für die Entwicklung von medizinischer Software als relevant angesehen:

- EN ISO 13485: Qualitätsmanagementsysteme
- IEC 62304: Software Lebenszyklus
- ISO 14971: Risikomanagement
- EN 62366 und EN 60601-1: Gebrauchstauglichkeit

Diese Normen werden in den nun folgenden Unterkapiteln genauer beleuchtet, stellen jedoch nicht zwingend alle erforderlichen Normen für die Entwicklung jeglicher medizinischer Software dar, der Normenkatalog muss gegebenenfalls in Abhängigkeit der Zweckbestimmung um relevante Normen für die spezifische Anwendung erweitert werden.

Aktuell befindet sich die Norm ISO 82304 für „Health Software“ in Entwicklung. Diese Norm soll unter anderem die Validierung von Stand-Alone Software beschreiben, dieser Bereich war im bisherigen normativen Umfeld noch nicht abgedeckt.

3.2 EN ISO 13485-Qualitätsmanagementsysteme

Der Hersteller von Medizinprodukte-Software muss beweisen, dass er fähig ist Software als Medizinprodukt herzustellen und sowohl die anwendbaren regulatorischen Anforderungen als auch die Kundenanforderungen dauerhaft zu erfüllen [49]. Diese Fähigkeiten können durch die Anwendung eines Qualitätsmanagementsystems erwiesen werden. Die Anforderungen an ein solches Qualitätsmanagementsystem werden in der Norm EN ISO 13485 festgelegt. Als Grundlage dieser Norm dient die ISO 9001:2000. Um die Anforderungen der EN ISO 13485 zu erfüllen muss der Hersteller:

- Die für das QMS erforderlichen Prozesse und die Anwendung dieser in der Organisation erkennen.
- Die Reihenfolge und den Zusammenhang zwischen den Prozessen festlegen
- Erforderliche Kriterien und Methoden bestimmen, um die wirksame Durchführung und Steuerung dieser Prozesse sicherzustellen
- Die benötigten Ressourcen und Maßnahmen zur Durchführung und Dokumentation dieser Prozesse bereitstellen
- Die Prozesse erfassen, messen und zu analysieren
- Benötigte Maßnahmen durchführen um die geplanten Ergebnisse zu erreichen und die Wirksamkeit der Prozesse aufrecht zu erhalten.

Diese Prozesse müssen vom Hersteller durch die Erstellung von Qualitätszielen, einem Qualitätsmanagementhandbuch sowie Dokumenten die vom Hersteller zur Sicherstellung der wirksamen Planung, Durchführung und Steuerung der Entwicklungsprozesse benötigt werden, dokumentiert werden.[53]

3.3 EN 62304 Medizingeräte-Software – Software-Lebenszyklus-Prozesse

Die EN 62304 [49] stellt den Softwarelebenszyklusprozess vor, welcher den sicheren Entwurf und die sichere Wartung von Software sicherstellen soll. Sie fordert in den allgemeinen Anforderungen die Verwendung eines Qualitätsmanagementsystems welches der ISO 13485 oder einer nationalen Norm für Qualitätsmanagementsysteme bzw. ein Qualitätsmanagementsystem welches von einer nationalen Vorschrift gefordert wird. Weiters

fordert die Norm die Anwendung eines Risikomanagementprozesses nach ISO 14971. Diese beiden Normen werden in den Kapiteln 3.2 und 3.4 genauer betrachtet.

Nach der EN 62304 hat der Hersteller jedem Software-System eine Software-Sicherheitsklasse zuzuordnen. Diese Sicherheitsklasse beschreibt die Schwere der möglichen Gefährdung eines Patienten oder Anwenders. Die Norm unterscheidet drei Sicherheitsklassen:

Klasse A: Keine Verletzung oder Schädigung der Gesundheit ist möglich.

Klasse B: Keine schwere Verletzung ist möglich.

Klasse C: Tod oder schwere Verletzung ist möglich.

Die Sicherheitsklasse wirkt sich in weiterer Folge auf die normativen Anforderungen während des Softwarelebenszyklus aus. Die Norm benennt die folgenden Prozesse als Teil des Lebenszyklusmodells für Software:

- Software-Entwicklungs-Prozess
- Software-Wartungs-Prozess
- Software-Risikomanagement-Prozess
- Software-Konfigurationsmanagement-Prozess
- Problemlösungsprozess für Software

Zunächst spezifiziert die EN 62304 den Software-Entwicklungs-Prozess. Als zweiter Prozess wird der Software-Wartungsprozess eingeführt. Hier muss der Hersteller einen Plan für die Wartung der Software festlegen in dem Verfahren für den Empfang, die Dokumentation, Evaluation, Lösung und Rückverfolgbarkeit von Rückmeldungen nach Freigabe der Medizinprodukte-Software festgelegt werden. Neben einem Verfahren für die Evaluierung und Implementierung von Drittsoftware (SOUP). (Software of Unknown Provenance) muss die Verwendung des Software-Risikomanagement-Prozesses, des Problemlösungsprozesses für Software und die Verwendung des Software-Konfigurationsmanagement-Prozesses geplant sein. Diese „Hilfsprozesse“ werden im nachfolgenden Abschnitt beschrieben. Jede Rückmeldung muss dahingehend evaluiert werden ob ein Problem mit der Software besteht und mit einem Problembereich dokumentiert werden. Diese Problembereiche müssen wiederum in Hinblick auf die Sicherheit der Software bewertet werden. Gegebenenfalls wird dann der Problemlösungsprozess für Software eingeleitet. Für jede Änderung der

freigegebenen Software muss in weiterer Folge eine Änderungsanforderung verfasst werden. Diese Anforderung muss überprüft, genehmigt, gegebenenfalls mit den Anwendern und Behörden kommuniziert und letztendlich umgesetzt werden. Die Änderungen können als Teil einer neuen System-Freigabe erfolgen oder als Änderungsbausatz, bzw. Software-Patch, welcher nur die geänderten Komponenten enthält und am System separat installiert werden kann.

Der Software-Risikomanagement-Prozess ist für Software der Sicherheitsklasse B und C gefordert und in der Norm ISO 14971 beschrieben. In der EN 62304 wird der Prozess um softwarespezifische Details erweitert. Auf die ISO 14971 wird im Kapitel 3.4 näher eingegangen. In weiterer Folge muss der Hersteller ein Schema für die eindeutige Identifizierung von jenen Konfigurationselementen festlegen, welche für das System überprüft werden sollen. Dieses Schema muss jedes Softwareelement, auch SOUP enthalten. Für jedes verwendete SOUP-Konfigurationselement muss der Titel, der Hersteller und die eindeutige SOUP-Kennzeichnung vom Hersteller dokumentiert werden. SOUP ist bereits fertig entwickelte und verfügbare Software, welche nicht dazu entwickelt wurde um in das Medizinprodukt integriert zu werden. Diese Konfigurationselemente dürfen nur als Reaktion auf genehmigte Änderungs-Anforderungen geändert werden. Werden Probleme mit, oder durch die Software identifiziert so kommt der Problemlösungs-Prozess für Software zur Anwendung. Ein Problembereich muss erstellt werden welcher den Typ, den Umfang und die Kritikalität des Problems beschreibt. Das Problem muss analysiert, alle beteiligten Stellen unterrichtet und das Problem gelöst werden. Abschließend muss die Lösung des Problems verifiziert werden.

3.3.1 Software-Entwicklungslebenszyklus als V-Modell

Zunächst wird der Entwicklungslebenszyklus in der EN 60601-1 [54] in Kapitel 14 und in Anhang H als sogenanntes V-Modell für programmierbare elektrische medizinische Systeme (PEMS) vorgestellt. Die Abbildung 6 zeigt dieses Modell und hebt den in der ISO 62304 abgedeckten Teil hervor. In Kapitel 5 werden die in der Entwicklung des Prototyps durchlaufenen Teilprozesse des Softwareentwicklungslebenszyklus detailliert ausgeführt.

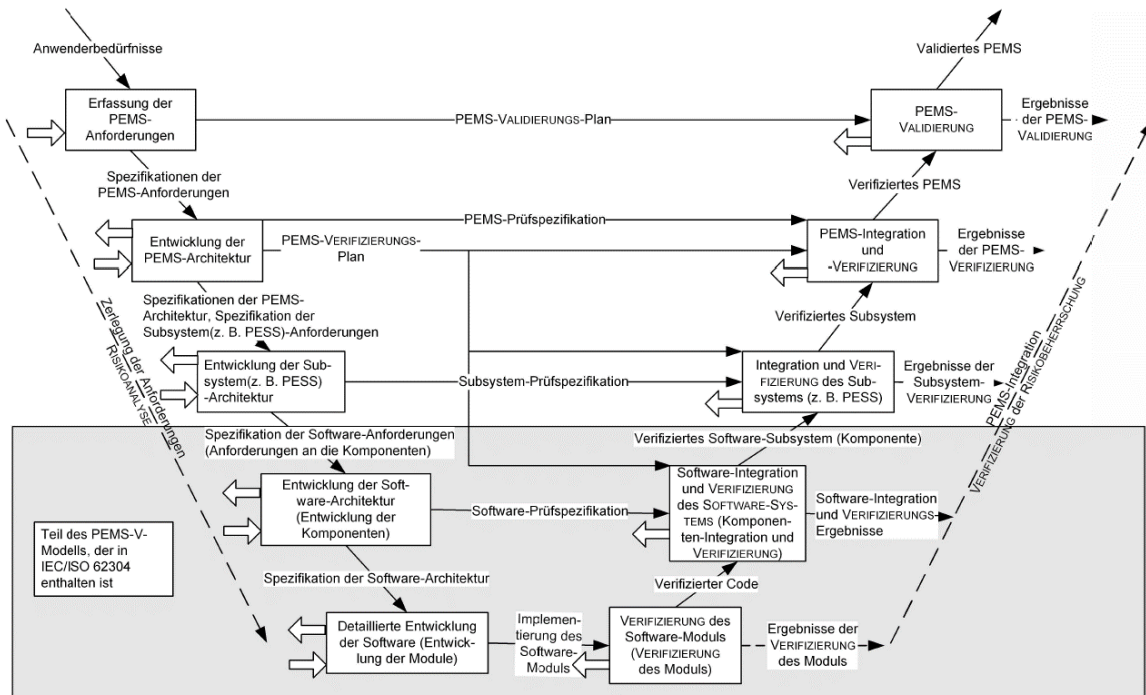


Abbildung 6: Modell des Software-Entwicklungslebenszyklus[49]

3.4 EN ISO 14971 - Anwendung des Risikomanagements auf Medizinprodukte

Da Fehler in medizinischer Software potentiell schwerwiegende Effekte, wie den die Verletzung oder den Tod von Patienten zur Folge haben können, ist bei der Entwicklung der Software die Verwendung eines strukturierten Risikomanagement essentiell. Die Norm EN ISO 14971 stellt Verfahren zum Management von Risiken, für Patienten, Anwender, Dritte und die Umwelt vor. Die Norm teilt Risiken in zwei Bestandteile auf [55]:

- Die Wahrscheinlichkeit des Auftretens des Schadens
- und die Auswirkung, bzw. schwere des Schadens

Um die mit der Software verbundenen Gefährdungen zu identifizieren, die daraus folgenden Risiken und einzuschätzen und zu beherrschen, sowie die Wirksamkeit dieser Beherrschung zu überwachen führt die Norm den Risikomanagementprozess ein.

Zu Beginn des Risikomanagementprozesses muss vom Hersteller eine Risikoanalyse durchgeführt werden. In der **Risikoanalyse** muss zunächst das Medizinprodukt identifiziert und beschrieben werden, die Personen und die Organisation, welche die Risikoanalyse durchführt angeführt werden. Weiters müssen für das Medizinprodukt die

Zweckbestimmung, der vorhersehbare Missbrauch, alle qualitativen und quantitativen Merkmale welche die Sicherheit des Medizinprodukts beeinflussen können dokumentiert werden. Alle potentiellen Gefährdungen müssen sowohl unter Normalbedingungen als auch unter Fehlerbedingungen identifiziert und dokumentiert werden. Abschließend muss das Risiko für jede Gefährdungssituation eingeschätzt werden. Hierbei müssen auch alle vorhersehbaren Kombinationen und Abfolgen der Gefährdungssituationen berücksichtigt werden. Für Situationen in denen die Wahrscheinlichkeit des Auftretens eines Schadens nicht eingeschätzt werden kann, müssen alle möglichen Auswirkungen für die weitere Analyse in der Risikobewertung und Risikobeherrschung dokumentiert werden.

Bei der **Risikobewertung** muss für jede Gefährdungssituation entschieden werden, ob risikomindernde Maßnahmen gesetzt werden müssen. Diese Maßnahmen werden in der **Risikobeherrschung** evaluiert. Die ISO 14971 stellt folgende Risikominderungsmethoden vor:

- integrierte Sicherheit durch Design;
- Schutzmaßnahmen im Medizinprodukt selbst oder im Herstellungsprozess;
- Informationen zur Sicherheit.

Diese Maßnahmen müssen in der angeführten Reihenfolge durchgeführt werden. Wird festgestellt, dass die geforderte Minderung des Risikos nicht realisierbar ist, so muss eine Risiko-Nutzen-Analyse für das Restrisiko durchgeführt werden. Anschließend muss die Umsetzung jeder risikomindernden Maßnahme verifiziert werden und jedes Restrisiko bewertet werden. Bei zu hohem Restrisiko muss wieder eine Risiko-Nutzen-Analyse durchgeführt werden. Weiters müssen durch Risikobeherrschungsmaßnahmen neu eingeführte Risiken überprüft werden. Abschließend muss vom Hersteller sichergestellt werden, dass die Risiken für alle Gefährdungssituationen berücksichtigt wurden und dass das durch das Medizinprodukt verursachte Gesamt-Restrisiko akzeptabel ist. Sollte das Restrisiko nicht akzeptabel sein, so kann der Hersteller durch eine Daten- und Literatur- Analyse bewerten ob der medizinische Nutzen der Zweckbestimmung der Anwendung überwiegt. Ist dem so, so kann das Gesamt-Restrisiko als Akzeptabel beurteilt werden. Alle Aktivitäten müssen in der Risikomanagementakte aufgezeichnet werden, diese Akte wird bei einer Inspektion des Risikomanagements überprüft.

Weiters muss der Hersteller ein System für die Sammlung von Informationen über das Medizinprodukt oder ähnlichen Produkten aus der Herstellung und der Herstellung nachgelagerten Phasen etablieren. Diese Informationen müssen auf mögliche Sicherheitsrelevanz bewertet werden und gegebenenfalls die Risikomanagementakte überprüft werden. Auswirkungen auf die Risikobewertung und auf die Maßnahmen der Risikobeherrschung müssen möglicherweise bewertet werden.

3.5 EN 62366 – Anwendung der Gebrauchstauglichkeit auf Medizinprodukte

Durch die Einhaltung der EN 62366 [56] soll sichergestellt werden, dass die Benutzung der finalen Benutzeroberfläche der Software intuitiv oder einfach zu erlernen ist. Um Benutzungsfehlern durch unzureichende Gebrauchstauglichkeit vorzubeugen, wird mit der EN 62366 der gebrauchstauglichkeitsorientierte Entwicklungsprozess eingeführt. Die Einhaltung dieses Prozesses und die Führung der Gebrauchstauglichkeitsakte sollen zu einer angemessenen Gebrauchstauglichkeit führen. Im Rahmen des gebrauchstauglichkeitsorientierten Entwicklungsprozesses muss die Spezifikation des Benutzungskontexts des Medizinprodukts dokumentiert werden. In weiterer Folge werden die häufig benützten Funktionen definiert, dies sind jene Funktionen die zur Ausführung einer Kernaufgabe benötigt werden. Anschließend muss die Ermittlung von sicherheitsbezogenen Merkmalen mit Schwerpunkt auf die Gebrauchstauglichkeit gemäß der EN ISO 14971 [55] durchgeführt werden. Anhand der EN ISO 14971 werden Gefährdungssituationen, welche von der Benutzerschnittstelle ausgehen bewertet. Diese Funktionen werden als Hauptbedienfunktionen bezeichnet und müssen vom Hersteller festgelegt und dokumentiert werden. Vom Hersteller wird in weiterer Folge die Gebrauchstauglichkeit spezifiziert. Diese Spezifikation soll überprüfbare Forderungen an die Gebrauchstauglichkeit der Hauptbedienfunktionen und zur Verifizierung der Gebrauchstauglichkeit enthalten. Anschließend ist ein Plan für die Validierung der Gebrauchstauglichkeit zu erstellen. Nachdem die Benutzerschnittstelle anhand der definierten Spezifikationen implementiert wurde, wird diese nach dem Plan zur Validierung der Gebrauchstauglichkeit validiert. Alle diese Aktivitäten müssen in der Gebrauchstauglichkeitsakte dokumentiert werden. Eine ausführlichere Beschreibung der Umsetzung der Validierung der Gebrauchstauglichkeit ist im Kapitel 7 ausgeführt.

Da der Fokus der EN 62366 auf der Vermeidung der Gefährdung von Patienten und Anwendern liegt, verweist sie normativ auf die Verwendung eines Risikomanagements nach ISO 14971. In Abbildung 7 ist die enge Verzahnung des gebrauchstauglichkeitsorientierten Entwicklungsprozesses mit dem Risikomanagementprozess illustriert.

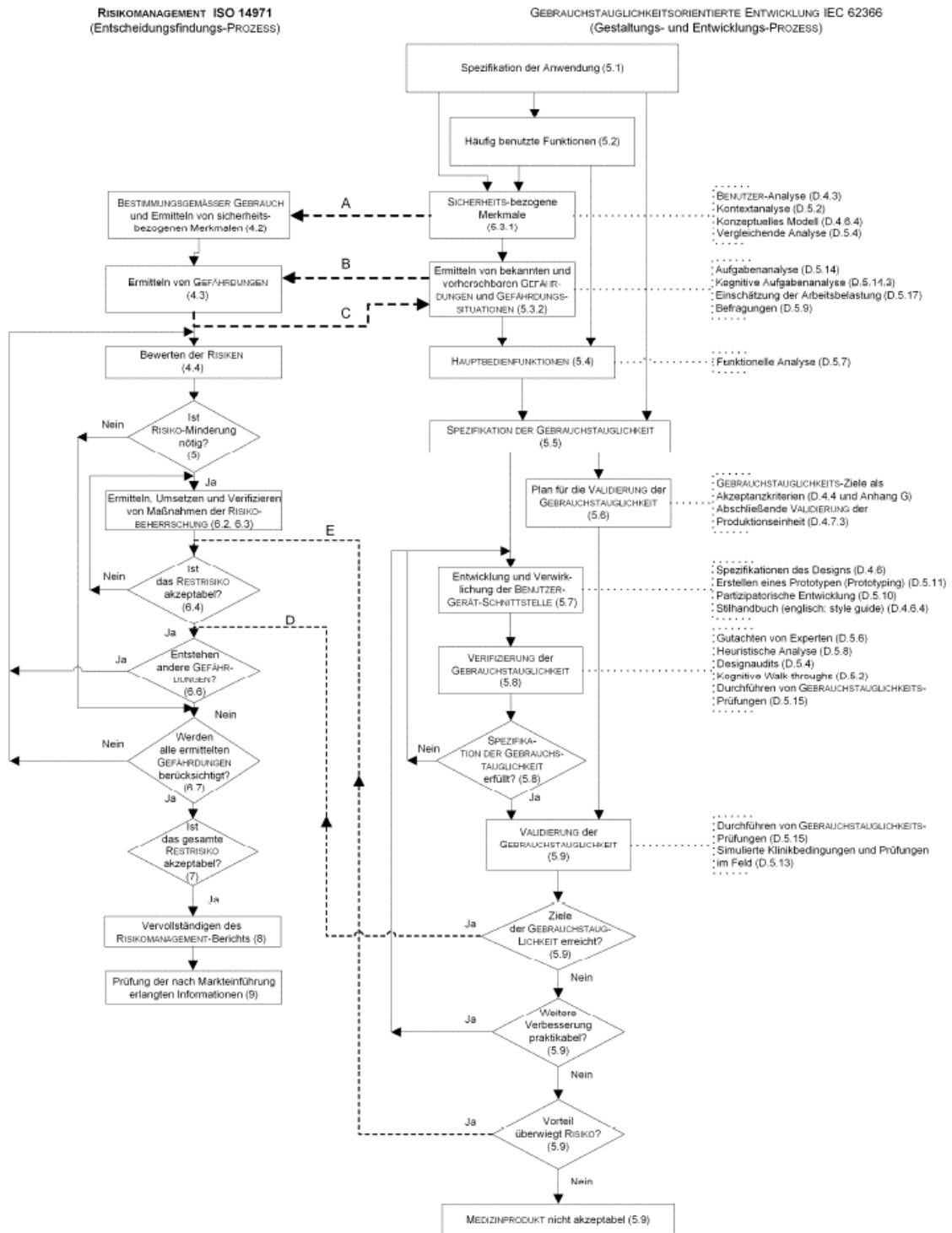


Abbildung 7: Zusammenhang des Gebrauchstauglichkeitsorientierten Entwicklungsprozesses (EN62366) und des Risikomanagementprozesses (EN 14971) [56]

Methoden und Ergebnisse:

4 Technischer Hintergrund

Im folgenden Kapitel wird auf den technischen Hintergrund des im Rahmen dieser Masterarbeit entwickelten Prototyps eingegangen. Zunächst wird der theoretische Hintergrund, wie die Architektur der Anwendung und die bei der Entwicklung verwendete Projektmanagementmethode erläutert. Im letzten Teil dieses Kapitels werden die eingesetzten Technologien beschreiben.

4.1 Architektur von Webapplikationen

Webapplikationen sind Programme, welche auf einem Webserver laufen und einen Internetbrowser als Benutzeroberfläche verwenden [57]. Eine Webapplikation ist also eine Anwendung basierend auf dem Client-Server Modell. Grundsätzlich kann man die Entwicklungsansätze von Webapplikationen in serverseitige Ansätze (Thin-Client), bei denen die Verarbeitung der Programmlogik am Server stattfindet (z.B. durch PHP), und clientseitige Ansätze (Fat-Client), bei denen zumindest ein Teil der Logik am Client ausgeführt wird (z.B. bei der Verwendung von AJAX) einteilen. In der Realität kann jedoch häufig keine strikte Trennung zwischen den Ansätzen vorgenommen werden, daher wird meist eine Kombination eingesetzt.

Typischerweise wird eine Webapplikation in unterschiedliche voneinander getrennte Schichten geteilt. Die serverseitige Einteilung erfolgt zumeist in einem Drei-Schichten-Modell bestehend aus Präsentation-, Business- und Data-Schicht wie in Abbildung 8 dargestellt. Gängige Design-Pattern für Webapplikationen sind in [58] gelistet. Nachfolgend wird auf die Model-View-Controller-Architektur (MVC) im Speziellen eingegangen, da dieses Design-Pattern auch technologiebedingt bei der Entwicklung des Web-Frontend-Prototyps verwendet wurde.

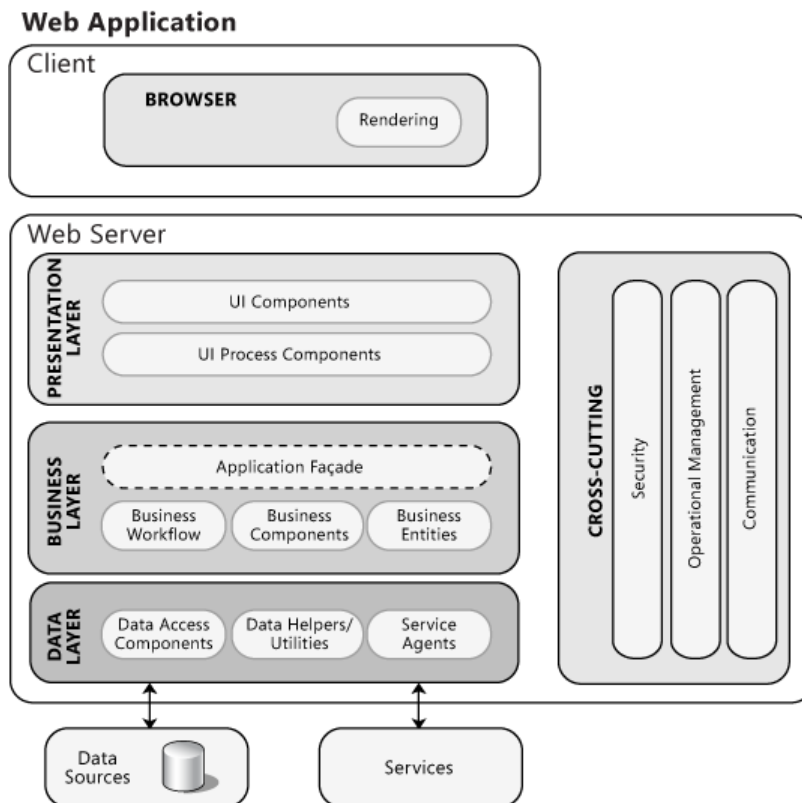


Abbildung 8 Typische Darstellung der Elemente einer Webanwendung [58]

4.1.1 Model-View-Controller-Architektur

Die MVC-Architektur ist ein Muster (Pattern) um die strukturierte Entwicklung von Anwendungen zu ermöglichen. Der Ursprung der MVC – Architektur liegt in der Programmiersprache *Smalltalk*, bei der sie zur Erstellung von grafischen Benutzeroberflächen verwendet wurde [59]. Mittlerweile zählt diese Architektur zu den Standards bei der Entwicklung von grafischen Benutzeroberflächen und Web-Anwendungen.

Die dabei verwendeten Softwarekomponenten werden in die drei Kategorien Model, View und Controller eingeteilt (siehe Abbildung 9).

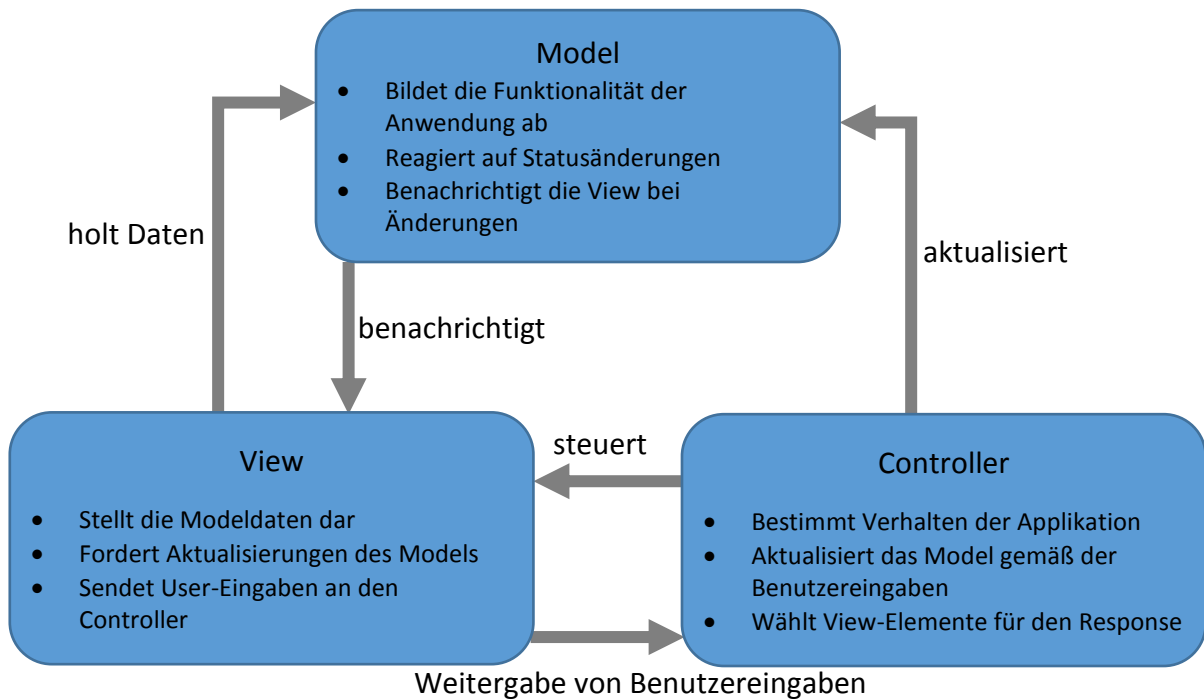


Abbildung 9: Die Model-View-Controller-Architektur

- Die Model-Komponenten dienen zum einen zur Repräsentation und Haltung von Anwendungsdaten, zum anderen auch zur Verarbeitung dieser Daten. Aus diesem Grund stellen sie die eigentliche Geschäftslogik dar.
- Als View-Komponenten bezeichnet man alle Software-Teile, welche zur Darstellung der Benutzeroberfläche verwendet werden. Die darzustellenden Daten werden dazu von den Model-Komponenten bereitgestellt. Alle vom Benutzer ausgelösten Aktionen werden von den View-Komponenten zur Verarbeitung an die Controller-Komponenten weitergeleitet.
- Die Controller-Komponenten koordinieren die Anwendungsabläufe und Navigation innerhalb der Anwendung. Sie stellen die Verbindung zwischen Model und View dar. Zum einen rufen sie den Benutzereingaben entsprechende Funktionen der Model-Komponenten auf, und zum anderen entscheiden sie darüber, welche Teile der View-Komponenten dem Benutzer dargestellt werden.

Durch diese strikte Trennung der Komponenten wird die Wartbarkeit der Software erhöht. Dadurch wird der Austausch, die Bearbeitung sowie die Erweiterung einzelner Teile und Komponenten erleichtert, ohne die restliche Anwendung komplett überarbeiten zu müssen.

Die Verwendung der Model-View-Controller-Architektur bietet im Hinblick auf Webanwendungen jedoch auch Nachteile. Webanwendungen sind wie in Kapitel 4.1 erwähnt

in Client- und Server-Seite geteilt. Die View-Komponenten werden am Client angezeigt, die Model- und Controller-Komponenten können jedoch theoretisch beliebig zwischen Client und Server verteilt sein. Umso dynamischer eine Webanwendung implementiert wurde, desto eher werden Model- oder Controller-Komponenten am Client ausgeführt. Ein Entwickler muss dadurch schon im Vorhinein die Webanwendung aufteilen um eine Trennung zwischen den Komponenten zu ermöglichen. Diese Entscheidungen werden zudem auch aufgrund der verwendeten Technologien beeinflusst. Es ist sozusagen die Entscheidung zwischen einer *Thin-* oder *Fat-Client* Architektur zu treffen. Bei der Verwendung der MVC-Architektur fällt somit zunächst durch die strikte Trennung der Komponenten ein Mehraufwand bei der Implementierung der Anwendung an [60].

4.2 Entwicklung von Software mit agilen Entwicklungsmethoden

In der Praxis bietet es sich gerade bei Softwareprojekten mit hoher Kunden- bzw. Endbenutzereinbeziehung an die Software mit agilen Entwicklungsmethoden zu entwickeln. Diese Entwicklungsmethoden beruhen auf dem agilen Manifest, und sind im Wesentlichen nicht nur auf die Softwareentwicklung begrenzt sondern sind als allgemeine Projektmanagementmethoden anzusehen.

„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.“ [61]

Als agile Entwicklungsmethoden wären Pairprogramming, test-driven Development, Adaptive Software Development, Scrum und andere zu nennen. Im nachfolgenden Absatz wird Scrum als spezielle Methode vorgestellt, da diese bei der Entwicklung des Prototyps angewandt wurde. Als Quelle für den folgenden Absatz wurde Scrum Guides [62] verwendet.

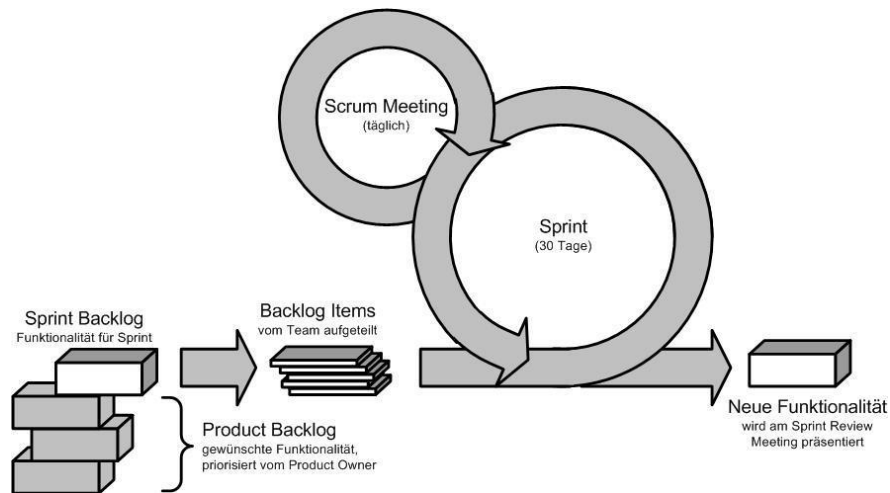


Abbildung 10: Der Scrum Prozess [63]

Scrum folgt einem empirischen, iterativen und inkrementellen Ansatz (siehe Abbildung 10) zur Risikominimierung und Steigerung der Vorhersehbarkeit des Projektausganges. Der *Sprint* ist das Herzstück von Scrum. Er ist ein beschränkter Zeitabschnitt von maximal einem Monat in dem ein Produkt-Inkrement hergestellt wird. Beim Sprintstart wird das *Sprint Planning* durchgeführt, in dem bestimmt wird, welche Inhalte im nächsten Produkt-Inkrement enthalten sein sollen und wie diese zu erstellen sind. Diese Inhalte werden vom *Product-Backlog* in das *Sprint-Backlog* übergeführt. Während des Sprints gibt es ein tägliches *Scrum-Meeting* von 15 Minuten innerhalb dessen sich das Team synchronisiert, den Fortschritt in Richtung des Sprint Zieles überprüft und die Arbeit des nächsten Tages plant. Die Projektbeteiligten sind als *Scrum-Team* und *Stakeholder* anzusehen. Am Ende des *Sprints* findet das *Sprint-Review* statt indem das Produkt-Inkrement überprüft wird und das *Product-Backlog* gegebenenfalls angepasst wird. Das *Scrum-Team* besteht aus dem *Product-Owner*, dem *Entwicklerteam* und dem *Scrum-Master*. Der *Product-Owner* ist für die Wertmaximierung des Produkts und die Arbeit des *Entwicklerteams* verantwortlich. Er ist der Einzige der für die Verwaltung des *Product-Backlogs* verantwortlich ist. Das *Entwicklungsteam* erstellt innerhalb jedes *Sprints* ein fertiges Produkt-Inkrement. Das Team organisiert seine Arbeit während des *Sprints* selbstständig. Der *Scrum-Master* ist für die Durchführung von Scrum zuständig. Er sorgt dafür, dass sich das *Scrum-Team* an die Theorien und Regeln von Scrum hält. Die Kunden bzw. der Endbenutzer stellen normalerweise die *Stakeholder* dar.

Die Einhaltung des, in der EN 62304 benannten, Softwareentwicklungslebenszykluses ist auch mit agilen Entwicklungsmethoden möglich. So müssen die einzelnen Phasen des Lebens-

zyklus nicht vollständig und strikt aufeinanderfolgend durchlaufen werden, es genügt wenn die Software der Norm entsprechend dokumentiert wird und die Nachvollziehbarkeit der Entwicklungsschritte gewährleistet werden kann. Wie die eigentliche Entwicklung durchgeführt wird ist für die Norm nicht relevant. So kann während der Entwicklung leichter auf Veränderungen an den Softwarespezifikationen reagiert werden, als bei einer Entwicklung nach dem V-Modell. Es kann jedoch nicht auf eine umfangreiche Dokumentation verzichtet werden. In diesem Punkt weicht die Entwicklung von Software als Medizinprodukt zwangsweise vom agilen Manifest ab.

4.3 Verwendete Technologien

Da sowohl die Backend- als auch die Android-Frontend- Komponente auf der Java Technologie basiert war es naheliegend auch für den Prototyp des Web-Frontends auf diese Technologie zurückzugreifen. Dies ermöglichte die Wiederverwendung von gemeinsamen Software-Bibliotheken und erleichterte die Zusammenarbeit mit anderen Entwicklern im Softwareteam. Da das System plattformunabhängig und ohne zusätzliche Installation auf den Client-PC-Systemen im Krankenhaus funktionieren muss, fiel die Wahl auf eine browserbasierte Lösung. Dadurch wurde als grundlegendes Web-Framework JavaServer Faces 2.2. gewählt welches in der Java Platform, Enterprise Edition 7 enthalten ist. Die Funktionalität des Web-Frameworks wird durch die Verwendung der Komponentenbibliothek Primefaces erweitert. Diese Technologien stellen die Basis für den Prototyp des Web-Frontends dar. In den folgenden Unterkapiteln werden sie und andere verwendete Software von Drittherstellern näher vorgestellt.

4.3.1 JavaServer Faces (JSF)

Wie in Kapitel 5.1 erwähnt gab es vor der Einführung von JSF keinen Standard für die Anwendung der MVC-Architektur für Java-Webanwendungen. Durch den Java Specification Request 127 wurde die Spezifikation eines komponentenbasierten Frameworks zur Vereinfachung der Entwicklung von Java-Webanwendungen mit graphischen Benutzeroberflächen entwickelt. Das Ergebnis dieser Spezifikation ist das JSF - Framework:

JavaServer Faces (JSF) is a user interface (UI) framework for Java web applications. It is designed to significantly ease the burden of writing and maintaining applications that run on a Java application server and render their UIs back to a target client. [64]

Das JSF – Framework stellt eine Grundlage zur Entwicklung von Java-Webanwendungen dar und führt zur Verwendung der Model-View-Container Architektur.

Die Geschäftslogik (Model-Komponente) wird mittels Java-Beans umgesetzt. Diese Beans werden nicht direkt in die View-Komponenten eingebunden sondern durch das Controller-Servlet automatisch verwaltet. Die automatisch verwalteten Beans werden als „managed Bean“ bezeichnet und annotiert. In ihnen werden die Anwendungsdaten verwaltet und in ihnen wurde im GlucoTab-Web-Frontend-Prototyp die Kommunikation mit dem Backend umgesetzt.

Die eigentliche Benutzeroberfläche (View) wird mit ihren UI-Elementen in JSP-XML Dateien organisiert. Die JSF-Standard Komponenten können in Renderer-, Converter- und Validator-Komponenten eingeteilt werden. Die Renderer-Komponenten dienen zur Darstellungsgenerierung im Browser. Aus ihnen wird vom Renderer des Webservers HTML-Code erzeugt welcher vom Browser interpretiert werden kann. Die Validator-Komponenten dienen zur Überprüfung von Benutzereingaben im Hinblick auf vordefinierte Parameter und durch Converter-Komponenten kann die Benutzereingabe nach definierten Regeln verändert werden, bevor sie an die Model-Komponenten weitergegeben wird. Dies kann z.B. bei Internationalisierung von Webanwendungen oder bei der Umsetzung von Benutzereingaben in Java-Objekte genutzt werden.

Als Controller dient das *FacesServlet* welches vom JSF Framework zur Verfügung gestellt wird. Das *FacesServlet* stellt die zentrale Steuereinheit jeder JSF-Anwendung dar und verarbeitet alle *Requests* welche vom Client an die Webanwendung gestellt werden. Das *FacesServlet* wird zum Einen über die XML-Konfigurationsdatei `web.xml` und zum Anderen über `faces-config.xml` konfiguriert. Letztere dient zur Konfiguration von Navigationsregeln, Registrierung von Komponenten wie Validatoren, Convertern und Renderern [65].

4.3.2 Primefaces

Um die Funktionalität der JSF-Komponenten zu erweitern werden sogenannte Komponentenbibliotheken eingesetzt. Primefaces ist eine der populärsten Komponentenbibliotheken für JSF. Seit Primefaces im Jahr 2010 von PrimeTek (Ankara, Türkei) veröffentlicht wurde, ist die Komponentenbibliothek auf über 100 individuelle Komponenten angewachsen. Sie erweitern die JSF-Standardkomponenten um zusätzliche AJAX

Funktionalitäten und Funktionen im Bereich der Tabellen, Bäumen, Listen Diagrammen und Eingabekomponenten. Die Einbindung in ein bestehendes JSF-Projekt erfolgt sehr einfach über das Hinzufügen einer Maven-Abhängigkeit in der Datei `pom.xml`. Primefaces wurde aufgrund der erweiterten Funktionalität der Tabellenkomponenten für die Darstellung des Medikationsplanes, der Baumfunktionen bei der Medikamentensuche, sowie zusätzlicher AJAX-Funktionalitäten für den Programmablauf der einzelnen modalen Dialoge im Prototyp verwendet.

4.3.3 Diagram-Bibliothek Dygraphs

Für die Darstellung des BZ-Verlaufsdigrams und der Historie der Medikamentengaben wurden wie in 5.3.1 beschrieben mehrere Prototypen entwickelt. Im Verlauf der Entwicklung dieser Prototypen wurden die Spezifikationen für die Darstellung der Diagramme konkretisiert.

Die Darstellung des BZ-Verlaufsdigramms und der Medikamentengaben wird in mehreren Diagrammen umgesetzt. Essentiell bei der Darstellung dieser Informationen ist die Synchronisation der Zeitachsen aller Diagramme. Deshalb wurde die Synchronisation der Zoom und Scroll-Funktionen sowohl für das BZ-Verlaufsdigramm als auch die Darstellung der Medikamentengabehistorie angestrebt. Zur Darstellung der Informationen, wie Dosierung und Uhrzeit des Medikamentes muss die Softwarelösung auch über eine einfach zu automatisierende Möglichkeit der Darstellung von Texten auf der Diagrammfläche verfügen. Weiters sollte es sich bei der Software um eine Open-Source-Lösung handeln, welche in den aktuell gängigen Internetbrowsern lauffähig ist.

Während einer Internetrecherche wurde neben kommerziellen Lösungen von Highcharts (Vik i Sogn, Norwegen) und plotly (Montreal, Kanada) auch die Opensource-Lösung Dygraphs gefunden, welche die oben genannten Anforderungen erfüllt.

Die Anfänge von Dygraphs liegen im Jahr 2006 als von Dan Vanderkam entwickeltes internes Tool bei Google. Im Jahr 2009 wurde die JavaScript-Bibliothek als Open-Source-Software auf github veröffentlicht. In der darauffolgenden Zeit wurde Dygraphs von unterschiedlichen Personen weiterentwickelt, bis die Version 1.0 im August 2012 veröffentlicht wurde. Im Prototyp des Web-Frontends wird die Bibliothek in der Version 1.1.1 verwendet. Auf github [66] gibt es Meldungen zu 175 offenen Problemen, welche jedoch entweder Funktionen des

Paketes betreffen, welche im Prototyp keine Verwendung finden, oder kein Risiko für den Patienten darstellen.

4.3.4 Apache Maven

Als Building-Tool wird sowohl im Frontend als auch im Backend *Apache Maven* (Apache Software Foundation, Los Angeles, USA) verwendet. So konnten während der Entwicklung automatisiert SOUPs wie zum Beispiel *JodaTime* (joda.org), als Ersatz für die Java Datums- und Zeit-Bibliothek, oder Primefaces eingebunden werden. Während der Entwicklung des Prototyps wurde die Funktionalität des Backends stetig erweitert. Die neuen Subversionen wurden getrennt vom Frontend gebaut und ebenfalls über Maven in das Web-Frontend-Projekt integriert. In diesem Kapitel wird basierend auf die Apache-Maven Website [67] grob auf die Grundfunktionalität von Apache Maven eingegangen.

Apache Maven ist ein Open-Source Werkzeug zum Verwalten und Bauen von Java-Anwendungen. *Apache Maven* stellt ein einheitliches Build-System zu Verfügung. Das bedeutet, dass der Build-Prozess in allen Maven Projekten derselbe ist. Alle Maven-Projekte haben eine ähnliche Datenstruktur. Das Herzstück jedes Maven-Projektes stellt die POM.xml (Project Object Model) Datei dar. Die `POM.xml`-Datei beinhaltet Informationen über die Struktur, Versionierung, Ressourcen, Abhängigkeiten und das Konfigurations-Management des Projekts. Zusätzlich zur POM.xml-Datei verfügt jedes Maven-Projekt über ein *src*-Verzeichnis, in dem das Quellmaterial (wie z.B. der Quellcode) liegt, und ein *target*-Verzeichnis, in welchem sich das Ergebnis des *Builds* befindet. Apache Maven baut Projekte anhand des sogenannten *Build-Lifecycle*. Der *Build-Lifecycle* besteht aus mehreren Phasen, welche nacheinander ausgeführt werden. Die wichtigsten Phasen sind:

- **Validate:** überprüft ob alle benötigten Informationen zum Bauen des Projekts vorhanden sind
- **Compile:** kompiliert den Quellcode des Projekts
- **Test:** verwendet ein Unit-Test Framework zur Überprüfung des kompilierten Codes
- **Package:** packt den kompilierten Quellcode entsprechend des im `POM.xml` definierten Paket-Typs
- **Integration-test:** führt optionale Integration-Tests durch
- **Verify:** verifiziert die Validität des gepackten Pakets

- **Install:** installiert das generierte und verifizierte Paket in das lokale Maven-Repository
- **Deploy:** kopiert das generierte Paket an eine fremde Adresse

Jede dieser Phasen kann durch den Befehl `mvn <Phase>` ausgeführt werden, hierbei werden zuerst alle vorherigen Phasen ausgeführt.

5 Entwicklung des Prototyps für das Web-Frontend

5.1 Anforderungsanalyse

Bei Beginn der Masterarbeit sollte ein Web-Frontend mit den Funktionen entwickelt werden welche die Release 3.0 der Android-Anwendung den Ärzten zur Verfügung stellte. Im Wesentlichen war dies die Patientenaufnahme in das System sowie anschließend die Tagesdosisanpassung innerhalb der Basis-Bolus Therapie. Für diese Anforderungen wurden basierend auf dem Design der Android Analyse Mock-up Screenshots erstellt (siehe Abbildung 11).

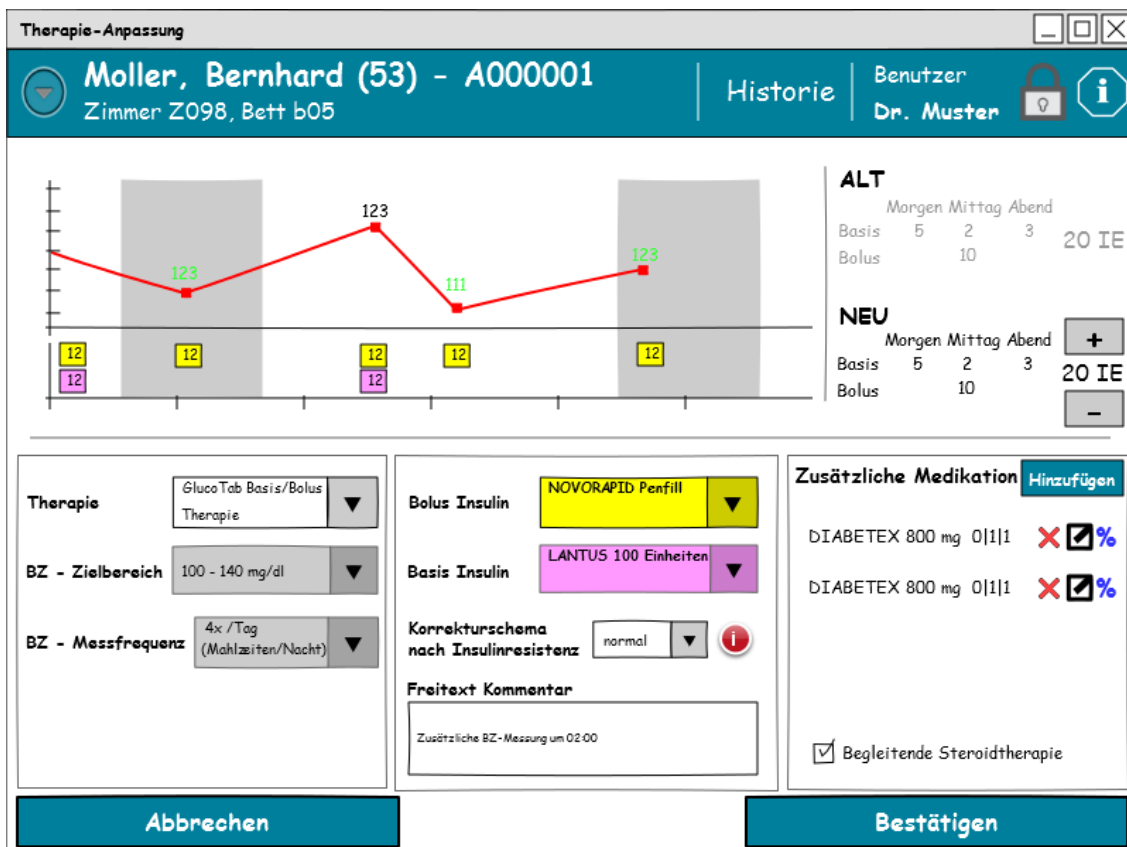


Abbildung 11: Früher Mock-Up Prototyp des Web-Frontends

Diese Mock-Up Screenshots wurden in einem iterativen Prozess in Rücksprache mit dem Entwicklerteam, sowie dem klinischen Personal der medizinischen Universität Graz diskutiert und weiterentwickelt. Durch diesen Prozess näherten sich die Anforderungen an das System näher an jene der Android-Anwendung für Version 4.0 an. In dieser Softwareversion soll die Dokumentation einer eigenen Therapie ohne Entscheidungsunterstützung sowie die Dokumentation und Übernahme der Vortherapie des Patienten ermöglicht werden. Ein späterer Entwurf dieser Benutzerschnittstelle befindet sich in Abbildung 12. Weiters wurde die Darstellung der Medikamentenverordnung an jene einer virtuellen Fieberkurve angepasst. Diese ist eine gängige Darstellung in elektronischen Medikamentenverordnungssystemen.

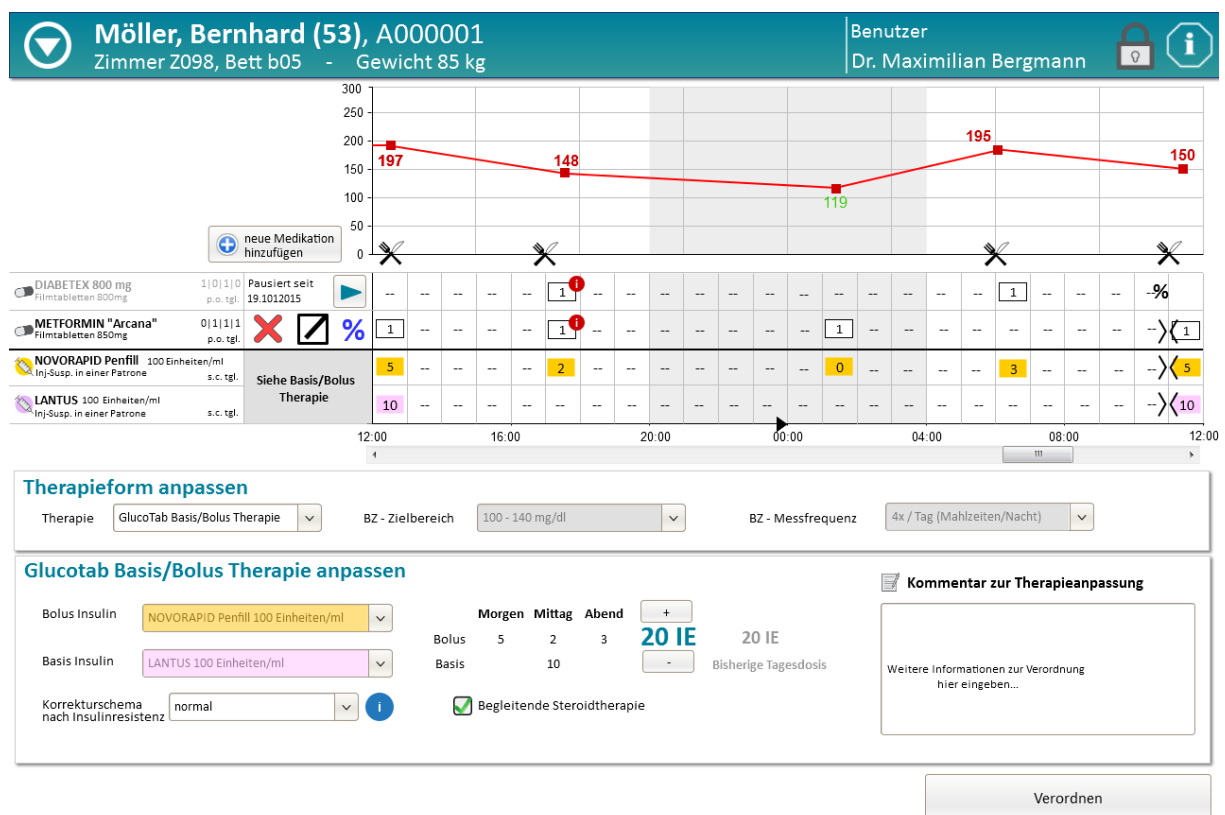


Abbildung 12: Weiter fortgeschrittener Entwurf der Benutzeroberfläche

Der Fokus auf die Anforderungen der Ärzte an das System wurde beibehalten, da von Ärzteseite verstärkt die Bedienbarkeit des Systems über eine Schnittstelle am PC gewünscht wurde. Dies ist darin begründet, dass einerseits ein großer Teil der Arbeitszeit der Ärzte am PC verbracht wird und häufig während der Visite ein Notebook mitgeführt wird. Das Pflegepersonal schätzt hingegen die Mobilität, welche durch die Androidanwendung ermöglicht wird.

5.1.1 Spezifikation der Anwendung

Die Spezifikation der Anwendung des Medizinprodukts muss in der Gebrauchstauglichkeitsakte zu finden sein. Für das Web-Frontend ist sie in Tabelle 2 spezifiziert, die Spezifikation unterscheidet sich von der des Tablet-Systems bei der Benutzergruppe und den Gebrauchsbedingungen. Für den Prototyp des Web-Frontends sind nur Ärzte als Benutzer vorgesehen, da nur jene Funktionen im Prototyp umgesetzt wurden welche zur Therapieplanung und Verwaltung benötigt werden.

Tabelle 2: Spezifikation des Web-Frontend-Prototyps

Medizinische Indikation	Therapieunterstützung bei der Behandlung von DM Typ 2 Patienten welche stationär in einer Krankenanstalt behandelt werden
Vorgesehene Patientengruppe	
Alter	Älter als 18 Jahre
Gewicht	Keine Einschränkung
Gesundheit	Erkrankt an DM Typ 2
Vorgesehenes Benutzerprofil	Ärzte in Krankenanstalten welchen die Behandlung von DM Typ 2 gesetzlich gestattet ist
Vorgesehene Gebrauchsbedingung	Verwendet von professionellen Anwendern innerhalb von Krankenanstalten am Computerarbeitsplatz mit Netzwerkverbindung

Die folgenden Anforderungen an das System wurden festgelegt:

- Einschreiben von Patienten in das System vom Krankenausinformationssystem aus und Initialisierung der Diabetestherapie:
 - Eingabe der Vortherapie
 - Auswahl der Therapieform aus Basis/Bolus-Therapie (mit Entscheidungsunterstützung) oder der eigenen Therapie (mit optionaler Übernahme der Vortherapie, ohne Entscheidungsunterstützung)
 - Initialisierung der gewählten Therapieform

- Mit Eingabemöglichkeiten für das Gewicht und den Kreatininwert des Patienten
- Anpassung der Therapie und Therapieplanung
 - Darstellung des BZ-Verlaufs des Patienten
 - Darstellung der bislang verabreichten Mahlzeiten und Medikamentengaben
 - Anpassung der Tagesdosis, welche durch die Entscheidungsunterstützung vorgeschlagen wird
 - Medikamentensuche mit Filteroptionen
 - Dokumentation der periodischen Verordnung von Medikamenten (Intervallverordnung)
 - Dokumentation von Einmalgaben von Medikamenten
 - Dokumentation der abgesetzten Medikamente
 - Nachträgliche Dokumentation der Vortherapie
 - Änderung der gewählten Therapieform

5.1.2 Definition der Hauptbedienfunktionen

Die Hauptbedienfunktionen setzen sich aus allen häufig benützten Funktionen und den sicherheitsbezogenen Funktionen zusammen [56]. Da im Webfrontend nur Funktionen implementiert werden welche auch im Android-Frontend umgesetzt sind wurde bei der Definition der Hauptbedienfunktionen des Prototyps auf die Hauptbedienfunktionen des GlucoTab-Systems mit Android-Frontend zurückgegriffen. Auf die nur das Webfrontend-betreffenden technischen Sicherheitsrisiken wird kurz im nächsten Kapitel eingegangen. Alle 37 Hauptbedienfunktionen werden gemäß der Spezifikation der Gebrauchstauglichkeit des Prototyps beim Usability-Test überprüft. Die folgenden Hauptbedienfunktionen sind für den Prototyp definiert:

- Öffnen der Patientenansicht
- Ändern der Therapieform, ausgehend von der Basis/Basis/Bolus-Therapie
- Auswahl des Basal und Bolus Insulins in der Basis/Basis/Bolus-Therapie
- Start der Berechnung der initialen Tagesdosis
- Eingabe von Kreatininwert und Gewicht
- Berechnung der Initialen Tagesdosis in der Basis/Bolus – Therapie
- Overrulen der Berechnung der initialen Tagesdosis

- Ändern der Therapieform, ausgehend von der eigenen Therapie
- Auswahl eines Korrekturschemas für die Insulinresistenz
- Bestätigen der initialen / angepassten Tagesdosis
- Start der Tagesdosisanpassung in der Basis/Bolus-Therapie
- Start des Einschreibprozesses für einen neuen Patienten
- Hinzufügen der Vortherapie
- Bestätigen des Einschreibprozesses eines neuen Patienten
- Start der Therapieanpassung in der eigenen Therapie
- Anpassen des BZ-Zielbereichs in der eigenen Therapie
- Anpassen der BZ-Messfrequenz in der eigenen Therapie
- Aktivieren der Option der begleitenden Steroidtherapie
- Bestätigen der Therapieinitialisierung/anpassung
- Auswahl einer aktiven Intervallverordnung in der virtuellen Fieberkurve
- Pausierung einer Intervallverordnung
- Absetzen einer Intervallverordnung
- Auswahl einer pausierten Intervallverordnung in der virtuellen Fieberkurve
- Fortsetzen einer pausierten Intervallverordnung
- Auswahl einer abgesetzten Verordnung in der Tabelle für abgesetzte Verordnungen
- Wiederaufnahme einer abgesetzten Verordnung
- Dosisangabe für jede Tageszeit einer Intervallverordnung
- Auswahl eines Korrekturschemas in der eigenen Therapie
- Ändern der Verabreichungsfrequenz einer Intervallverordnung
- Bestätigen oder Ändern des Startdatums einer Intervallverordnung
- Einstellen der Dauer einer Intervallverordnung
- Bestätigen einer Intervallverordnung
- Bestätigen eines geänderten Korrekturschemas
- Filtern der Medikamentenliste durch Texteingabe oder Auswahl einer ATV-Gruppe
- Auswahl eines Medikaments aus der Medikamentenliste
- Auswahl der Verordnung einer Einmalgabe
- Dosisangabe für die Verordnung einer Einmalgabe

5.1.3 Definition technischer Rahmenbedingungen auf der Clientseite

Viele KIS verwenden zur Darstellung von Drittherstellere Software den mit dem Betriebssystem ausgerollten Internetbrowser. Da in Krankenhäusern häufig Betriebssysteme von Microsoft verwendet werden [68], wird als Standardbrowser der Microsoft Internet Explorer angenommen. Da aktuell ca. 89% der benutzten Internet Explorer der Version 10 oder neuer angehören [69], und es gerade bei älteren Versionen des Internet Explorers zu Kompatibilitätsproblemen bei der Verwendung von aktuellen Technologien kommt, wurde beschlossen Internet Explorer Version 10 oder neuer als Standardbrowser für die Entwicklung des Prototyps heranzuziehen. Grundsätzlich wurde jedoch auf die Komptabilität des Systems mit den aktuellen Versionen von gängigen Internetbrowsern wie Mozilla Firefox und Google Chrome geachtet.

Durch die Verwendung eines responsiven Webdesigns könnte es leichter zu Anzeige-problemen und zu einer wesentlich aufwändigeren Validierung der Benutzeroberfläche kommen. Dennoch sollte der Prototyp so entworfen werden dass die Darstellung in den gängigsten Bildschirmauflösungen uneingeschränkt möglich ist. Daher wurde der Prototyp mit einem klassischen, sich nicht automatisch anpassenden Design entworfen. Gemäß [70], einer Statistik unter Berücksichtigung von mehr als 2,5 Millionen Webseiten, verwenden rund 85% der Benutzer eine Bildschirmauflösung höher als 1280x768 px. Dies gibt zwar keinen spezifischen Einblick in die in Krankenhäusern verwendeten Bildschirmauflösungen, es ist jedoch davon auszugehen das von Ärzten im Krankenhaus verwendete PCs dem aktuellen Stand der Technik entsprechen. Basierend auf dieser Annahme wurde als niedrigste vom Prototyp unterstützte Bildschirmauflösung eine Auflösung von 1280x768 px festgelegt.

Der Prototyp wird als Webapplikation umgesetzt. Diese Applikation ist per Spezifikation nur innerhalb des IT-Netzwerks des Krankenhauses verwendbar. Dies verringert die Sicherheitsrisiken durch externe Angriffe. Dennoch wird hier kurz auf die Sicherheitsaspekte des Prototyps als Webapplikation eingegangen. Das Open Web Application Security Project definiert die 10 häufigsten und risikobewerteten Sicherheitsschwächen wie in der folgenden Auflistung [71], gleichzeitig wird die Situation des Prototyps in Bezug auf diese Risiken erläutert:

Injection: Wird auf dem Persistenz-Layer im Backend gehandelt. Des Weiteren findet im Frontend standardmäßiges „escaping“ der Benutzereingaben statt (siehe Cross-Site Scripting). Im Frontend werden jedoch grundsätzlich nur in Kommentarfeldern andere Zeichenfolgen als Zahlenwerte zugelassen. Alle anderen Eingaben werden validiert.

Fehler in Authentifizierung und Session-Management: Im produktiven Webfrontend geschieht die Authentifizierung über das KIS, es findet kein gesonderter Login in das GlucoTab-System statt. Innerhalb des Prototyps werden alle veränderbaren Parameter im kürzest-möglichen Scope gehalten. Session-IDs sind nicht von außen erkennbar. Sonst findet jedoch kein Session-Handling statt, Ereignisse wie Session-Timeouts werden nicht verarbeitet.

Cross-Site Scripting: JSF verhindert Cross-Site Scripting durch das Standardmäßige „escaping“ der Benutzereingabe.

Unsichere Direkte Objektreferenzen: Für Zugriff auf die Daten des Backends muss bei jedem Webservice-Aufruf ein Identifier mitgesendet werden. Alle übermittelten, für die Therapie verwendeten Daten werden im Backend nochmals auf Plausibilität geprüft. Dies sollte die Risiken der Übermittlung von gefälschten Daten vermeiden.

Sicherheitsrelevante Fehlkonfiguration: Grundsätzlich werden bei der Implementierung die aktuellsten Versionen der Tools und SOUPs verwendet. Eine sicherheitsbezogene Konfiguration des verwendeten Webserver findet nicht statt. Beim Endprodukt wird der Webserver vollständig konfiguriert, es sind z.B. nur die benötigten Ports freigeschaltet.

Verlust der Vertraulichkeit sensibler Daten: Die Übertragung der eingegebenen Daten im Prototyp findet ohne Verschlüsselung zwischen Server und Client statt. Ein Angreifer innerhalb des Krankenhausnetzwerks könnte die Daten mitloggen. Grundsätzlich wäre eine Kommunikation per https möglich. Hierfür muss während des Softwarebetriebes für den Server innerhalb des Krankenhaus-Netzwerks ein Zertifikat ausgestellt werden. Diese Maßnahme kann erst während des Betriebes getroffen werden.

Fehlerhafte Autorisierung auf Anwendungsebene: Im Prototyp gibt es nur eine Benutzergruppe, mit denselben Berechtigungen. Eine fehlerhafte Autorisierung kann somit nicht stattfinden. Das Webfrontend muss sich bei Backend autorisieren, dies findet im Prototyp nach dem Login des Benutzers in das System statt. Ohne Benutzerkennung kann nicht auf den Prototyp zugegriffen werden.

Cross-Site Request Forgery: JSF 2.2 schützt vor Cross-Site Request Forgery. Ein Versuch führt zu einer `ViewExpiredException`.

Nutzung von Komponenten mit bekannten Sicherheitsschwächen: Die verwendeten SOUPs werden grundsätzlich auf Sicherheitsschwachstellen untersucht. Eine genauere Analyse der SOUPs ist jedoch bei der Implementierung der Produktiv-Version des Webfrontends nötig.

Ungeprüfte Um- und Weiterleitungen: Der Gesamte Prototyp, abgesehen vom Login-Mock Up befindet sich auf einer Seite. Es gibt keine Um und Weiterleitungen auf andere URLs.

5.2 Systemarchitektur

Im Rahmen des Softwareentwicklungsprozesses fordert die EN 62304 in Kapitel 5.3 das Design der Software-Architektur. Vom Hersteller muss sichergestellt werden, dass die Anforderungen an die Medizinprodukte-Software in eine strukturierte Architektur umgesetzt werden. Für Software der Klasse C, wie das GlucoTab-System müssen hierzu Software-Komponenten identifiziert und die Architektur der Schnittstellen zwischen den Software-Komponenten sowie den Softwarekomponenten außerhalb des Systems definiert werden. In weiterer Folge müssen für SOUP-Komponenten Funktions- und Leistungsanforderungen spezifiziert werden. Um eine Risikobeherrschung sicherzustellen, muss vom Hersteller in allen genannten Tätigkeiten eine Abgrenzung zwischen den Software-Komponenten festgelegt werden. Diese Architektur von Komponenten und Schnittstellen muss für medizinische Software der Klasse C während des detaillierten Software-Designs solange unterteilt werden bis sie durch Software-Einheiten dargestellt werden kann. Der Hersteller ist zur Verifizierung und Dokumentation dieser Architektur verpflichtet. [49]

In den folgenden Unterkapiteln wird ein Einblick in die Architektur des Gesamtsystems und die Schnittstellen zwischen dem Prototyp und dem restlichen System gegeben. Weiters wird ein Überblick über die Software-Einheiten des Prototyps geboten.

5.2.1 Gesamtsystem

Das GlucoTab – System besteht aus Android-Frontend, Web-Frontend und dem Backend. Der Benutzer interagiert mit dem System entweder durch das Android-Frontend auf dem Tablet oder durch das Web-Frontend welches im KIS des Benutzers integriert werden soll. Das Android-Frontend verwendet zur Kommunikation mit dem Backend SOAP (Simple Object

Access Protocol) Webservice-Aufrufe über eine WIFI-Verbindung. Das Web-Frontend ist am selben Server installiert wie das Backend und kommuniziert mit dem Backend über die Service-Schicht. Die Anwendungsdaten werden vom Backend in der GlucoTab Datenbank gespeichert. Weiters sichert das Backend vom Benutzer generierte PDF-Berichte am Filesystem ab. Zur Benutzer-Authentifikation kann das GlucoTab-System optional auf ein *ActiveDirectory* zugreifen. Zur Aktualisierung der Medikamentendaten und der Konfiguration kann das GlucoTab-System auf ein SVN zugreifen. Das KIS stellt dem Gesamtsystem über die HL7 Schnittstelle Patienten und Labordaten zur Verfügung. In Abbildung 13 wird das System mit seinen Schnittstellen nach außen hin dargestellt.

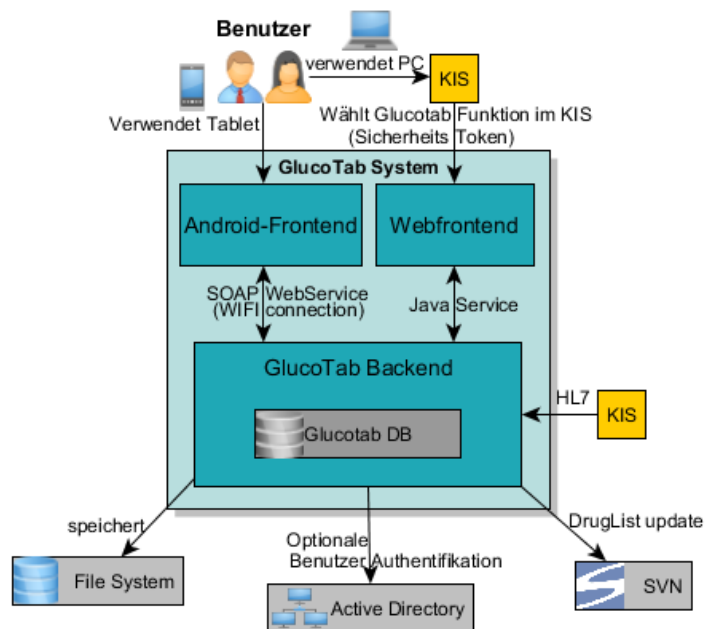


Abbildung 13: GlucoTab Gesamtsystem

Im Rahmen dieser Masterarbeit wurde ein Prototyp des Web-Frontends ohne KIS-Integration entwickelt, sowie Teile der Logik im Backend mitentwickelt. Auf die Architektur und das Design der Softwareteile wird im Folgenden präziser eingegangen. Alle anderen Teile des GlucoTab-Systems werden als Blackbox betrachtet, die verwendeten Services werden in 5.2.3 als Schnittstellen kurz vorgestellt.

5.2.2 Architektur des Web-Frontend-Prototyps

JSF führt wie in 4.3.1 beschrieben zur Verwendung der Model-View-Controller Architektur. Wie diese Architektur konkret im Prototyp umgesetzt wurde ist in Abbildung 14 dargestellt. Das `FacesServlet` wurde unverändert vom JSF-Framework übernommen und verwaltet die *Requests* vom und die *Responses* zum Client. Die View-Komponente setzt sich grundsätzlich aus dem Layout der Hauptansicht mit den Templates für die einzelnen Abschnitte zusammen. In diese Templates sind Dialoge für die Durchführung von Funktionen, wie der Medikamentenverordnung, Änderung der Therapieparameter, Therapiewechsel oder Dokumentation der Vortherapie integriert. Diese Dialoge sind modular aufgebaut und setzen sich aus einer oder mehrerer *Compositions* zusammen. Die darzustellenden Informationen werden vom Model bereitgestellt und aufbereitet. Das Model wurde in sieben *packages* eingeteilt welche funktional zusammengehörende Beans zusammenfassen.

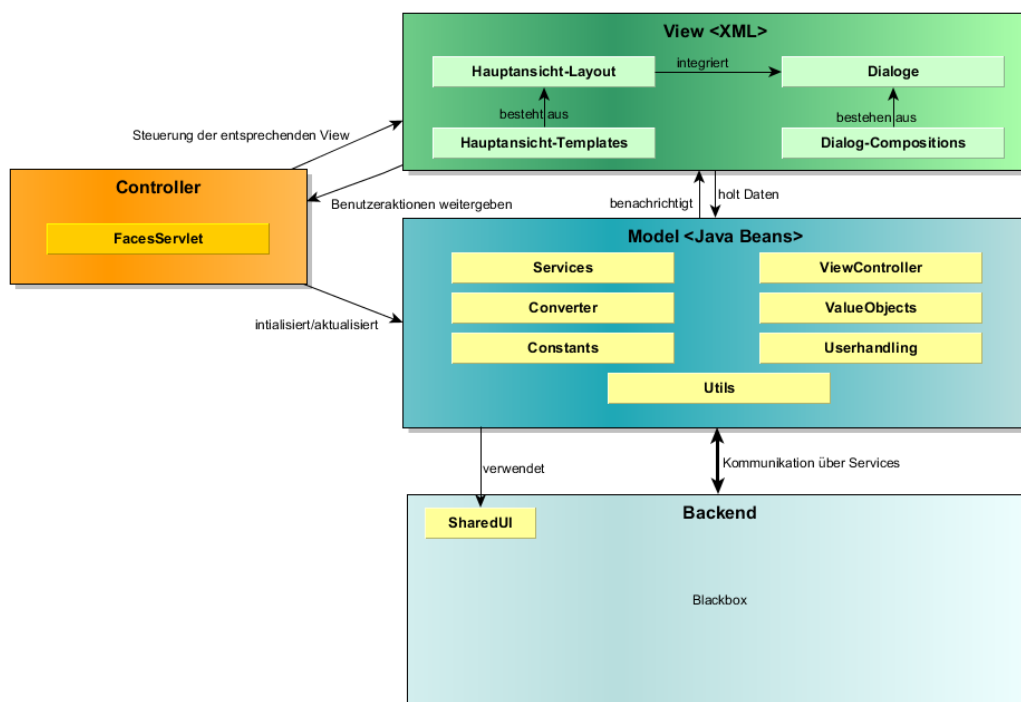


Abbildung 14: Umsetzung der Model-View-Controller Architektur im Prototyp

Die Kommunikation mit dem Backend erfolgt wie in 5.2.3 beschrieben über Services. Weiters sind in beiden Frontends verwendete Funktionen zur Therapieanpassung in der `SharedUI`-Bibliothek umgesetzt. Eine Beschreibung dieser Bibliothek befindet sich in 5.2.4.

5.2.2.1 View-Komponenten

Im Prototyp stellt die Therapieanpassungsansicht die Hauptansicht des Web-Frontends dar. Diese Hauptansicht wird durch modale Dialoge für die Durchführung einzelner Arbeitsschritte während des Therapieanpassungs-Workflows erweitert. Das Layout der Therapieanpassungsansicht wird durch `mainLayout.xhtml` modular in drei Bereiche eingeteilt. Diese drei Bereiche werden durch die Inhalte aus den Dateien `header.xhtml`, `therapySettingsAndDiagram.xhtml` und `adjustTherapySettings.xhtml` befüllt. Die Aufteilung der Ansicht in ihre Module ist in Abbildung 15 ersichtlich. Das Einbeziehen von Inhalten aus anderen Quelldateien erfolgt im Prototyp durch den `ui:insert` Codeblock. Im obersten Bereich der Ansicht befindet sich das Modul `header.xhtml`, hier werden die patientenbezogenen Informationen, wie Name, Alter, Aufnahmezahl, sowie der Benutzername des aktuell eingeloggtten Benutzers dargestellt. Diese Informationen sind zu jedem Zeitpunkt sichtbar um Verwechslungen der Patienten zu vermeiden.

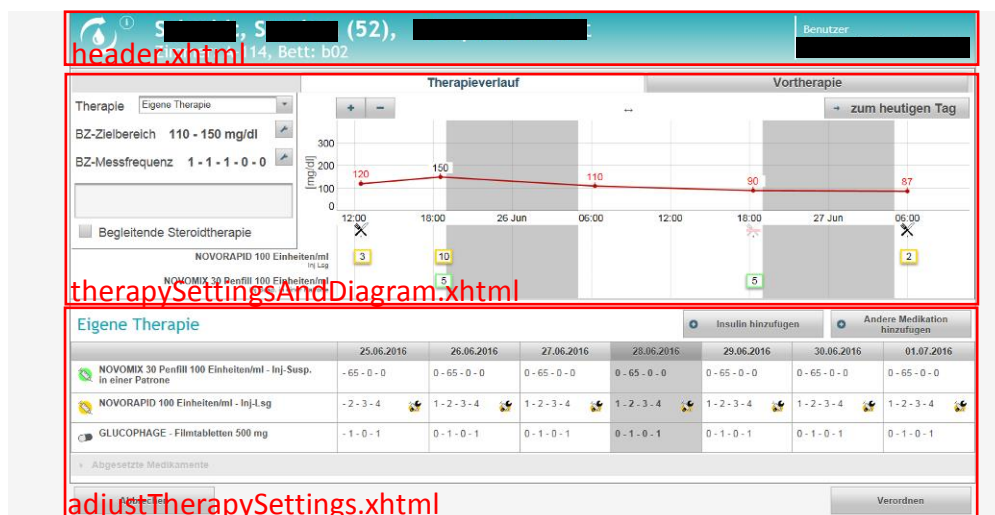


Abbildung 15: Layout der Therapieanpassungsansicht

Im Abschnitt welcher durch `therapySettingsAndDiagram.xhtml` definiert wird befinden sich im linken Bildschirmbereich zum einen das durch Dygraphs realisierte BZ-Verlaufdiagramm, mit den verabreichten Medikamentengaben und zum anderen die Informationen bezüglich der Vortherapie des Patienten. Im rechten Bildschirmabschnitt sind die Therapieparameter wie die Therapieart, der BZ-Zielbereich, die BZ-Messfrequenz sowie die begleitende Steroidtherapie dargestellt. Je nach Therapieart können diese Parameter vom Benutzer bearbeitet werden.

Im untersten Bildschirmabschnitt befindet sich die virtuelle Fieberkurve, welche die Anpassung der verordneten Medikation ermöglicht. Sollten sich abgesetzte Medikamente im Verordnungsverlauf des Patienten befinden werden diese in einer optionalen zusätzlichen Tabelle am unteren Ende der virtuellen Fieberkurve angehängt. Diese Komponenten der View sind in `adjustTherapySettings.xhtml` definiert.

In der virtuellen Fieberkurve steht dem Benutzer ein Kontextmenü zum Bearbeiten der Verordnungen zur Verfügung. Über dieses Menü können Medikationen direkt pausiert, abgesetzt und Pausierungen wieder aufgenommen werden. Andere, komplexere, Funktionen wie das Bearbeiten oder hinzufügen von Verordnungen wurden modale Dialoge implementiert. Die Anpassung von Therapieparametern erfolgt im Detail ebenso in modalen Dialogen wie der Therapiewechsel. Auch für das Einschreiben neuer Patienten in das System und das Erfassen der Vortherapie sind in modalen Dialogen umgesetzt. Insgesamt sind 18 unterschiedliche modale Dialoge im Prototyp realisiert, welche sich wiederum aus bis zu drei Modulen zusammensetzten. Das Design des Prototyps wurde in drei Cascading Style Sheets (CSS) definiert.

Um die Schnittstelle zwischen View- und Model-Komponenten zu beleuchten wird kurz ein dafür verwendeter Code-Ausschnitt erläutert.

```
<h:outputText value="#{medicationHistoryHandler.selectedDrugVO.orderUnit}"/>
```

Die Ausgabe eines Texts im Frontend stellt hier die einfachste Variante einer View-Komponente mit Verbindung zum Model dar, hierfür wird die JSF-Standardkomponente `outputText` verwendet. Über das Attribut `value` wird durch das `selectedDrugVO`-Objekt des `medicationHistoryHandlers` auf den Parameter `orderUnit` des Objekts zugegriffen. Diese Art von Zugriff wird als *Value-Binding* bezeichnet. In diesem Fall wird im Frontend die Verabreichungseinheit des Medikaments, wie eine Tablette oder 1 ml, dargestellt.

5.2.2.2 Model-Komponenten

Das Model setzt sich aus 25 Java-Beans zusammen, welche in sieben `packages` eingeteilt sind. Die Aufteilung der Beans erfolgte anhand ihrer abgedeckten Funktionalität und ist in Abbildung 16 dargestellt. Ein vereinfachtes UML-Diagramm der Architektur der Model-Komponenten befindet sich im Anhang in Abbildung 27.

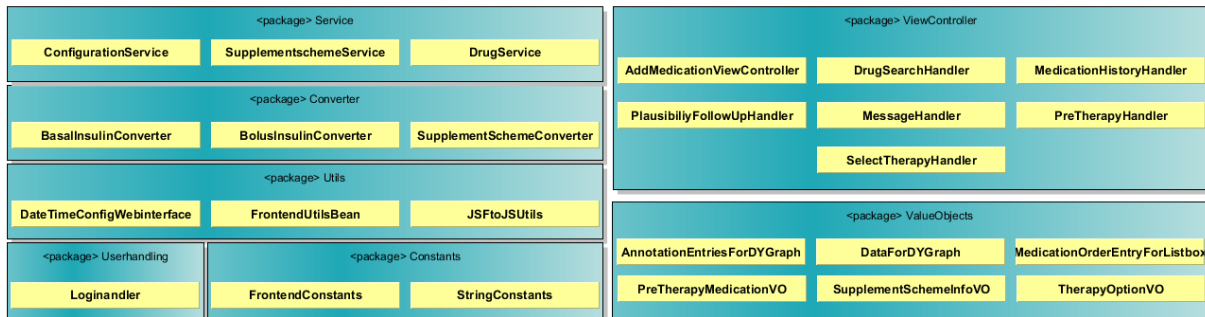


Abbildung 16: Übersicht über die in der Model-Komponente enthaltenen Beans

Die einzelnen `packages` und die enthaltenen Beans lassen sich wie folgend beschreiben:

Constants:

In der Klasse `FrontendConstants` werden für das Frontend spezifische Konstanten wie Verweispfade auf Bilder und andere Dateien, Identifier und ASCII-Codes für in der `View` verwendete Sonderzeichen bereitgestellt. In `StringConstants` werden als Konstanten verwendete Bausteine für die im `SharedUI` generierten Anzeige-Strings definiert.

Converter:

Die Primefaces-Komponente für dropdown-Menüs `p:selectOneMenu` verwendet eine Converter-Klasse um die im Model verwendeten Objekte in der View auswählbare Elemente umzuwandeln. Konkret wird der definierte Identifier des ausgewählten Elements mit Identifiern der verfügbaren Objekte verglichen, und je nach Methode entweder ein Anzeigetext für die Elemente oder das zum Element gehörende Objekt zurückgegeben. Die Klassen `BasalInsulinConverter`, `BolusInsulinConverter` und `SupplementSchemeConverter` dienen dazu die im Model und Backend verwendeten Java-Objekte, `drugVO` und `SupplementSchemeVO` zu konvertieren.

Service:

Das Service Package ist nicht mit den vom Backend bereitgestellten Services zu verwechseln. Vielmehr dienen die Beans in diesem `package` dazu, das Handling von im Frontend im Hintergrund verwendeten Funktionalitäten zu bündeln. Das `ConfigurationService` des Frontends holt sich vom `ConfigurationCoreService` des Backends im Frontend verwendete Konfigurationsparameter. Weiters dient es zur Verwaltung und Bereitstellung eben dieser Konfigurationsparameter im gesamten Prototyp. Das

`SupplementSchemeService` erledigt die Verwaltung von Korrekturschemata im Frontend. Es stellt vordefinierte Schemata aus der Konfiguration des Backends bereit und verarbeitet die vom Benutzer erstellten Schemata. Das `DrugService` stellt allgemeine Informationen im Dialog der Medikamentenauswahl, wie den Baum mit ATC-Codes und Darreichungsformen bereit, die eigentliche Medikamentensuche wird jedoch im `DrugSearchHandler` implementiert.

Userhandling:

Dieses Package beinhaltet nur die `LoginHandler` Klasse. Neben einem vereinfachten Login-Prototyp und der Registrierung des Frontends am Backend werden bei Auswahl eines Patienten im KIS-Mockup auch die Patientendaten und die Daten zum aktuellen Aufenthalt über das `PatientService` und das `EnrolmentService` abgerufen. Danach werden je nach Einschreibung des Patienten ins System die benötigten Dialoge der View aufgerufen.

Utils:

Durch dieses package werden dem restlichen Frontend häufig benötigte Werkzeuge zur Verfügung gestellt. `DateTimeConfigWebinterface` wird zur Instanziierung der `DateTimeUtilities` aus dem Backend benötigt, welches für die Synchronisation der Zeit im Front- und Backend sorgt. Wesentlich umfangreichere Funktionen stellt die Klasse `FrontendUtilsBean` bereit. So wird die Sichtbarkeit und Auswahl von Icons im Web-Frontend gesteuert, darzustellende Strings formatiert und bereitgestellt. Als wesentliches Steuerelement für das mit Dygraphs umgesetzte BZ-Verlaufsdiagramm dient die `JSFTToJSUtils`. Hier werden die vom Backend bereitgestellten Informationen zu Medikamentengaben, BZ-Messerwerten und eingenommenen Mahlzeiten in von Dygraphs verständliche JavaScript-Code-Ausschnitte übersetzt, welche dann in den in der View bereitgestellten JavaScript-Code zur Diagramm-Generierung eingefügt werden.

ViewController:

In diesem Package befinden sich die komplexesten Klassen des Prototyps. Der `AddMedicationViewController` stellt alle vom Model bereitgestellten Informationen beim Hinzufügen oder Bearbeiten von Medikationen zur Verfügung. Diese Klasse ist als Middleware zwischen den View-Dialogen und dem `MedicationHistoryHandler`

eingesetzt. Der `MedicationHistoryHandler` die Verwaltung und Bearbeitung des Therapieplans mit Hilfe des `AdjustTherapyController` aus der `SharedUI`-Bibliothek übernimmt. Im `MedicationHistoryHandler` werden alle Funktionen zur Bearbeitung der Therapie, wie das Absetzen, Pausieren und Hinzufügen von Intervall- und Einmalverordnungen umgesetzt. Beim Therapiewechsel wird auf die Funktionalität des `SelectTherapyHandlers` zurückgegriffen. Diese Klasse setzt die Veränderungen der Benutzeroberfläche beim Therapiewechsel um. Die Änderung des Therapieplans beim Wechsel wird jedoch durch dem `MedicationHistoryHandler` mit Hilfe des `AdjustTherapyHandler` umgesetzt. Die Schnittstelle zum Backend bezüglich der Medikamentensuche und Bereitstellung der Medikamentendaten im Frontend stellt der `DrugSearchHandler` dar. Für die Verwaltung der vorangegangenen Therapie der Patienten ist der `PreTherapyHandler` zuständig. Dieser greift ebenso wie der `MedicationHistoryHandler` auf die Funktionalität des `DrugSearchHandler` bei der Verarbeitung von Medikamentendaten zurück. Die Klassen `MessageHandler` und `PlausibilityFollowUpHandler` stehen in enger Verbindung und stellen die Einhaltung des vorgesehenen Prozessablaufs beim Aufruf der einzelnen Dialoge insbesondere bei auftretenden Fehlermeldungen sicher.

ValueObjects:

Durch die Klassen in diesem `package` wird eine Struktur für Objekte geschaffen welche Informationen im View bereitstellen. Die Klassen `AnnotationEntryForDygraphs` und `DataForDygraphs` dienen zur Bereitstellung von im BZ-Verlaufdiagramm dargestellten Informationen. Sie werden durch die `Utils`-Klasse `JSFToJSUtils` befüllt und die enthaltenen Informationen in den JavaScript-Code der View integriert. Das Objekt welches durch die Klasse `MedicationEntryForListbox` erstellt werden kann wird zur Auswahl der Verordnung bei mehreren Verordnungen an einem Tag genutzt. Bei der Verwaltung der Vortherapie werden die einzelnen Vortherapie-Verordnungen in `PreTherapyMedicationVO`-Objekten zwischengespeichert bevor sie in das vom Backend verwendete Objekt umgesetzt werden. Für die Auswahlliste der Therapien beim Therapiewechsel wird ein Objekt benötigt. Dieses Objekt wird durch das `TherapyOptionVO` umgesetzt. Es stellt den Titel, das verwendete Icon, und die Information ob in der Therapieform die Entscheidungsunterstützung bei der Dosisanpassung verfügbar ist, zur Verfügung. Zur Darstellung der Informationen des

Korrekturschemas im Frontend wird das `SupplementSchemeInfoVO` verwendet. Dieses Objekt wird von `SupplementSchemeUtils` befüllt und bereitgestellt.

5.2.3 Schnittstellen zum Backend

Der Datenaustausch zwischen dem Backend und dem Prototyp wird komplett über dafür vorgesehene *Services* durchgeführt. Dadurch bleibt der Prototyp modular und prinzipiell unabhängig von der Entwicklung des Backends, solange die *Services* als definierte Schnittstellen unverändert bleiben. Die folgenden *Services* wurden im Prototyp verwendet:

ConfigurationCoreService:

Stellt die Schnittstelle für die Frontends zum Laden von Parametern aus der Systemkonfiguration während der Laufzeit dar. Diese Konfiguration kann sich von Abteilung zu Abteilung unterscheiden. Im Prototyp wurden zum Beispiel Konfigurationsparameter für die vordefinierten Korrekturschemata, Informationen zur Internationalisierung des Systems, wie Datums-Formate, verwendete Einheitensysteme oder Grenzwerte für Plausibilitätschecks der Benutzereingaben geladen.

DoseCalculationService:

Dient zur Berechnung von Dosisvorschlägen für die verordneten Medikamente. Dies wird einerseits in der Basis/Bolus-Therapie zur Berechnung der Tagesdosis verwendet sowie bei der Medikamentengabe für die Berechnung der Gesamtdosis des zu verabreichenden Medikaments. Dies ist zum Beispiel bei einem verordneten Korrekturschema nötig. Im Prototyp des Web-Frontends wurde nur die Funktion zur Berechnung der Tagesdosis verwendet, da die Funktion der Medikamentengabe nur im Android-Frontend zur Verfügung steht.

DrugService:

Mit Hilfe des `DrugService` kann nach Medikamentendaten gesucht sowie die Medikamentenliste geladen werden. Zusätzlich zur kompletten Medikamentenliste wird auch eine Liste der meist verwendeten Medikamente bereitgestellt.

EnrolmentService:

Durch das `EnrolmentService` ist es möglich Patienten in das GlucoTab-System einzuschreiben, die Aufnahmedaten zu aktualisieren und die Aufnahme wieder zu beenden. Bei der Aufnahme eines Patienten in das System werden auch die initialen Therapieeinstellungen generiert. Im Prototyp wurden die Funktionen für das Starten und Verändern der Aufnahmedaten verwendet, das Ausschreiben eines Patienten aus dem System wurde nicht implementiert.

PatientService:

Dient zur Suche nach Patienten, dem Laden der detaillierten Patientendaten sowie der Informationen der aktuellen Aufnahme des Patienten in der Krankenanstalt. Diese Patientendaten stammen ursprünglich aus dem KIS aus welchem sie mittels eines anderen Service über eine HL7-Schnittstelle in die Backend-Datenbank übermittelt werden.

TherapyCoreService:

In diesem Service ist die gesamte Logik zur Verwaltung der Therapie bzw. des Therapieplans im Backend implementiert. Hier werden zum Beispiel therapiebezogene Aufgaben für das klinische Personal generiert oder die vom `AdjustTherapyController` generierten `OrderEvent`-Objekten verarbeitet. Die in den Frontends getätigten Änderungen im Therapieplan werden durch dieses Service im Backend entsprechend umgesetzt.

5.2.4 Shared-UI Bibliothek

Da sich das Android- und das Web-Frontend Teile der Logik zur Erstellung, Darstellung und Veränderung des Verordnungsverlaufes teilen wurden diese in eine geteilte Bibliothek (Shared Library) ins Backend ausgelagert. UML- Diagramme der Bibliothek befinden sich im Anhang unter 11.1. Diese Bibliothek besteht aus drei Softwareeinheiten:

5.2.4.1 AdjustTherapyController

Der `AdjustTherapyController` ist das Kernstück der *Shared-UI* Bibliothek. Er verarbeitet und verwaltet die `MedicationOrderCourses` des `TherapySettingsHistoryVOs`. Über den Konstruktor wird dem `AdjustTherapyController` der aktuelle Stand der verordneten Insuline und nicht-Insuline übergeben. Für alle vom Benutzer getätigten Änderungen am Medikationsplan während der Therapieanpassung stellt der *AdjustTherapyController*

entsprechende Funktionen zur Verfügung, welche von den Frontends aufgerufen werden müssen. So werden zum Beispiel zum Hinzufügen neuer Medikamentenverordnungen entweder die Methode `setIntervalOrder`, für Intervallverordnungen, oder `addSingleOrder`, für Einmalgaben, verwendet. Der `AdjustTherapyController` stellt zusätzlich zu den Methoden zum Bearbeiten der Medikamentenverordnungen auch die Logik beim Wechsel von der Basis/Bolus-Therapie in die eigene Therapie und umgekehrt zur Verfügung. Bei Abschluss der Therapieanpassung vergleicht der `AdjustTherapyController` die `MedicationOrderCourses` vor Beginn der Anpassung mit den aktuellen und erstellt in der Methode `buildUpdateRequest` die entsprechenden `OrderEvents`, welche dem Backend mit der Methode `updateCurrentTherapySettings` des `TherapyCoreService` übergeben werden.

5.2.4.2 SharedUIUtils

Die `SharedUIUtils`-Klasse stellt den Frontends Methoden zur korrekten Formatierung von Zahlen, Brüchen, der BZ-Messfrequenz sowie zur Berechnung des Alters bereit. Durch das Teilen dieser Methoden soll eine konsistente Darstellung der formatierten Werte in allen Frontends sichergestellt werden. Die Methoden von `SharedUIUtils` werden auch zur Internationalisierung des Systems verwendet. Hierzu werden den Methoden zur Formatierung von Zahlen die entsprechenden `NumberFormats` aus dem `ConfigurationService` übergeben.

5.2.4.3 SupplementSchemeCalculator

Vordefinierte Schemata		normal
ab 141mg/dl + 2 IE / 40mg/dl + 4 IE		
ab	141 mg/dl	<ul style="list-style-type: none"> • 141 - 181 mg/dl → + 4 IE • 181 - 221 mg/dl → + 6 IE • 221 - 261 mg/dl → + 8 IE • 261 - 301 mg/dl → + 10 IE • 301 - 341 mg/dl → + 12 IE • 341 - 381 mg/dl → + 14 IE • 381 - 421 mg/dl → + 16 IE
→ 2 IE pro	40 mg/dl	
bis	400 mg/dl	
Start mit	4 IE	

Abbildung 17: Darstellung des Korrekturschemas im Prototyp

Im Backend werden die Parameter der Korrekturschemata gespeichert. In den Frontends wird bei der Anzeige, beziehungsweise bei der Bearbeitung der Korrekturschemata zusätzlich eine Liste mit den einzelnen Korrekturstufen dargestellt. Diese Liste ist in Abbildung 17

hervorgehoben. Um die Darstellung der Liste in allen Frontends einheitlich zu gestalten wurde die `SupplementSchemeCalculator` - Klasse erstellt. aus den Korrekturschema-Parametern erstellt der `SupplementSchemeCalculator` eine Liste von Datensätzen. Im Web-Frontend-Prototyp werden diese Datensätze in eine Liste von `SupplementSchemeInfo`-Objekten umgewandelt, um dann mit der Primefaces-Komponente `dataList` dargestellt zu werden.

5.3 Implementierung

Die EN 62304 fordert im Software-Entwicklungsprozess, für Software der Klassen B und C, eine Implementierung der Software, gemäß der Softwarearchitektur und dem detaillierten Softwaredesign. Im Rahmen der Implementierung der im Softwaredesign definierten Softwareeinheiten muss eine Verifikation dieser Softwareeinheiten sichergestellt werden. Zur Verifikation der Softwareeinheiten müssen zuvor definierte Akzeptanzkriterien eingehalten werden. Dementsprechend durchläuft jede Softwareeinheit im Backend und im Android-Frontend sogenannte Unit-Tests. Im Prototyp des Web-Frontends wurde auf die Durchführung von Unit-Tests verzichtet, da diese Software-Teile im Wesentlichen über wenig komplizierte Logik verfügen. Bei der produktiven Entwicklung des Web-Frontends müssen Unit-Tests durchgeführt werden um dem in der Norm definierten Entwicklungsprozess zu entsprechen. Bei der Implementierung aller Softwareeinheiten im GlucoTab-System mussten zudem, zur Qualitätssicherung des Codes, Codingstandards eingehalten werden.

Zur Dokumentation der Nachvollziehbarkeit und Verwaltung des Entwicklungsprozesses wurde *JIRA*, ein Projektmanagement-Tool von Atlassian (London, Vereinigtes Königreich) verwendet. *JIRA* ist eine webbasierte Anwendung zur Verwaltung aller im Entwicklungsprozess auftretenden Ereignisse und eignet sich daher zum Aufgaben-, Fehler- und Qualitäts-Management. Während der Entwicklung des GlucoTab-Systems wurde *JIRA* zur Dokumentation und Verwaltung der in Kapitel 4.2 beschriebenen Sprintiterationen verwendet. Während des Sprintplans wurden in *JIRA* Entwicklungsaufgaben (*Issues*) zum aktuellen Sprint hinzugefügt. Im Laufe des Sprints wurden diese *Issues* bei Erfüllung der Aufgabe mit Verweis auf den implementierten Code geschlossen. Beim Sprintabschluss wurden die geschlossenen *Issues* von gesamten Entwicklungsteam überprüft. Parallel zur Dokumentation der Softwareentwicklung wurde in *JIRA* ebenso die Erstellung der technischen Dokumentation verwaltet.

In den zwei folgenden Unterkapiteln wird die Implementierung zweier essentieller Funktionen des Prototyps beschrieben.

5.3.1 Implementierung des Blutzuckerverlaufsdigramms

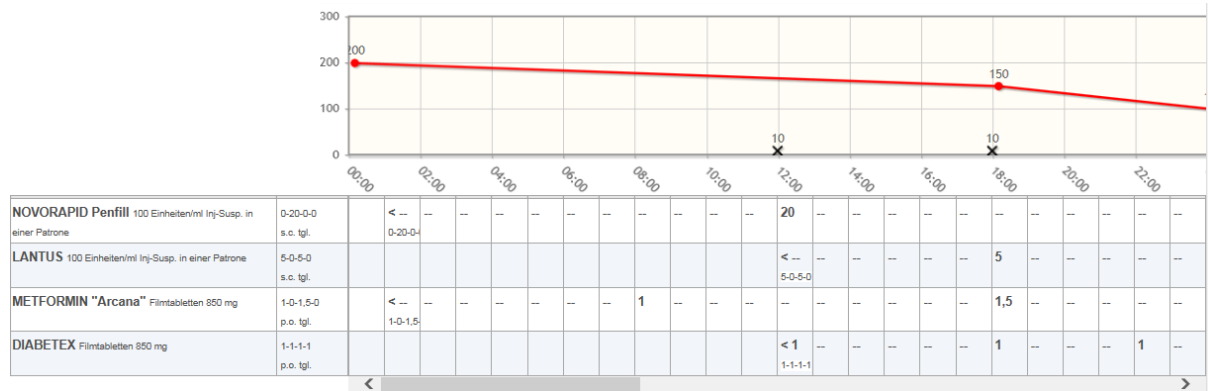


Abbildung 18: Prototyp des BZ-Verlaufsdigramms und der Verabreichungshistorie

Zur Darstellung des Verlaufs der BZ-Werte und der Historie der Medikamentengaben wurden unterschiedliche Lösungen in mehreren Prototypen umgesetzt. Neben dem Versuch der Darstellung der Historie der Medikamentengaben in Tabellenform, dargestellt in Abbildung 18, wurde auch eine Umsetzung mit der in Primefaces integrierten Lösung für Diagramme angestrebt. Die in Primefaces integrierte Diagram-Bibliothek stellte jedoch keine Möglichkeit zur Anzeige von vom Programm generierten Textfelder (*Annotations*) zur Verfügung. Dieses Feature ist essentiell für die Darstellung der Medikamentenverabreichungshistorie. Nach einer Recherche zu anderen Lösungen wurden die Diagramme letztendlich durch Verwendung der in 4.3.3 beschriebenen Bibliothek Dygraphs implementiert.

Da Dygraphs eine reine JavaScript-Bibliothek ist, war es nötig eine Schnittstelle zwischen der JSF-Anwendung und dem Dygraphs-Code zu erstellen. Dies geschah durch die Implementierung der in 5.2.2.2 beschriebenen Klasse `JSFTtoJSUtils`. Diese Klasse generiert aus den vom Backend übermittelten Informationen zu den BZ-Messwerten, eingenommenen Mahlzeiten und verabreichten Medikamente Eingabedaten für die Dygraphs Bibliothek.

Sowohl zur Übergabe der Eingabedaten als auch zur automatischen Erstellung der für die Darstellung der Medikamentengaben benötigten Diagramme müssen Primefaces und JSF-Komponenten in den JavaScript-Code eingefügt werden. Ein beispielhafter Codeausschnitt zur Veranschaulichung der Lösung befindet sich im Anhang unter 11.2.

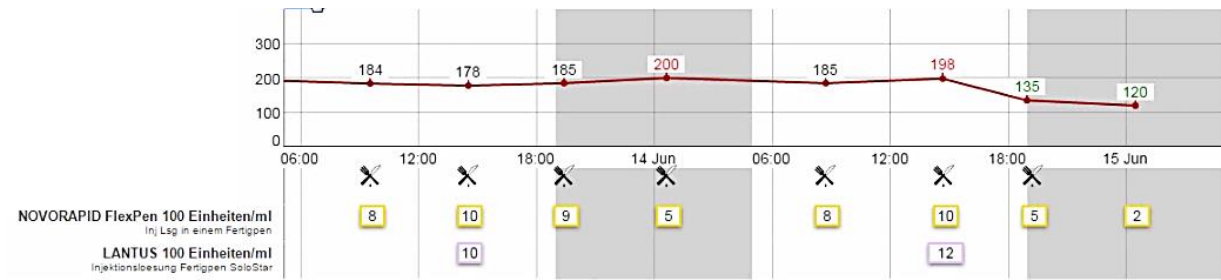


Abbildung 19: BZ-Verlaufdiagramm und Historie der Medikamentengaben

In Abbildung 19 ist die Endfassung dieser Diagramme dargestellt. Für die Darstellung der BZ-Verlaufskurve wird ein Diagramm generiert, ebenso für die Darstellung der eingenommenen Mahlzeit und für jedes Medikament in der Verabreichungshistorie. Die Anzahl der generierten Diagramme ist also abhängig von der Anzahl der während dieser Aufnahme im Krankenhaus verabreichten Medikamente.

5.3.2 Internationalisierung

Um Softwaresysteme unabhängig von unterschiedlichen regionalen Gegebenheiten, wie Einheitensystemen und der Sprache, zu machen werden sie so entwickelt das eben diese länderspezifischen Elemente modular und leicht austauschbar sind. Im Prototyp wurden alle angezeigten Elemente im Hinblick auf dieses Problem entworfen. Das verwendete Einheitensystem, das Datums- und das Zahlenformat werden über das `ConfigurationCoreService` aus der Konfigurationsdatei des Backends geladen. Alle Daten und numerischen Werte im Prototyp werden diesen Formaten entsprechend formatiert.

Texte werden zur Internationalisierung in sogenannten *Resource-Bundles* bereitgestellt. In beiden Frontends werden dieselben Texte angezeigt. So liegt es nahe auf dasselbe *Resource-Bundle* zuzugreifen. Das in Android-Anwendungen verwendete *Resource-Bundle strings.xml*, im XML-Format gehalten, unterscheidet sich jedoch im Aufbau von dem in JSF-Anwendungen verwendeten `messages.properties`. Dieses Problem kann über das Parsen der `strings.xml`-Datei behoben werden. Während der Entwicklung des Prototyps wurde dieser Parser noch nicht implementiert, vielmehr wurde das Android-*Resource-Bundle* manuell, halbautomatisch unter der Verwendung eines Texteditors in das im Web-Frontend benötigte Format umgewandelt.

5.4 Testen der Software

Der gesamte aufsteigende Ast auf der rechten Seite des in Abbildung 6 dargestellten V-Modells beschreibt die geforderte Verifikation und Validierung von Software als Medizinprodukt. Was dokumentiert getestet werden muss ist wiederum von der Sicherheitsklasse der Software abhängig.

Wie in Kapitel 5.2 beschrieben, fordert die EN 62304 [49], für Software der Klasse C, die Aufteilung des Software-Systems in Softwarekomponenten, welche wiederum aus kleineren Softwareeinheiten bestehen. Während der Implementierung dieser Softwareeinheiten sind Tests für jede Einheit, sogenannte Unit-Tests, gefordert. Im Rahmen der Entwicklung des Prototyps wurden Unit-Tests nur bei Softwareeinheiten in der mitentwickelten *Shared-UI* Bibliothek implementiert. Im Frontend-Prototyp an sich wurden keine Softwaretests durchgeführt. Im Rahmen der Systemprüfung muss die Gebrauchstauglichkeit des Softwaresystems nachgewiesen werden. Dies geschieht durch Einhaltung des gebrauchstauglichkeitsorientierten Entwicklungsprozesses, der Validierung der Gebrauchstauglichkeit durch einen Usability-Test und anschließende Feldbeobachtung. Im Laufe der Entwicklung des Prototyps wurde der gebrauchstauglichkeitsorientierte Entwicklungsprozess eingehalten und ein summativer Usability-Test durchgeführt. Der Usability-Test ist in Kapitel 7 dokumentiert.

In der produktiven Entwicklung des GlucoTab Systems werden alle geforderten Softwaretests umgesetzt. Die Softwareeinheiten werden mit Unit-Tests validiert. Die Schnittstellen der Komponenten durch Integrationstests geprüft. Und bei Release der Software werden Systemtests, zur Validierung der Funktion des Gesamtsystems, durchgeführt. Sowohl die Integrations- als auch die Systemtests werden auf dem mit JIRA verbundenen Bamboo-Testserver (Atlassian, London, Vereinigtes Königreich) durchgeführt.

Überprüfung der Gebrauchstauglichkeit

„Gebrauchstauglichkeit ist die Eignung eines Produktes, seinen bestimmungsgemäßen Verwendungszweck zu erfüllen. Sie ergibt sich aus dem Zusammenwirken von Mensch und Technik in einem Arbeitssystem.“ [72]

Im Rahmen des gebrauchstauglichkeitsorientierten Entwicklungsprozesses wurden dem Personal der medizinischen Universität während der wöchentlichen Treffen Prototypen der Benutzeroberfläche des Webinterfaces vorgestellt. Durch diese Diskussion wurden Verbesserungsvorschläge gesammelt, welche in die weitere Entwicklung einbezogen wurden. Dieses Vorgehen der Benutzereinbeziehung in die Entwicklung entspricht dem Prinzip der partizipatorischen Entwicklung welches in der EN 62366 vorgeschlagen wird [56].

Mit einem fortgeschrittenen Prototyp wurde eine heuristische Evaluation zur Identifikation von Usability-Mängeln und vorhandenen Fehlern durchgeführt. Das Ergebnis dieser Evaluation floss in die Entwicklung des finalen Prototyps des Web-Frontends ein. Die heuristische Evaluation wird in Kapitel 6 ausführlich dokumentiert.

Um die Gebrauchstauglichkeit des Web-Frontend-Prototyps zu validieren wurde abschließend ein summativer Usability-Test mit Ärzten der Medizinischen Universität Graz durchgeführt. Die Ergebnisse dieses Usability-Tests sollen in die produktive Entwicklung des Web-Frontends und des Android Clients miteinfließen. Die detaillierte Beschreibung des abschließenden Usability-Tests findet sich in Kapitel 7.

6 Heuristische Evaluation

Mit einer fortgeschrittenen Entwicklungsstufe des Prototyps wurde eine heuristische Evaluation durchgeführt. Ziel der Evaluation war es Schwachstellen in der Bedienbarkeit und im Design des Prototyps zu identifizieren. Ein zusätzlicher Fokus wurde auf die noch im Prototyp enthaltenen Programmfehler gelegt, um diese bis zum abschließenden Usability-Test zu entfernen.

6.1 Evaluierungsmethodik

Zur Evaluation soll das System anhand der in Tabelle 1 beschriebenen Heuristiken bewertet werden. Dabei sollen sowohl positive als auch negative Eigenschaften, sowie Fehlfunktionen des Systems im Hinblick auf die Heuristiken beobachtet und dokumentiert werden. Die Dokumentation der Ergebnisse soll in dem bereitgestellten Textfile erfolgen. Jedes Ergebnis soll zusätzlich zu der Beschreibung mit einem Screenshot dokumentiert werden. Diese Screenshots sollen zur leichtern Zuordnung mit einer eindeutigen ID benannt und zu den jeweiligen Punkten in der Dokumentation angefügt werden.

Zunächst sollen sich die Usabilityexperten mit dem System vertraut machen. Anschließend soll das Web-Frontend für jede Heuristik einzeln nacheinander betrachtet und bewertet werden. Dadurch soll eine strukturierte Beurteilung des Web-Frontends unterstützt werden. Sollten Fehler oder unerwartete Reaktionen des Web-Frontends bemerkt werden, sollen diese in der Textdatei in einer eigenen Kategorie dokumentiert werden.

Tabelle 3: Bei der Evaluation verwendete Heuristiken, angelehnt an [73]

Nr.	Usability Heuristiken (In Anlehnung an die Liste von Jakob Nielsen [73])
1	Rückmeldung des Systems Das System soll dem Benutzer zum aktuellen Zeitpunkt Rückmeldung darüber geben womit es gerade beschäftigt ist (bei Verzögerungen: Verwendung von Busy-Cursor oder andere Benachrichtigungen)
2	Das System soll die Realität abbilden Verwendung der Sprache, gängiger Wörter und Phrasen der Benutzergruppe. Informationen sollen in natürlicher und logischer Weise dargestellt werden.
3	Umkehrbare Aktionen Fehler oder unbeabsichtigte Aktionen des Benutzers sollen so einfach wie möglich rückgängig gemacht werden können.
4	Konsistenz Das gleiche Wort, die gleiche Aktion, die gleichen Phrasen sollen immer dasselbe bedeuten und bewirken.
5	Fehlerprävention Das System soll durch sein Design Benutzerfehlern vorbeugen. Vermiedene Fehler sind besser als behobene. (in bestimmten Feldern nur die Eingabe von realistischen Werten, z.B. positive BZ-werte, Datumseingabe)
6	Intuitivität Bedienelemente und Funktionen des Web-Frontend sollen klar erkennbar sein. Die Bedienung des Systems sollte selbsterklärend und möglichst ohne zusätzliche Anleitung möglich sein.
7	Effizienz Die Bedienung des Systems sollte so einfach wie möglich sein. Unterstützung von Tastenkürzeln, Favoritenlisten usw. Formulare sollen mit der wahrscheinlichsten Option so weit wie möglich vorausgefüllt sein.
8	Minimales Design Dialoge sollen keine unnötigen Informationen enthalten. Das Web-Frontend soll nicht überladen sein und den Benutzer mit zu vielen Elementen auf einmal erschlagen. Wichtige Elemente sollen sofort ersichtlich sein.
9	Gute Fehlermeldungen und Benachrichtigungen Die Fehlermeldungen und Benachrichtigungen sollen dem Benutzer ermöglichen, Fehler zu erkennen und zu bewältigen. Dabei sollen die Meldungen einfach, präzise und konstruktiv sein.
10	Identifizierte Fehler Zusätzlich sollen Fehlfunktionen des Web-Frontends beobachtet und dokumentiert werden. Dies ist grundsätzlich kein Hauptziel einer heuristischen Evaluation, welche ja zur Überprüfung der Gebrauchstauglichkeit dient. Die erstmalige genaue Betrachtung des Prototyps durch mehrere Tester bietet jedoch auch die Gelegenheit unentdeckte Fehlfunktionen im System zu identifizieren und zu beheben

6.2 Benutzerprofile

Als Benutzergruppe des Web-Frontends ist das klinische Personal in Krankenanstalten, in der vorliegenden Version insbesondere Ärzte, anzusehen.

Durch das Web-Frontend soll die Einschreibung des Patienten in das GlucoTab-System, die Therapieüberwachung, Therapieanpassung, Wechsel der Therapieform sowie Dokumentation der Medikation in der Therapie ermöglicht werden.

6.3 Umfang der Evaluation

Die gesamte Funktionalität des Web-Frontends vom Einschreiben des Patienten mit Erfassung der Vortherapie, der Auswahl der Therapieform im Krankenhaus, der Anpassung dieser Therapie, Verwendung aller Therapieformen, Dokumentation der Medikation (Verordnung von Intervallverordnungen, Einzelgaben, Pausierung, Veränderung, sowie Absetzung der Medikamente) soll auf Gebrauchstauglichkeit überprüft werden. Für die Durchführung der Evaluation sollen sich die Testpersonen ca. 90 Minuten Zeit nehmen.

6.4 Usabilityexperten und Evaluationsumgebungen

Es wurden zwei Gruppen von Usabilityexperten zur Beurteilung des Prototyps gewählt. Die erste Gruppe setzt sich aus erfahrenen Personen mit Softwareentwicklungshintergrund zusammen. Die Fähigkeit zur Beurteilung des Systems in Bezug auf Heuristiken stammt in dieser Gruppe aus der Entwicklung und Weiterentwicklung mehrerer Softwareprodukte mit Benutzerschnittstelle. Im Rahmen der Entwicklung dieser Systeme führten diese Personen in der Vergangenheit schon andere Usability-Tests durch. Diese Gruppe legte zusätzlich noch speziellen Fokus auf die Identifizierung von technischen Fehlern in der Software.

Die zweite Gruppe setzte sich aus erfahrenen medizinischen Personal der medizinischen Universität Graz zusammen. Diese Gruppe betrachtete die Benutzerschnittstelle mit Fokus auf den klinischen Workflow und problematische Bedienelemente. Die Qualifikation dieser Gruppe als Usabilityexperten für diese heuristische Evaluation stellt die langjährige Erfahrung bei der Bedienung von medizinischer Software in der klinischen Routine dar.

Der soziodemographische Hintergrund der Probanden ist in Tabelle 6 zusammengefasst.

Tabelle 4: Soziodemographische Daten der Probanden der heuristischen Evaluation

Hintergrund / Anzahl der Probanden	Softwareentwicklung	Klinischer Hintergrund
	4	2
Altersdurchschnitt	37,75 Jahre	31 Jahre
Durchschnittliche Berufserfahrung	10,75 Jahre	4 Jahre
Davon Erfahrung mit der Durchführung heuristischer Evaluationen	4	1

Als Evaluationsumgebung sollen die Arbeitsplätze der Usabilityexperten dienen. Es sollen State-of-the-Art Internetbrowser verwendet werden. Im besonderen Fokus auf Microsoft Internet Explorer 10 bzw. Edge, da diese meist mit dem Betriebssystem im Krankenhaus ausgerollt werden und für die geplante Inplace-Anwendung im KIS am Ehesten in Frage kommen.

Als Bildschirmauflösung wurde während der Entwicklung des Web-Frontends eine Auflösung von mindestens 1280x720 Pixel festgelegt. Niedrigere Bildschirmauflösungen werden vom Prototyp nicht unterstützt.

6.5 Ziele der Evaluation

Durch die Evaluation sollen Schwachstellen des Systems in Bezug auf die Bedienbarkeit und Gebrauchstauglichkeit identifiziert und in weiterer Folge behoben werden. Dadurch soll die Benutzerfreundlichkeit des Web-Frontend erhöht werden. Weiters sollen durch den Test noch vorhandene Fehler identifiziert werden. Durch Behebung dieser Fehler soll es zu einer Verbesserung der Stabilität des Systems kommen.

6.6 Ergebnisse und Maßnahmen

Durch die heuristische Evaluation konnten insgesamt 36 Programmfehler und 81 potentielle Schwachstellen der Gebrauchstauglichkeit identifiziert werden. Sieben der 36 identifizierten Fehler standen im Zusammenhang mit älteren Browserversionen. Diese sieben Fehler wurden dokumentiert, jedoch im Rahmen der Weiterentwicklung des Prototyps nicht behoben. Eine genaue Auflistung der identifizierten Fehler findet sich im Dokument „HE_Ergebnisse.xls“ im

digitalen Anhang. Die Zuordnung der Fehler zu den betroffenen Programmkomponenten ist in Abbildung 20 dargestellt.

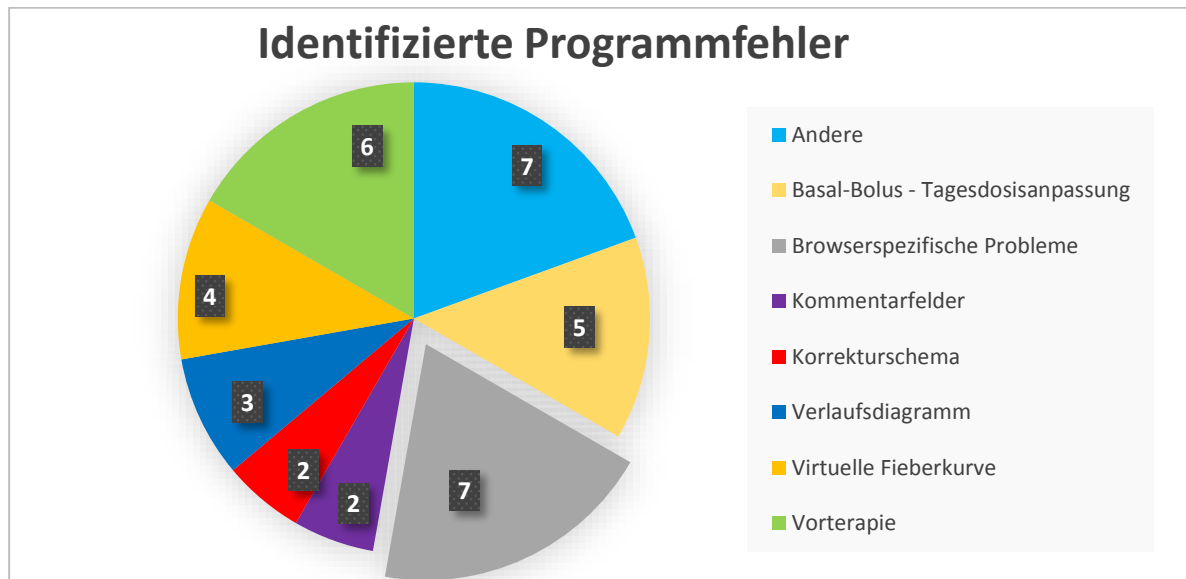


Abbildung 20: Aufteilung der identifizierten Programmfehler auf einzelne Komponenten des Prototyps

Das Hauptaugenmerk der heuristischen Evaluation lag auf der Identifikation von Schwachstellen der Gebrauchstauglichkeit. Die 81 von den Probanden identifizierten Mängel teilten sich wie in Abbildung 21 dargestellt auf die neun Heuristiken auf.

Mit 27 identifizierten Designproblemen war die Kategorie „Minimales Design“ jene mit den meisten Funden. Diese Probleme konnten meist jedoch sehr rasch durch Änderungen der Stylesheets behoben werden. Insgesamt wurde das Design dadurch konsistenter. Das Layout des Dialogs zur Bearbeitung von Medikamentenverordnungen wurde durch das Feedback aus dieser Kategorie angepasst. Am zweithäufigsten wurden, mit 17 Funden, Schwachstellen in der Intuitivität des Systems identifiziert. Diese Rückmeldungen führten zu einer Anpassung des Kontextmenüs und zu Anpassung der Positionen mehrerer Buttons. Weiters wurde die Navigation zwischen den Dialogen zur Medikamentensuche und der Neuverordnung von Medikamenten verändert. Insgesamt wurden 70 der 81 Schwachstellen bearbeitet. Die 11 Unbearbeiteten wurden dokumentiert, bei diesen Funden wurden Wünsche für zusätzliche Funktionen des Web-Frontends geäußert. Diese neuen Funktionen erfordern zum Teil neue Funktionen im Backend und andere waren mit großem Aufwand verbunden so dass auf diese bei der Weiterentwicklung des Prototyps verzichtet wurde.

Usabilityschwächen geordnet nach Heuristiken

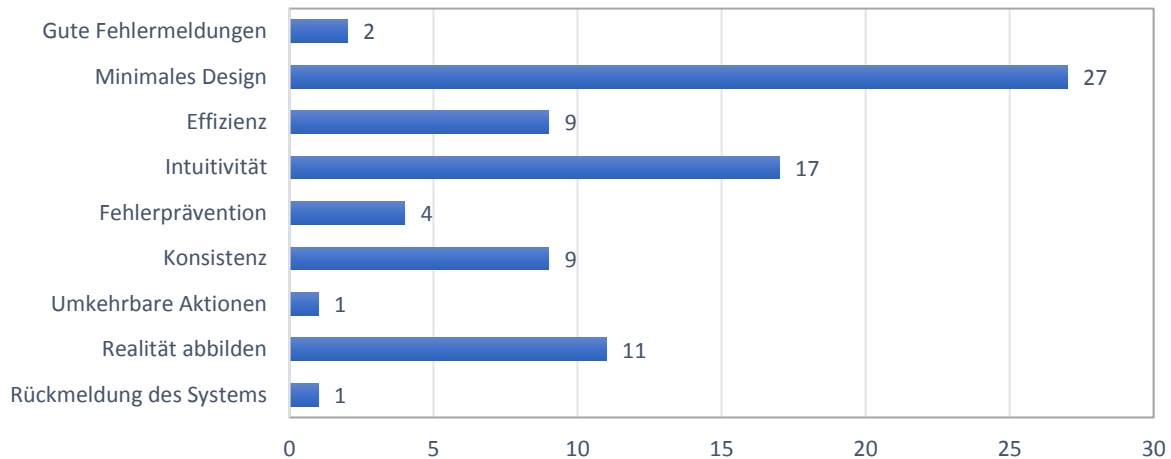


Abbildung 21: Aufteilung der identifizierten Usabilityschwächen nach den untersuchten Heuristiken

7 Abschließender summativer Usability-Test

Entsprechend der EN 62366 [56] wurde ein Plan für die Validierung der Gebrauchstauglichkeit für den Prototyp angelegt. Den Hauptteil der abschließenden Überprüfung des Prototyps stellt ein Benutzertest (Thinking Aloud) dar. In EN 62366 wird darauf verwiesen, dass Ergebnisse welche nur auf einer einzigen Bewertungsmethode beruhen unzureichend sein können. Zur Überprüfung von Gebrauchstauglichkeitsmängeln oder fehlenden Funktionen des Prototyps, welche nicht durch den Benutzertest bemerkt wurden, wurde zusätzlich noch ein Interview mit den Testpersonen durchgeführt. Dieses Vorgehen wird auch in [72] empfohlen, um ergänzende Informationen zur Bedienbarkeit der zu testenden Benutzeroberfläche zu erhalten. Zusätzlich soll durch dieses Interview ein Einblick auf die unterschiedliche Wahrnehmung der Benutzeroberfläche des Tablets und des Web-Frontend-Prototyps gewonnen werden.

7.1 Vorbereitung des summativen Usability-Tests

7.1.1 Testpersonen und Testumgebung

Als Umgebung für die Durchführung von Usability-Tests wird in der EN ISO 62366 ein Usabilitylabor, eine simulierte Arbeitsumgebung oder die wirkliche Umgebung des bestimmungsgemäßen Gebrauchs genannt [56]. Aufgrund der logistischen Situation und der kurzen Verfügbarkeit der Probanden wurden die Tests direkt am Computerarbeitsplatz der

Mediziner, also in ihrer eigentlichen Arbeitsumgebung, durchgeführt. Das Web-Frontend wurde auf einem vom Entwicklungsteam bereitgestellten Notebook ausgeführt.

Die Anzahl der benötigten Personen wird unterschiedlich eingeschätzt. Bei guter Abgrenzbarkeit der Benutzergruppe, d.h. Personal einer medizinischen Fachdisziplin, werden fünf Testpersonen sowohl von Nielsen [74] als auch von Backhaus [72] als ausreichend für die Durchführung eines Usability-Tests angesehen. Wie in Abbildung 22 ersichtlich sollen mit fünf Testpersonen rund 85% der Usability-Mängel gefunden werden. Eine größere Anzahl von Testern würde die Anzahl der gefundenen Mängel nur mehr langsam erhöhen. Der Nutzen wird entgegen dem Mehraufwand bei der Durchführung und Auswertung mit mehreren Testern als zu gering angesehen.[74]

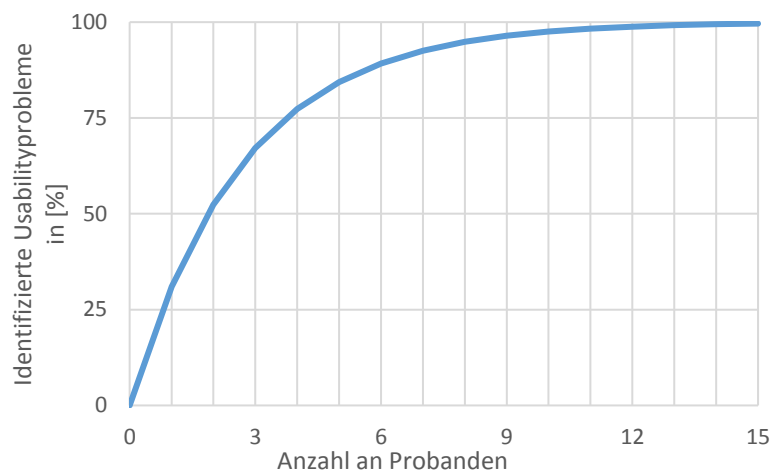


Abbildung 22: Identifizierte Usabilityprobleme in Abhängigkeit der Probandenanzahl nach Nielsen [74]

Dem widerspricht Faulkner in [75]. Bei einer Stichprobenanzahl von 5 Probanden belief sich der Anteil der identifizierten Usabilityprobleme im schlechtesten Fall nur auf 55%. Wie in Tabelle 5 werden erst ab einer Anzahl von 10 Probanden sicher über 80% der Usabilityprobleme identifiziert. Im FDA-Guidance Dokument für Usability Engineering bei Medizinprodukten wird eine Mindestzahl von 15 Probanden definiert [76].

Tabelle 5: Anteil der gefundenen Usabilityprobleme in Abhängigkeit von der Probandenanzahl in Prozent [75]

Probandenanzahl	Minimaler Anteil identifizierter Probleme in [%]	Mittlerer Anteil identifizierter Probleme in [%]	Standardabweichung
5	55	85,55	9,2957
10	82	94,686	3,2187
15	90	97,050	2,1207
20	95	98,4	1,6080
30	97	99,0	1,1343
40	98	99,6	0,8141
50	98	100	0

Für die Durchführung des Usability-Tests des Prototyps wurde aufgrund der Verfügbarkeit der Probanden und des entstehenden Aufwand/Nutzen-Verhältnisses eine Anzahl von fünf Probanden gewählt.

Die fünf Probanden sind als Mediziner an der Abteilung für Endokrinologie und Diabetologie der Medizinischen Universität Graz tätig. Sie verfügen alle über Erfahrung mit der Android-Benutzerschnittstelle des GlucoTab-Systems, waren jedoch nicht in die Entwicklung des Web-Frontend-Prototyps involviert. In Tabelle 6 ist ein Überblick über die soziodemographischen Daten der Probanden zu finden.

Tabelle 6: Soziodemographische Daten der Probanden

Geschlecht	männlich	weiblich
	3	2
Alter \bar{x}	39,6 Jahre	
Berufserfahrung	Minimum	Maximum
	6 Jahre	22 Jahre
Erfahrung mit GlucoTab	Minimum	Maximum
	2 Monate	3 Jahre

7.1.2 Definition der Gebrauchstauglichkeitsziele

Grundsätzlich gibt es mehrere Methoden zur Quantifizierung der Gebrauchstauglichkeit wie die Auswertung nach Gebrauchstauglichkeitszielen PROMEDIKS nach Backhaus oder UseProb nach Büchel. Die Verwendung von Gebrauchstauglichkeitszielen wird von

Zulassungsbehörden und in nationalen und internationalen Normen als optimale Vorgehensweise empfohlen [56].

Die Gebrauchstauglichkeitsziele werden entsprechend den Hauptbedienfunktionen definiert und stellen objektive Ziele für die Effektivität und Effizienz der Bedienung der Benutzeroberfläche dar. In Kapitel 5.1.2 wurden die Hauptbedienfunktionen während der Spezifikation der Anwendung definiert. Anhand dieser Hauptbedienfunktionen wurden Aufgaben und alle erforderlichen Durchführungsschritte definiert, diese Definition findet sich in Tabelle 9 im Anhang. Für die Erfüllung der Gebrauchstauglichkeitsziele mussten Teilschritte vollständig durchgeführt werden, andernfalls wurde die Aufgabe als negativ bewertet.

7.2 Durchführung

Der Prototyp wurde den Probanden kurz vorgeführt, es fand keine vollständige Einschulung statt. Durch dieses Vorgehen sollte die Intuitivität der Benutzeroberfläche im Besonderen bewertet und festgestellt werden, ob das System bei Problemen einen Lerneffekt begünstigt. Als Dauer wurden ungefähr 60 Minuten eingeplant. Der Usability-Test wurde mit **Thinking-Aloud** Methodik durchgeführt. Hierbei wird die Versuchsperson angehalten während der Durchführung des Usability-Tests seine Gedanken laut auszusprechen. So soll ein Eindruck über das geplante Vorgehen der Versuchsperson gewonnen werden. Durch diese Methode besteht jedoch auch die Gefahr der Ablenkung der Versuchsperson vom eigentlichen Usability-Test. [72] Im Anschluss an den Usability-Test wurde das Interview abgehalten.

7.2.1 Bewertungsmethodik

Der Usability-Test wurde für die Auswertung bezüglich Gebrauchstauglichkeitszielen geplant. Hierbei wurden nur jene Teilziele als positiv durchgeführt bewertet wenn sie von allen Probanden ohne Probleme durchgeführt werden konnten. Die Dauer der Durchführung wurde nicht bewertet, sobald ein Proband jedoch Probierverhalten bei der Durchführung einer Aufgabe zeigte wurde diese Aufgabe als negativ beurteilt.

Um einen Vergleichswert für die quantifizierte Gebrauchstauglichkeit zu erhalten wurden die Rohdaten des Ergebnisses sowohl nach Gebrauchstauglichkeitszielen bewertet als auch nach PROMEDIKs [72] Usability quantifiziert. Zur Auswertung der quantifizierten Usability wurde die Relevanz der Teilhandlungen anhand einer dreistufigen Skala (1 - geringe Relevanz, wird nicht häufig benötigt; 2 – mittlere Relevanz, Teilhandlung wird häufiger bzw. bei mehreren

Aufgaben benötigt; 3 – hohe Relevanz, Teilhandlung wird bei jeder Benutzung des Systems durchgeführt) bewertet. Ebenso wurden die Handlungen der Benutzer nach einer dreistufigen Skala bewertet, diese Einteilungscharakteristika sind in Tabelle 7 dargestellt. Die Prozessunterstützung nach PROMEDIKs wurde nicht ausgewertet da dafür benötigte Informationen nicht während der Usability-Tests erhoben wurden.

Tabelle 7: Drei-Stufen-Bewertungsverfahren als Bewertungsmaßstab für Usability-Evaluationen nach Backhaus [72]

Bewertungsstufe	Benutzerverhalten
3	Schwerer Bedienfehler, Handlung kann nicht ausgeführt werden
2	Leichter Bedienfehler, Probierverhalten mit Zielerreichung
1	Sicheres und schnelles Ausführen der Aufgabe

7.3 Ergebnisse Usability-Test und Interview

7.3.1 Auswertung nach Gebrauchstauglichkeitszielen

Die Auswertung des Usability-Tests zeigte, dass 45 der 55 Teilziele (81,8%) und 19 der 27 Gesamtziele (70,4%) positiv bewältigt werden konnten. Für die Bewertung der prozentuell erreichten Gebrauchstauglichkeitsziele liegt im Grunde keine Skala vor. Die erwähnten Ergebnisse sind in Tabelle 8 dargestellt, die Detailergebnisse befinden sich im Anhang unter 11.5 in Tabelle 11. Bewertet man diese Ergebnisse nach einem Schulnotensystem, beginnend mit einer mangelhaften Gebrauchstauglichkeit bei 60% der erreichten Ziele, und inkrementeller Verbesserung der Note bei der Steigerung um 10%, so wäre die Gebrauchstauglichkeit des Systems, nach Gebrauchstauglichkeitszielen, mit Befriedigend (für die Erreichung der Gesamtziele) bis Gut (für die Teilziele) zu bewerten. Im Wesentlichen soll der Fokus bei der Auswertung von Usability-Tests jedoch nicht auf der Beurteilung des Systems anhand einer Skala liegen, sondern bei der Analyse der aufgetretenen Gebrauchstauglichkeitsprobleme und deren Gegenmaßnahmen. Dies geschieht sowohl in Kapitel 7.3.4 als auch in Kapitel 8.

Tabelle 8: Auswertung der Gebrauchstauglichkeitsziele

	Anzahl	positiv	negativ	Positive in %
Teilziele	55	45	10	80,8
Gesamtziele	27	19	8	70,4

7.3.2 Auswertung nach PROMEDIKs quantifizierter Usability

Die quantifizierte Gebrauchstauglichkeit des Prototyps nach PROMEDIKs liegt bei 90,8%. Die Detailergebnisse befinden sich im Anhang in Tabelle 10. Nach dem Schulnotensystem liegt die quantifizierte Gebrauchstauglichkeit somit bei einem knappen „Sehr gut“. Um die Validierung der Gebrauchstauglichkeit nach PROMEDIKs vollumfänglich durchzuführen müsste noch eine Bewertung der Prozessunterstützung durchgeführt werden. Die Analyse der ermittelten Gebrauchstauglichkeitsprobleme befindet sich in 7.3.4.

7.3.3 Interview der Probanden

An dieser Stelle werden mehrere positive sowie negative Aspekte des Prototyps welche bei dem Interview der Probanden identifiziert wurden, genannt. In weiterer Folge wird ein Vergleich der Benutzeroberfläche des Prototyps mit der letzten Version der Android-Benutzeroberfläche angestellt.

Grundsätzlich zeigen sich die Probanden mit der Benutzeroberfläche und dem Funktionsumfang des Prototyps zufrieden. Das System wurde mehrfach als übersichtlich bezeichnet. Auch der größere Umfang zur Therapiedokumentation, im Vergleich zur letzten Version der Android-Anwendung, wurde positiv erwähnt. Zugleich wurde jedoch der insgesamt große Funktionsumfang als möglicherweise zu komplex für Benutzer außerhalb von Endokrinologie-Stationen betrachtet.

Negativ erwähnt wurde die Darstellung der Tagesgrenzen und die Unterscheidbarkeit der einzelnen Tage sowie die Identifizierbarkeit des Datums für die jeweiligen Tage bemerkt. Dies deckt sich mit den Ergebnissen bei der Durchführung des Usability-Tests. Zur Bedienung der virtuellen Fieberkurve wurde bemerkt, dass eine kurze Einschulung in das System hilfreich sein könnte. Bei Bekanntsein der Bedienung wird diese Lösung als intuitiv und für den klinischen Einsatz passend Angesehen.

Im Vergleich des Prototyps mit dem Android-Frontend zeigen sich die Probanden unterschiedlicher Meinung. Während gerade die Möglichkeit der Bedienung des Systems am PC positiv bemerkt wurde, wird das System von manchen Probanden, im Vergleich mit der Tablet-Benutzeroberfläche, als überladen bezeichnet.

7.3.4 Identifizierte Gebrauchstauglichkeitsprobleme und Gegenmaßnahmen

Betrachtet man die Detailauswertung des Usability-Tests nach PROMEDIKs so erkennt man, dass bei ca. 85% aller 83 Teilaufgaben keine Bedienfehler aufgetreten sind. Bei 8 der Teilaufgaben, also bei knapp 10%, trat zumindest ein schwerer Bedienfehler auf, bei 4 weiteren kam es zumindest zu zwei leichten Bedienfehlern. Eine Darstellung dieser Verteilung befindet sich in Abbildung 23.

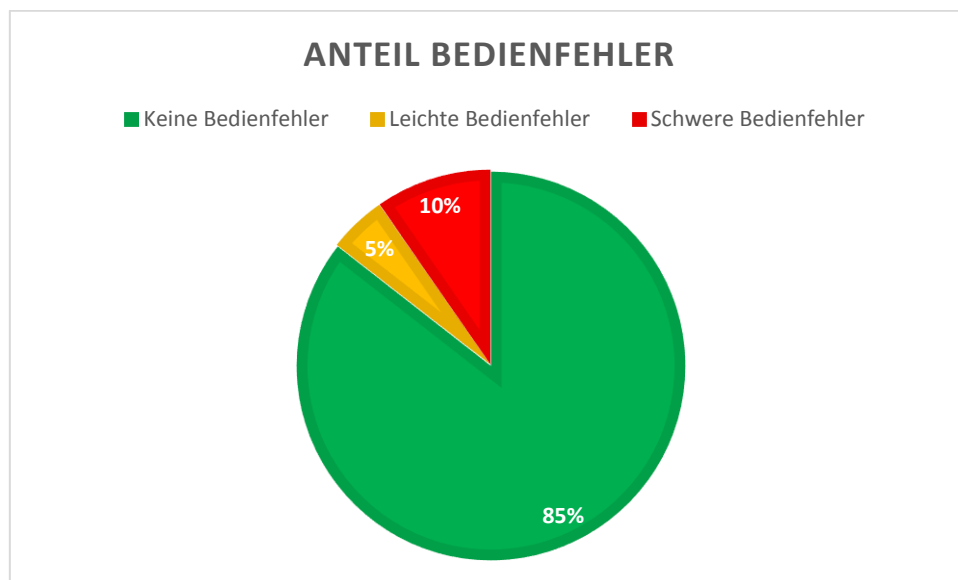


Abbildung 23: Prozentuelle Verteilung der Bedienfehler in der PROMEDIKs Auswertung

Im folgenden Abschnitt wird nun auf die Teilaufgaben mit Bedienfehlern eingegangen. Hier werden die Fehlerhergänge erläutert und mögliche Gegenmaßnahmen genannt. Hierzu wird wieder die Detailauswertung der PROMEDIKs-Analyse betrachtet, da diese eine feinere Aufteilung der Bedienfehler pro Benutzer erlaubt wie die Auswertung nach den Gebrauchstauglichkeitszielen. Die gescheiterten Gebrauchstauglichkeitsziele sind jedoch deckungsgleich mit den mit Bedienfehlern behafteten Teilaufgaben der PROMEDIKs-Analyse. Die Verteilung der Benutzerfehler auf die einzelnen betroffenen Teilaufgaben ist in Abbildung 24 zu finden.

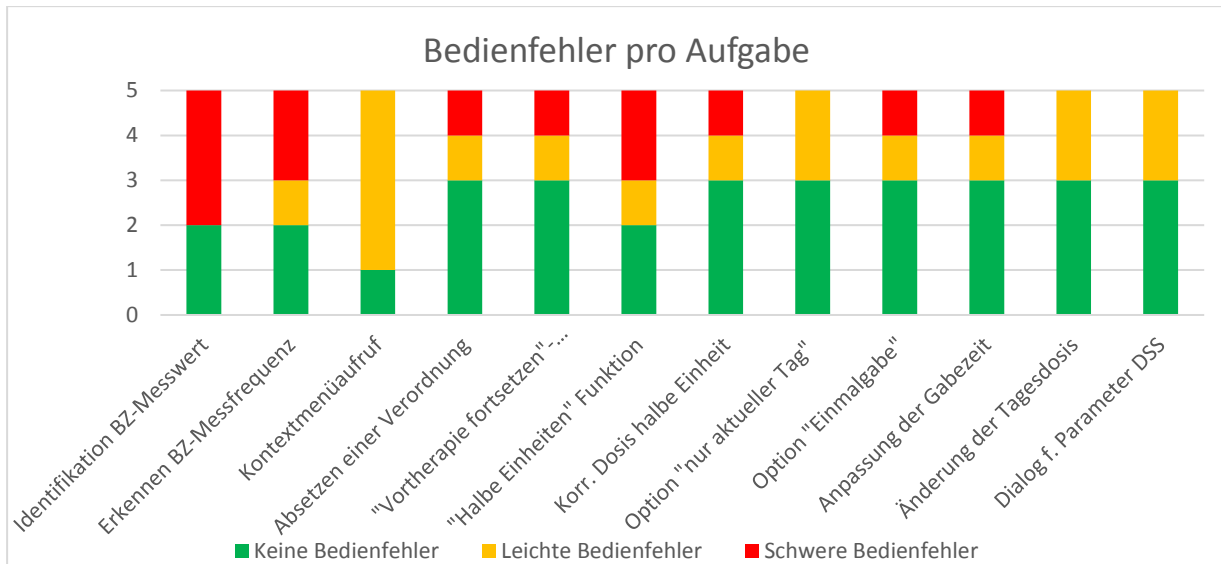


Abbildung 24: Identifizierte Bedienfehler während des Usability-Tests

Probleme der Identifikation des korrekten Blutzuckermesswertes

Drei der Probanden war es nicht möglich dem Testleiter den in der Aufgabe gewünschten BZ-Messwert zu benennen. Das Hauptproblem liegt dabei in der Unterscheidbarkeit der einzelnen Tage, bzw. der Zuordnung der dargestellten Tage zu dem korrekten Datum. In Abbildung 19 und Abbildung 25 ist das BZ-Verlaufdiagramm abgebildet. Hier erkennt man den in grau hervorgehobenen Nachtzeitraum. Eine fehlende Markierung des Datumswechsels sowie das nicht zusätzlich hervorgehobene Datum sind zu erkennen.

Um die Unterscheidung der einzelnen Tage zu begünstigen, wäre eine farbliche Unterscheidung der einzelnen Tage, sowie eine bessere Markierung des Datumswechsels und eine Vergrößerung des Datumstextes in der Zeitachse des Diagramms anzudenken.

Erkennen der eingestellten Blutzuckermessfrequenz

Zwei der Probanden fanden die am linken Bildschirmrand dargestellte BZ-Messfrequenz nicht, bzw. konnten diese nicht als solche identifizieren. Zur Veranschaulichung ist der Parameter in Abbildung 25 hervorgehoben.

Dieses Problem beruht laut dem Feedback der Ärzte darauf, dass im klinischen Alltag maximal vier BZ-Messungen an einem Tag durchgeführt werden. Eine Nachtmessung findet äußerst selten statt. Die Darstellung von fünf Messzeitpunkten entspricht somit nicht dem klinischen Terminus und führt zur Verwirrung der Benutzer. Die Darstellung von nur vier Messzeitpunkten wäre hier als Gegenmaßnahme zu nennen.

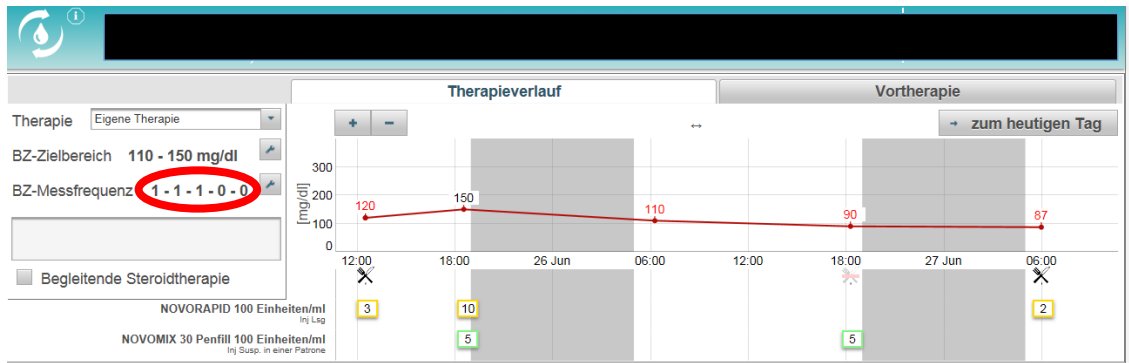


Abbildung 25: Hervorhebung des schwer zu identifizierenden Messfrequenzparameters

Erster Aufruf des Kontextmenüs

Vier der fünf Probanden hatten Probleme beim ersten Aufruf des Kontextmenüs in der virtuellen Fieberkurve. Sie zeigten langes Probierverhalten, so versuchten sie das Menü mit Klicks auf den Medikamentennamen am linken Rand der Zeile, sowie durch Rechtsklicks in den Tabellenbereich der Medikamentenverordnung aufzurufen. In weiterer Folge kam es zum Aufruf des Menüs am falschen Tag. Nach dem Probierverhalten konnten jedoch alle Probanden das Kontextmenü am richtigen Tag aufrufen. Das Problem trat auch bei keiner weiteren Aufgabe, bei welcher das Kontextmenü verwendet werden musste, wieder auf.

Dieses Verhalten der Probanden zeigt eine mögliche Schwäche in der Intuitivität der virtuellen Fieberkurve. Da das Problem jedoch nur beim ersten Aufruf des Menüs auftrat, und die Probanden diese Lösung positiv erwähnten, kann auf eine Änderung des Systems wohl verzichtet werden. Durch eine Schulung der Benutzer kann diesem Problem bei der Erstbenutzung entgegengewirkt werden. Den Probanden wurde vor Beginn des Tests das System nur kurz vorgestellt, sie wurden nicht eingeschult.

Absetzen einer Verordnung

Diese Teilaufgabe wurde direkt im Anschluss auf den ersten Aufruf des Kontextmenüs ausgeführt, bei dem die Probanden mit Problemen zu kämpfen hatten. Diese Verwirrung, bzw. der Aufruf des Menüs am falschen Tag führte dazu dass ein Proband das gewünschte Medikament am falschen Tag absetzte. Einer der Probanden pausierte die Medikation anstatt sie abzusetzen. Dies hat für die Medikamentengabe des Patienten dieselbe Wirkung wie ein Absetzen der Medikation, könnte jedoch zur Verwirrung bei der Therapieplanung im weiteren Therapieverlauf führen.

Bei keiner der weiteren Aufgaben mit dem Kontextmenü wurde eine falsche Option bzw. ein falscher Tag gewählt. Da die Methodik des Absetzens sich nicht von anderen Aufgaben, wie dem Pausieren, unterscheidet ist davon auszugehen, dass es sich um kein Benutzerproblem bei der Absetzen-Funktion handelt. Vielmehr steht dieses Benutzungsproblem im Zusammenhang mit der Erstbenutzung des Kontextmenüs.

Auswahl der „Vortherapie fortsetzen“ Therapieoption

Nach Eingabe der Vortherapie bei einem neu ins System eingeschriebenen Patienten waren die Probanden angehalten die Vortherapie über die Auswahl von „Vortherapie fortsetzen“ als aktuelle Therapie im GlucoTab-System fortzusetzen. Hier war es zwei Probanden nicht klar, dass es sich bei der Fortsetzung der Vortherapie um eine Form der eigenen Therapie handelt. Einer der beiden Probanden wählte die richtige Therapieoption während der andere sich für die falsche entschied. Die drei anderen Probanden hatten keine Probleme bei der Durchführung der Aufgabe.

Als Feedback für diese Funktion wurde von den Probanden bei denen die Probleme auftraten bemerkt, dass bei der Auswahl der Therapieformen zusätzliche Informationen zu diesen hilfreich wäre, da es durch das Wording nicht klar ersichtlich war worum es sich handelt. Eine Darstellung zusätzlicher Informationen über Tooltips oder eine Änderung des Anzeigetextes zur Therapieoption wäre als Gegenmaßnahme für diese Probleme denkbar.

Auswahl der Funktion „halbe Einheiten“ für die Dosiseingabe und Eingabe der korrekten Dosis mit halben Einheiten

Bei teilbaren oralen Antidiabetika bietet das GlucoTab-System die Möglichkeit der Verordnung von halben oder, wenn vom Medikamentenkodex her möglich, gedrittelten Tabletten. Das Standardinkrement bei den Eingabeelementen der Medikamentendosis beträgt jedoch eine Einheit. Als Möglichkeit zur Änderung des Inkrements wird im Prototyp eine Auswahlbox für teilbare Einheiten angeboten. Drei der Probanden hatten Probleme bei der Auswahl bzw. Identifikation dieser Option. Einer der drei Probanden mit Problemen konnte die Option zumindest nach längerer Suche entdecken. Den beiden anderen war es nicht möglich, die Verordnung der halben Einheiten durchzuführen, auch die Eingabe über die Tastatur war nicht möglich.

Für dieses Problem gibt es zwei Gründe: die Auswahlbox ist im Prototyp schlecht positioniert und das Eingabeelement lässt bei standardmäßig eingestelltem Inkrement keine Tastatureingabe von halben Einheiten zu. Als erste Gegenmaßnahme muss die Auswahlbox prominenter platziert werden. Die zweite Maßnahme ist die Verwendung einer selbst erstellten Komponente, welche bei Teilbarkeit des Medikaments und eingestellten Standardinkrement auch eine Tastatureingabe der teilbaren Einheit zulässt.

Auswahl der Option „Änderung nur am aktuellen Tag“

Bei der Änderung einer Verordnung gibt es ein Auswahlmenü, um diese Änderung nur am gewählten Tag gültig zu machen. Zwei der Probanden fanden diese Option im Dialog beim ersten Mal nicht. Der Fehler wurde umgehend bemerkt und die Änderung rückgängig gemacht. Im zweiten Anlauf konnten beide Probanden die Option finden.

Da es hier zu keinen schwerwiegenden Bedienfehlern kam, und die Probanden die ursprünglich falsche Eingabe selbstständig sofort ausbesserten ist keine dringende Maßnahme nötig. In weiterer Folge sollte die Bedienung dieser Funktion jedoch im Feld beobachtet werden. Hierbei handelt es sich um ein „Komfort“-Feature welches einen anderen Lösungsweg der Verordnungsänderung verkürzt.

Auswahl der Option „Einmalgabe“ und Anpassung des Verabreichungszeitpunktes der Einmalgabe

Bei der Verordnung von Einmalgaben traten Probleme bei zwei Probanden auf. Beiden war es nicht möglich die Option der „Einmalgabe“ im Neuverordnungsdialog auf den ersten Blick zu erkennen. Sie verordneten eine Intervallverordnung nur für diesen Tag, mit Gabe in der Tageszeit „Abend“ (gewünschter Zeitpunkt der Einmalgabe: 17:00). Einer der beiden Probanden bemerkte seinen Fehler und stornierte die Verordnung. Bei der erneuten Neuverordnung erkannte der Benutzer die Möglichkeit der Einmalgabe und verordnete diese richtig.

Für den Patienten hätte die einmalige Verordnung der Intervallverordnung mit Verabreichung zur richtigen Tageszeit keine Auswirkung. Beide Probanden äußerten sich dahingehend bei einer erneuten Verordnung einer Einmalgabe die Option wiederzufinden. Da diese Option grundsätzlich sehr prominent im Dialog platziert ist und bei nicht-Verwenden grundsätzlich

keine Gefährdung des Patienten erfolgt, wird keine Gegenmaßnahme vorgeschlagen. Das Benutzerverhalten dieser Funktion sollte jedoch im Feld beobachtet werden.

Änderung der Tagesdosis

Bei der Anpassung der Tagesdosis zeigten zwei Probanden Probierverhalten beim Aufrufen des Anpassungsdialogs. Letzten Endes konnte die Tagesdosisanpassung jedoch von allen durchgeführt werden.

Die Fieberkurve unterscheidet sich bei Insulinen in der Basis/Bolus-Therapie von der Darstellung in der eigenen Therapie. Der Unterschied ist in Abbildung 26 dargestellt. Die zwei Probanden versuchten den Dialog zur Tagesdosisanpassung mehrmals durch Anklicken der verordneten Medikamentendosen aufzurufen anstelle des direkt danebenliegenden Buttons.

Freie Therapie	18.03.2016	19.03.2016	20.03.2016	21.03.2016	22.03.2016	23.03.2016	24.03.2016
ACTRAPID Penfill 100 I.E./ml - Inj.-Lsg i.e.Patrone (Parallelimp.EU)		0 - 2 - 2 - 2	0 - 1 - 2 - 3	0 - % - % - 3	0 - 1 - 2 - 3	0 - 1 - 2 - 3	0 - 1 - 2 - 3
Basal/Bolus Therapie	18.03.2016	19.03.2016	20.03.2016	21.03.2016	22.03.2016	23.03.2016	24.03.2016
APIDRA 100 E/ml - Injektionsloesung in einem Fertipgen SoloStar				-- 8 - 0	13 - 7 - 8 - 0		
LANTUS 100 Einheiten/ml - Injektionsloesung Fertipgen SoloStar				-- 0 - 0	0 - 28 - 0 - 0		
						Tagesdosis anpassen	

Abbildung 26: Unterschiedliche Darstellung der Insuline in der eigenen und der Basis/Bolus-Therapie

Grundsätzlich wurde versucht die Darstellung in beiden Therapieformen möglichst ähnlich zu gestalten. Die Titelleiste mit den eingeblendeten Daten scheint jedoch in der Basis/Bolus-Therapie manche Benutzer zu verwirren. Als Gegenmaßnahme wäre die Datumsdarstellung in der Titelleiste zu überarbeiten und im letzten Feld für die Insuline nur mehr das Datum des aktuellen Tages anzuzeigen. Für parallel verordnete OADs müsste dann jedoch eine zweite Zeitleiste eingeblendet werden.

Aufruf des Dialogs zur Anpassung des Kreatininwertes und Gewichts des Patienten für die Berechnung der initialen Tagesdosis

Bei der Einschreibung eines Patienten in die Basis/Bolus-Therapie muss zur Berechnung der initialen Tagesdosis das Gewicht sowie der zuvor gemessenen Kreatininwert über einen Dialog eingetragen werden. Ein Proband konnte diesen Dialog nur nach einer kurzen Suche mit Probierverhalten starten. Ein anderer gab im Zuge des Versuchs einer schnellen Ausführung die Werte mehrmals falsch ein und musste diese korrigieren, auch dies wurde als Probierverhalten bewertet.

Da die Eingabe der benötigten Werte allen Probanden letztendlich selbstständig gelang, und der Dialog direkt über einen zentral positionierten Button gestartet werden kann, wird keine dringende Gegenmaßnahme empfohlen. Auch die Bedienung dieser Funktion sollte in Zukunft im Feld beobachtet werden. Gegebenenfalls ist eine direkte Integration der Eingabekomponenten für die benötigten Parameter in den Dialog mit der Berechnungsfunktion zu überdenken. Dieses Layout wurde im Prototyp nicht genutzt, da versucht wurde den Dialog für die Berechnung der initialen Tagesdosis möglichst gleich zu gestalten wie den Dialog zur allgemeinen Tagesdosisanpassung.

8 Diskussion

Ein interessanter Aspekt bei der Betrachtung des regulativen Umfelds betraf die Klassifizierung des GlucoTab Systems als Medizinprodukt. Das System kann, bei Fehlfunktionen oder vorsätzlichem Missbrauch, bei strikter Einhaltung der Dosisvorschläge des CDSS zu schwerer Beeinträchtigung und sogar zum Tod des Patienten führen. Dies führt im Rahmen der Implementierung des Systems nach EN 62304 zur Software-Sicherheitsklasse C, welche den größten Umfang der Dokumentation und der durchzuführenden Tests fordert. Dadurch dass das System jedoch keine direkte Wirkung auf den Patienten in irgendeiner Form ausführt, sondern das klinische Personal durch Workflow Management und Dosisvorschläge unterstützt, wurde es als Medizinprodukt der Klasse I eingeteilt. Diese Medizinprodukteklasse stellt den geringsten Zertifizierungsaufwand dar, das System wird vom Hersteller selbst gekennzeichnet. Diese Diskrepanz wird jedoch durch die Einführung neuer Regeln in der in Zukunft eingeführten MDR aufgehoben, das System wird voraussichtlich zu einem Medizinprodukt der Klasse III.

Bei der Implementierung des Prototyps wurde ein Fokus auf die Benutzeroberfläche gelegt. Der Aspekt der Sicherheit des Systems bezogen auf technologische Risiken wurde vernachlässigt. Es wurde kein umfangreiches Risikomanagement für Sicherheitsprobleme der Webapplikation (Hackerangriffe, Fehlverwendung, Session-Handling, der Verwendung von mehreren Browserfenstern usw.) durchgeführt. Bei der produktiven Entwicklung des Webfrontends müssen diese Aspekte beleuchtet und analysiert werden. Softwarepakete, welche in diesem Bereich standardmäßig zur Verbesserung der technischen Sicherheit verwendet werden, wie Spring Security sollen dann eingesetzt werden.

Eine weitere Verletzung des normativen Umfelds bei der Entwicklung des Prototyps war der Verzicht auf automatisierte Softwaretests im Frontend. Es gäbe auch automatisierte Tests zum Validieren der graphischen Benutzeroberfläche, auf diese Möglichkeit wurde aufgrund des großen Aufwandes bei der Testerstellung verzichtet. Vielmehr wurde die Benutzeroberfläche direkt durch das Ausführen der implementierten Funktionen überprüft. Auf die Erstellung der automatisierten Tests wurde verzichtet um die Zeit zur Umsetzung aller Funktionen im Prototyp zu nutzen. Auf die Umsetzung aller für die Therapieplanung benötigten Funktionen wurde großer Wert gelegt da das Ziel der Erstellung des Prototyps die Durchführung des

summativen Usability-Tests war. Sollte es bei der produktiven Entwicklung des Systems zu häufigen Software-releases an die Kunden kommen, so werden auch automatisierte Tests der Benutzeroberfläche durchgeführt werden müssen. Aktuell steht der Aufwand der Erstellung dieser Tests in keinem Verhältnis mit dem Nutzen.

Aus Zeitmangel konnten nicht alle Verbesserungsvorschläge aus dem Feedback der heuristischen Evaluation umgesetzt werden. Für die Dosis-eingabe wurde die Erstellung einer selbst erstellten Komponente, welche bei Teilbarkeit des Medikaments und eingestelltem Standardinkrement auch eine Tastatureingabe der teilbaren Einheit zulässt, vorgeschlagen. Im abschließenden Usability-Test wurden Probleme bei der Eingabe von halben Einheiten bemerkt welche durch eine selbsterstellte Komponente potentiell abgefangen worden wären. In der produktiven Entwicklung des Webfrontends sollten auch die nicht umgesetzten Vorschläge der heuristischen Evaluation betrachtet werden, da hier schon Usability-Schwächen identifiziert worden sein können welche im Usability-Test selbst nicht bemerkt wurden.

Bei der Planung des Usability-Tests wurde versucht, möglichst normgerecht vorzugehen. Durch die mangelnde Erfahrung bei der Durchführung und Planung solcher Tests könnte es zu Schwächen oder Beeinflussung des Tests gekommen sein. Ein Beispiel dafür wäre das Usability-Problem beim Absetzen der Medikamente, welches eigentlich auf Probleme bei der vorangegangenen Teilaufgabe entstanden ist. Bei einer anderen Reihenfolge der Aufgaben bzw. bei der alleinigen erstmaligen Ausführung des Aufrufs des Kontextmenüs wäre es nicht zum Problem beim Absetzen des Medikaments gekommen.

Durch die Durchführung des Usability-Tests mit der geforderten Mindestzahl an Probanden konnten zusätzliche Usabilityprobleme möglicherweise nicht identifiziert werden. Die Anzahl der Probanden konnte durch die eingeschränkte Verfügbarkeit der Mediziner zum Zeitpunkt der Durchführung des Usability-Tests nicht weiter erhöht werden.

Bei der Auswertung des Usability-Tests wurde je nach Analyse-methode ein unterschiedliches Ergebnis für die Gebrauchstauglichkeit erzielt, obwohl die Auswertung auf denselben Rohdaten basiert. Dieser Unterschied in der Bewertung der Gebrauchstauglichkeit liegt darin begründet, dass zum einen im Laufe des Tests gewisse Teilaufgaben zur Durchführung der Hauptaufgaben mehrmals ausgeführt werden. Diese mehrmals ausgeführten Aufgaben haben

bei der PROMEDIKS-Analyse bei jeder Ausführung einen Einfluss auf das Gesamtergebnis, bei der Auswertung nach Gebrauchstauglichkeitszielen jedoch nicht. Da bei der Durchführung der Aufgaben bei den Probanden ein Lernerfolg erkennbar war, sich somit einmal aufgetretene Probleme bei häufig durchzuführenden Teilaufgaben nicht wiederholten, führte dies zu einer höheren quantifizierten Usability bei der PROMEDIKS Methode.

Zum anderen findet bei der Auswertung nach PROMEDIKS-Methode eine Gewichtung der einzelnen Teilaufgaben nach ihrer Häufigkeit im klinischen Betrieb statt und eine Gewichtung des Schweregrad des Probleme, während bei der Auswertung der Gebrauchstauglichkeitsziele keine Gewichtung der einzelnen Aufgaben und Probleme stattfindet.

9 Schlussfolgerung und Ausblick

Bei der Betrachtung des Themas der klinischen Entscheidungsunterstützung konnte ein Einblick in die voranschreitende Entwicklung von Systemen für unterschiedlichste medizinische Indikationen gewonnen werden. Außerhalb des Fokus der Recherche auf Therapieunterstützungssysteme finden gerade bei der Unterstützung der Diagnostik zahlreiche Entwicklungen auf dem Gebiet von CDSS statt. Allgemein sind die Neu- und Weiterentwicklungen durch den technischen Fortschritt im Bereich der Informationstechnologien, wie die immer besser verfügbare Rechenleistung oder den Einsatz von mobilen Systemen direkt am Point-of-Care begünstigt.

Durch die Ergebnisse der Literaturrecherche und in Review-Literatur zur Effizienz von CDSS konnte festgestellt werden dass man keine konsistente Aussage zum Nutzen von CDSS treffen kann. Häufig bleibt die Verbesserung der Behandlung durch die Entscheidungsunterstützung unter dem erhofften Potential des Systems. Die Ergebnisse der einzelnen Studien sind häufig nicht statistisch signifikant und teilweise konnte gar kein Nutzen bei der Verwendung von CDSS festgestellt werden. Der Nutzen des CDSS hängt von der Akzeptanz der Anwender ab. Dazu gehören sowohl eine glaubwürdige und verständliche Begründung für den Vorschlag des CDSS, als auch eine möglichst hohe Gebrauchstauglichkeit des Systems.

Dem regulativen Umfeld, für die Entwicklung von Software als Medizinprodukt, merkt man die komplexe Verzahnung der einzelnen Regularien und Normen an. Durch diese Komplexität sind manche Dokumente (MDD, EN 62366) in Bezug auf Software sehr allgemein gehalten. Dieses Problem scheint durch Einführung neuer Inhalte in die zukünftige MDR und die Erstellung der neuen ISO 82304 in Angriff genommen zu sein. Gerade bei der Entwicklung einer Benutzerschnittstelle für Software wäre ein stärkerer Verweis auf die, in der EN 62366 nur nebenbei verwiesene, EN ISO 9241 wünschenswert.

Der Fokus bei der Entwicklung des Prototyps lag auf der Gebrauchstauglichkeit der Benutzeroberfläche. Die Gebrauchstauglichkeit wurde durch den abschließenden summativen Usability-Test mit 5 Ärzten validiert. Hier konnte bei der Auswertung nach Gebrauchstauglichkeitszielen ein „Befriedigendes bis Gutes“ bzw. bei der Auswertung nach PROMEDIKS ein „Sehr Gutes“ Ergebnis erreicht werden. Aus dem Feedback des Tests konnten Maßnahmen zur weiteren Verbesserung der Gebrauchstauglichkeit entwickelt werden.

Um das Web-Frontend zum Einsatz als Medizinprodukt zu bringen müssen die in der Diskussion in Kapitel 8 genannten nicht behandelten Sicherheitsrisiken analysiert und gegebenenfalls entsprechende Maßnahmen getroffen werden. Weiters muss eine Lösung zur Benutzer- und Rechteverwaltung sowie die eigentliche Integration in die von den Kunden verwendeten KIS implementiert werden.

Von manchen Probanden wurde das aktuelle Web-Frontend für Nicht-Diabetologen als möglicherweise zu komplex und mit zu vielen Dokumentationsfunktionen angesehen. Dieser Hinweis sollte bei der Weiterentwicklung und den Pilotstudien im Feld beachtet werden. In weiterer Folge könnte Medizinern anderer Fachdisziplinen eventuell ein vereinfachter Modus mit weniger Funktionen zur reinen Unterstützung der Basis/Bolus-Therapie bereitgestellt werden.

In manchen Krankenanstalten wird bei der Krankenvsichte ein Notebook mitgeführt und auch das Pflegepersonal arbeitet zum Teil mit PCs, hier wäre eine Erweiterung des Systems um die im Android-Frontend vorhandenen Pflegefunktionen denkbar.

Da das System aktuell alle für die Therapie von DM Typ 2 benötigten Funktionen abdeckt, wäre eine Ausweitung der Funktionalität auf ein komplettes Diabetesmanagementsystem möglich. Für die Dokumentation von DM Typ 1 Patienten wäre unter anderem die Möglichkeit der Erstellung von Korrekturschemata für Insulindosen pro Proteineinheit nötig. Allerdings führen DM Typ 1 Patienten das Diabetesmanagement solange wie möglich selbstständig durch und bei Insulinpumpen und Blutzuckermessgeräte sind teilweise CDSS-Funktionen implementiert.

Anhand dieser zum Teil gegensätzlichen Gedankengänge zur Komplexität des Systems und den während der Entwicklung des Prototyps gemachten Erfahrungen kann als Resümee gezogen werden, dass die Spezifikation der Funktionen der Software ein schwieriger und für ein erfolgreiches Produkt essentieller Prozess ist. Der Vorteil der direkten Einbindung der Endbenutzer in den Entwicklungsprozess durch agile Softwareentwicklung führt zu einer iterativen Verbesserung des Produkts. Schwachstellen in der Software und der Benutzeroberfläche können identifiziert und Informationen zu zusätzlich benötigten Features gewonnen werden. Das positive Feedback der Ärzte und das Ergebnis des Usability-Tests für den Prototyp zeigen, dass bei der Entwicklung des Systems als Medizinprodukt die Vorteile der agilen iterativen Entwicklungsmethoden gut genutzt werden konnten.

10 Literatur

- [1] American Diabetes Association, "Standards of Medical Care in Diabetes - 2016," *Diabetes Care*, vol. 39, no. 1, 2016.
- [2] L. Guariguata, D. R. Whiting, I. Hambleton, J. Beagley, U. Linnenkamp, und J. E. Shaw, "Global estimates of diabetes prevalence for 2013 and projections for 2035," *Diabetes Res. Clin. Pract.*, vol. 103, no. 2, pp. 137–149, 2014.
- [3] P. Beck, B. Bauer, und T. Pieber, *Zivilisationskrankheit Diabetes: Ausprägungen – Lösungsansätze – Herausforderungen. Österreichischer Diabetesbericht 2013*. 2013.
- [4] Deutsche Diabetes Gesellschaft, "Deutscher Gesundheitsbericht Diabetes 2016," 2016.
- [5] National Health Service (NHS), "National Diabetes Inpatient Audit 2015," 2015.
- [6] K. Donsa, P. Beck, B. Höll, J. K. Mader, L. Schaupp, J. Plank, K. Neubauer, C. Baumgartner, und T. R. Pieber, "Impact of errors in paper-based and computerized diabetes management with decision support for hospitalized patients with type 2 diabetes. A post-hoc analysis of a before and after study," *Int. J. Med. Inform.*, vol. 90, pp. 58–67, 2016.
- [7] C. Sinsky, L. Colligan, L. Li, M. Prgomet, S. Reynolds, L. Goeders, J. W. M. Tutty, und G. Blike, "Allocation of Physician Time in Ambulatory Practice: A Time and Motion Study in 4 Specialties," *Ann. Intern. Med.*, 2016.
- [8] A. X. Garg, N. K. Adhikari, H. McDonald, M. P. Rosas-Arellano, P. J. Devereaux, J. Beyene, J. Sam, und R. B. Haynes, "Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: a systematic review," *JAMA*, vol. 293, no. 10, pp. 1223–1238, 2005.
- [9] E. S. Berner, "Clinical decision support systems: state of the art," *Agency Healthc. Res. Qual.*, no. 9, pp. 4–20, 2009.
- [10] A. Wright, D. F. Sittig, J. S. Ash, J. Feblowitz, S. Meltzer, C. McMullen, K. Guappone, J. Carpenter, J. Richardson, L. Simonaitis, R. S. Evans, W. P. Nichol, und B. Middleton, "Development and evaluation of a comprehensive clinical decision support taxonomy: comparison of front-end tools in commercial and internally developed electronic health record systems."
- [11] M. Alther und C. K. Reddy, "Clinical decision support systems," in *Healthcare Data Analytics*, 2015, pp. 625–656.
- [12] D. R. Mould, G. D'Haens, und R. N. Upton, "Clinical Decision Support Tools: The Evolution of a Revolution.," *Clin. Pharmacol. Ther.*, vol. 66, no. 5, pp. 732–40, 2016.
- [13] L. T. Goodnough und N. Shah, "The next chapter in patient blood management real-time clinical decision support," *Am. J. Clin. Pathol.*, vol. 142, no. 6, pp. 741–747, 2014.
- [14] C. a Siegel und G. Y. Melmed, "Predicting Response to Anti-TNF Agents for the Treatment of Crohn's Disease.," *Therap. Adv. Gastroenterol.*, vol. 2, pp. 245–251, 2009.
- [15] J. K. Mader, T. Birngruber, S. Korsatko, S. Deller, G. Köhler, S. Boysen, T. Augustin, S. I.

- Mautner, F. Sinner, und T. R. Pieber, "Enhanced absorption of insulin aspart as the result of a dispersed injection strategy tested in a randomized trial in type 1 diabetic patients," *Diabetes Care*, vol. 36, no. 4, pp. 780–785, 2013.
- [16] K. M. Neubauer, J. K. Mader, B. Höll, F. Aberer, K. Donsa, T. Augustin, L. Schaupp, S. Spat, P. Beck, F. M. Fruhwald, C. Schnedl, A. R. Rosenkranz, D. B. Lumenta, L.-P. Kamolz, J. Plank, und T. R. Pieber, "Standardized Glycemic Management with a Computerized Workflow and Decision Support System for Hospitalized Patients with Type 2 Diabetes on Different Wards," *Diabetes Technol. Ther.*, vol. 17, no. 10, pp. 685–692, 2015.
- [17] L. Leibovici, M. Fishman, H. C. Schonheyder, B. Kristensen, I. Shraga, und S. Andreassen, "A Causal Probabilistic Network for Optimal Treatment of Bacterial Infection," *IEEE Trans. Knowl. Data Eng.*, vol. 12, no. 4, pp. 517–528, 2000.
- [18] S. Y. Ji, R. Smith, T. Huynh, und K. Najarian, "A comparative analysis of multi-level computer-assisted decision making systems for traumatic injuries," *BMC Med Inf. Decis Mak*, vol. 9, p. 2, 2009.
- [19] A. Wright und D. F. Sittig, "A Framework and Model for Evaluating Clinical Decision Support Architectures."
- [20] L. Leibovici, M. Paul, A. D. Nielsen, E. Tacconelli, und S. Andreassen, "The TREAT project: decision support and prediction using causal probabilistic networks," *Int. J. Antimicrob. Agents*, vol. 30, no. SUPPL. 1, pp. 93–102, 2007.
- [21] L. Leibovici, G. Kariv, und M. Paul, "Long-term survival in patients included in a randomized controlled trial of TREAT, a decision support system for antibiotic treatment.," *J. Antimicrob. Chemother.*, vol. 68, no. 11, pp. 2664–6, 2013.
- [22] TREAT Systems, "Clinical Values of Antimicrobial Stewardship," 2015. [Online]. Available: <http://www.treatsystems.com/pages/ClinicalValues.aspx?select=TreatmentQuality>. [Letzter Zugriff: 04-Jul-2016].
- [23] "About - Monarch Medical Technologies." [Online]. Available: <http://monarchmedtech.com/about/> [Letzter Zugriff: 05-Nov-2016].
- [24] C. Dumont und C. Bourguignon, "Effect of a computerized insulin dose calculator on the process of glycemic control," *Am. J. Crit. Care*, vol. 21, no. 2, pp. 106–116, 2012.
- [25] L. Molnar, M. Berhes, L. Papp, N. Nemeth, B. Fulesdi, J. Goto, K. Matsuda, N. Harii, T. Moriguchi, M. Yanagisawa, D. Harada, H. Sugawara, und O. Sakata, "P1 Cerebral autoregulation testing in a porcine model of intravenously administrated E. coli induced fulminant sepsis," *Crit. Care*, vol. 19, pp. 1–201, 2015.
- [26] M. K. Goldstein, B. B. Hoffman, R. W. Coleman, M. A. Musen, S. W. Tu, A. Advani, R. Shankar, und M. O'Connor, "Implementing clinical practice guidelines while taking account of changing evidence: ATHENA DSS, an easily modifiable decision-support system for managing hypertension in primary care.," *Proc. AMIA Symp.*, pp. 300–4, 2000.
- [27] M. Michel, J. Trafton, S. Martins, und D. Wang, "Improving Patient Safety Using ATHENA-Decision Support System Technology: The Opioid Therapy for Chronic Pain

- Experience," *Adv. Patient Saf.*, pp. 1–12, 2008.
- [28] J. A. Trafton, S. B. Martins, M. C. Michel, D. Wang, S. W. Tu, D. J. Clark, J. Elliott, B. Vucic, S. Balt, M. E. Clark, C. D. Sintek, J. Rosenberg, D. Daniels, and M. K. Goldstein, "Designing an automated clinical decision support system to match clinical practice guidelines for opioid therapy for chronic pain," *Implement. Sci.*, vol. 5, no. 1, p. 26, 2010.
- [29] A. M. Midboe, E. T. Lewis, R. C. Cronkite, D. Chambers, M. K. Goldstein, R. D. Kerns, and J. A. Trafton, "Behavioral medicine perspectives on the design of health information technology to improve decision-making, guideline adherence, and care coordination in chronic pain management," 2011.
- [30] K. H. Seal, "Improving Opioid Safety in Veterans Using Collaborative Care und Decision Support - Full Text View - ClinicalTrials.gov," 2015. [Online]. Available: <https://clinicaltrials.gov/ct2/show/NCT02230722>. [Letzter Zugriff: 22-Oct-2016].
- [31] R. Goud, A. Hasman, und N. Peek, "Development of a guideline-based decision support system with explanation facilities for outpatient therapy," *Comput. Methods Programs Biomed.*, vol. 91, no. 2, pp. 145–153, 2008.
- [32] R. Goud, N. Peek, A. M. Strijbis, P. A. De Clercq, und A. Hasman, "A computer-based guideline implementation system for cardiac rehabilitation screening," *Comput. Cardiol.*, vol. 32, no. February, pp. 323–326, 2005.
- [33] R. Goud, N. F. de Keizer, G. ter Riet, J. C. Wyatt, A. Hasman, I. M. Hellemans, und N. Peek, "Effect of guideline based computerised decision support on decision making of multidisciplinary teams: cluster randomised trial in cardiac rehabilitation.," *BMJ*, vol. 338, p. b1440, 2009.
- [34] M. M. Van Engen-Verheul, L. W. P. Peute, N. F. de Keizer, N. Peek, und M. W. M. Jaspers, "Optimizing the user interface of a data entry module for an electronic patient record for cardiac rehabilitation: A mixed method usability approach," *Int. J. Med. Inform.*, vol. 87, pp. 15–26, 2016.
- [35] M. M. Van Engen-Verheul, S. N. Van Der Veer, N. F. De Keizer, W. Tjon Sjoer Sjoer, E. P. A. Van Der Zwan, und N. Peek, "A Web-based system to facilitate local, systematic quality improvement by multidisciplinary care teams: Development and first experiences of CARDSS online," *Stud. Health Technol. Inform.*, vol. 192, no. 1–2, pp. 248–252, 2013.
- [36] M. M. van Engen-Verheul, N. F. de Keizer, S. N. van der Veer, H. M. C. Kemps, W. J. M. Scholte op Reimer, M. W. M. Jaspers, und N. Peek, "Evaluating the effect of a web-based quality improvement system with feedback and outreach visits on guideline concordance in the field of cardiac rehabilitation: rationale and study protocol," *Implement. Sci.*, vol. 9, p. 780, 2014.
- [37] R. Anchala, E. Di Angelantonio, D. Prabhakaran, und O. H. Franco, "Development and validation of a clinical and computerised Decision Support System for Management of Hypertension (DSS-HTN) at a Primary Health Care (PHC) setting," *PLoS One*, vol. 8, no. 11, pp. 1–10, 2013.
- [38] R. Anchala, S. Kaptoge, H. Pant, E. Di Angelantonio, O. H. Franco, und D. Prabhakaran,

- “Evaluation of effectiveness and cost-effectiveness of a clinical decision support system in managing hypertension in resource constrained primary health care settings: results from a cluster randomized trial,” *J. Am. Heart Assoc.*, vol. 4, no. 1, p. e001213, 2015.
- [39] P. Knaup, J. Pilz, J. Kaltschmidt, S. Ludt, J. Szecsenyi, und W. E. Haefeli, “Standardized Documentation of Drug Recommendations in Discharge Letters A Contribution to Quality Management in Cooperative Care,” *Methods Inf Med*, vol. 45, pp. 336–42, 2006.
- [40] J. Kaltschmidt, T. Bertsche, S. Spw, P. Mg, und H. We, “Successful switch to an electronic drug information system at a university hospital,” 2006.
- [41] T. Hecht, A. C. Bundscherer, C. L. Lassen, N. Lindenberg, B. M. Graf, K.-P. Ittner, und C. H. R. Wiese, “The expenditure of computer-related worktime using clinical decision support systems in chronic pain therapy,” *BMC Anesthesiol.*, vol. 15, p. 113, 2015.
- [42] E. Der Kass, “Mehr Sicherheit bei der Arzneimitteltherapie,” pp. 1–2.
- [43] B. V. Halldorsson, A. H. Bjornsson, H. T. Gudmundsson, E. O. Birgisson, B. R. Ludviksson, und B. Gudbjornsson, “A clinical decision support system for the diagnosis, fracture risks and treatment of osteoporosis,” *Comput. Math. Methods Med.*, vol. 2015, 2015.
- [44] B. R. Ludviksson, B. Gudbjornsson, B. V. Halldorsson, und D. Jonasdottir, “Clinical Decision Support System for Diagnosing and Monitorin of a Disease of a Patient,” US 2016/0259899 A1, 2016.
- [45] Expeda, “Expeda - Osteoporosis & Rheumatic Diseases,” 2012. [Online]. Available: <http://www.expeda.is/>. [Letzter Zugriff: 04-Apr-2016].
- [46] B. Höll, S. Spat, J. Plank, L. Schaupp, K. Neubauer, P. Beck, F. Chiarugi, V. Kontogiannis, T. R. Pieber, und A. Holzinger, “Design of a mobile, safety-critical in-patient glucose management system,” *Stud. Health Technol. Inform.*, vol. 169, pp. 950–954, 2011.
- [47] G. M. Treiber, J. K. Mader, K. M. Neubauer, L. Schaupp, T. Augustin, P. Beck, S. Spat, B. H, F. M. Fruhwald, T. R. Pieber, und J. Plank, “Efficacy, usability and sequence of operations of a workflow-integrated algorithm for basal–bolus insulin therapy in hospitalized type 2 diabetes patients,” pp. 1–10, 2013.
- [48] Europäischer Rat, “Richtlinie 90/385/EWG”, 2007.
- [49] EN 62304, “Medizingeräte-Software – Software-Lebenszyklus-Prozesse,” 2006.
- [50] Europäischer Rat, “Richtlinie 93/42/EWG,” 2007.
- [51] European Commission - DG Enterprise and Industry, *MEDICAL DEVICES: Guidance document - Borderline products, drug-delivery products and medical devices incorporating, as an integral part, an ancillary medicinal substance or an ancillary human blood derivative*, no. 2.1/3 rev 3. 2009.
- [52] COUNCIL OF THE EUROPEAN UNION, “Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on medical devices, and amending Directive 2001/83/EC, Regulation (EC) No 178/2002 and Regulation (EC) No 1223/2009,” vol. 2009, no. 178, p. 355, 2016.

-
- [53] EN ISO 13485, "Medizinprodukte — Qualitätsmanagementsysteme — Anforderungen für regulatorische Zwecke," 2012.
- [54] EN 60601-1, *Medizinische elektrische Geräte, Teil 1 : Allgemeine Festlegungen für die Sicherheit einschließlich der wesentlichen Leistungsmerkmale*. 2007.
- [55] EN ISO 14971, "Medizinprodukte — Anwendung des Risikomanagements auf Medizinprodukte," 2009.
- [56] EN 62366, "Medizinprodukte – Anwendung der Gebrauchstauglichkeit auf Medizinprodukte," 2008.
- [57] M. Safran, Christian; Lorenz, Anja; Ebner, "Webtechnologien. Technische Anforderungen an Informationssysteme 2.," 2013.
- [58] Microsoft, "Chapter 21: Designing Web Applications," 2016. [Online]. Available: <https://msdn.microsoft.com/en-us/library/ee658099.aspx>. [Letzter Zugriff: 27-Sep-2016].
- [59] G. E. Krasner und S. T. Pope, "A Cookbook for Using the Model- View-Controller User Interface Paradigm in Smalltalk-80," *Joop J. Object Oriented Program.*, vol. 1, pp. 26–49, 1988.
- [60] A. Leff und J. T. Rayfield, "Web-application development using the Model/View/Controller design pattern," *Proc. - 5th IEEE Int. Enterp. Distrib. Object Comput. Conf.*, vol. 2001–Janua, no. January, pp. 118–127, 2001.
- [61] 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, und D. Thomas, "Manifest für Agile Softwareentwicklung," 2001. [Online]. Available: <http://agilemanifesto.org/iso/de/manifesto.html>. [Letzter Zugriff: 30-Sep-2016].
- [62] J. Sutherland und K. Schwaber, "Scrum Guides," 1995. [Online]. Available: <http://www.scrumguides.org/>. [Letzter Zugriff: 30-Sep-2016].
- [63] C. Trummer, "Der Scrum Prozess," 2010. [Online]. Available: http://www.effiziente-webprogrammierung.info/_media/optimierung/scrum.png. [Letzter Zugriff: 30-Sep-2016].
- [64] E. Burns, "JavaServer Faces Specification Version 2.2," vol. 436, no. March, p. 436, 2013.
- [65] M. M. Michael Kurz, *JavaServer Faces 2.2: Grundlagen und erweiterte Konzepte*, Auflage: 3. dpunkt.verlag GmbH, 2010.
- [66] D. Vanderkam, "Issues · danvk/dygraphs · GitHub," 2016. [Online]. Available: <https://github.com/danvk/dygraphs/issues>. [Letzter Zugriff: 27-Jul-2016].
- [67] The Apache Software Foundation, "Maven - Welcome to Apache Maven," 2016. [Online]. Available: <https://maven.apache.org/>. [Letzter Zugriff: 20-Aug-2016].
- [68] A. Gärtner und M. Voth, "Computer-Betriebssysteme in der Medizintechnik: Erfahrungen und Empfehlungen eines Krankenhauses," *EHEALTHCOM*, 2013.
- [69] "Browser-Versionen » Statistiken » Browser-Statistik.de." [Online]. Available:
-

-
- <https://www.browser-statistik.de/statistiken/versionen/>. [Letzter Zugriff: 3-Nov-2016]
- [70] “Top 10 Desktop Screen Resolutions from Oct 2015 to Oct 2016 | StatCounter Global Stats.” [Online]. Available: <http://gs.statcounter.com/#desktop-resolution-ww-monthly-201510-201610> [Letzter Zugriff: 3-Nov-2016].
- [71] OWASP, “OWASP Top 10 - 2013,” 2013.
- [72] C. Backhaus, *Usability-Engineering in der Medizintechnik*. Springer International Publishing, 2010.
- [73] J. Nielsen, “10 Usability Heuristics for User Interface Design,” 1995. [Online]. Available: <https://www.nngroup.com/articles/ten-usability-heuristics/>. [Letzter Zugriff: 20-Apr-2016].
- [74] J. Nielsen, “Why You Only Need to Test with 5 Users,” *Nielsen Norman Group*, 2000. [Online]. Available: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>. [Letzter Zugriff: 13-Oct-2016].
- [75] L. Faulkner, “Beyond the five-user assumption: Benefits of increased sample sizes in usability testing,” *Behav. Res. Methods, Instruments Comput.*, vol. 35, no. 3, pp. 379–383, 2003.
- [76] FDA, “Applying Human Factors und Usability Engineering to Optimize Medical Device Design,” *FDA Guid. Ind.*, no. 301, pp. 1–37, 2011.

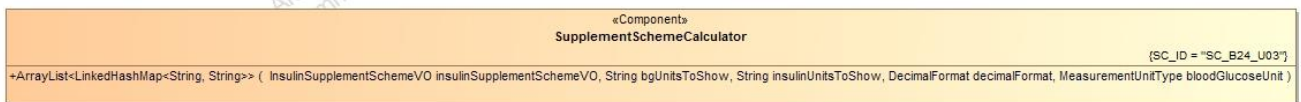


Abbildung 30: UML-Diagram des SupplementSchemeCalculators

11.2 Codeausschnitt – Kombination Dygraphs und JSF

```

var graphs = [];
var data = "#{jSUtils.bGDataForDYGraphs.CSVData}";

var orig_range = [#{jSUtils.getViewRangeString()}];
var graph = new Dygraph(
    document.getElementById("main-graph"),
    data,
    {
        labels: ['Date',
'#{jSUtils.bGDataForDYGraphs.seriesName}', 'border'],
        valueRange: [0, 400],
        ...
    },

    '#{jSUtils.bGDataForDYGraphs.seriesName}': {
        color: "rgba(180, 10, 5, 1.0)",
        ...
    },
    ...

var bgAnnotations = [];
//]]>
<ui:repeat value="#{jSUtils.bGDataForDYGraphs.annotationList}"
var="annotationEntry">
    //<![CDATA[
    bgAnnotations.push({
        series: "#{annotationEntry.seriesName}",
        x: "#{annotationEntry.dateAsString}",
        tickHeight: 5,
        shortText: "#{annotationEntry.textToDisplay}",
        text: "#{annotationEntry.longTextToDisplay}",
        width: #{annotationEntry.labelWidth},
        cssClass: "#{annotationEntry.cssClass}"
    });
//]]>
</ui:repeat>
    //<![CDATA[
    graph.setAnnotations(bgAnnotations);

    graphs.push(graph);
  
```

Der hier angeführte Codeausschnitt zeigt Teile des für die Generierung eines Diagramms benötigten Codes. Rot hervorgehoben sind JSF-Expression Language Aufrufe in welchen Informationen aus dem Model des Web-Frontends in den Code eingefügt werden.

Im blau markiertem Code wird eine `ui:repeat` Komponente als Schleife für die automatische Generierung von Texten (Annotations) im Diagramm verwendet.

11.3 Fragebogen und Aufgaben beim Usability-Test

Interviewte Person	Rolle im GlucoTab System:	
Name:	Datum:	Unterschrift:
Interviewer / Tester:		
Name:	Datum:	Unterschrift:

Allgemeine Angaben zur interviewten Person	
Vorname	
Nachname	
Alter	
Geschlecht	<input type="checkbox"/> männlich <input type="checkbox"/> weiblich
Berufsfeld/Tätigkeit	<input type="checkbox"/> ArztIn <input type="checkbox"/> ArztIn in Ausbildung
Klinischer Fachbereich / Erfahrung mit Diabetesbehandlung	
Dauer der Berufsausübung	
Praktische Erfahrung mit GlucoTab	<input type="checkbox"/> ja <input type="checkbox"/> nein
Dauer Erfahrung mit GlucoTab	

Allgemeine Angaben zu GlucoTab	
Version	Prototyp UI GlucoTab Web-Frontend vom: 01
Verwendetes Notebook	
Testenvironment	Siehe GlucoTab_SVVP_App06_Usability-Test-Plan

Schulungsinhalt	
Inhalt	Durchgeführt
Patientenliste	<input type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht notwendig, da bereits bekannt
Aufgabenliste	<input type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht notwendig, da bereits bekannt
Historie	<input type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht notwendig, da bereits bekannt
Hauptbildschirm	<input type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht notwendig, da bereits bekannt
Einschreibung „Eigene Therapie“	<input type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht notwendig, da bereits bekannt
Auswahl Vortherapie	<input type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht notwendig, da bereits bekannt
Auswahl zusätzliche Medikation	<input type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht notwendig, da bereits bekannt
Einschreibung „Basis/Bolus DSS“	<input type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht notwendig, da bereits bekannt
Verordnung ergänzen	<input type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht notwendig, da bereits bekannt
Verordnung absetzen	<input type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht notwendig, da bereits bekannt
Verordnung pausieren	<input type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht notwendig, da bereits bekannt
Verordnung bearbeiten	<input type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht notwendig, da bereits bekannt
Korrekturschema ändern	<input type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht notwendig, da bereits bekannt
Wechsel zwischen Therapieformen	<input type="checkbox"/> ja <input type="checkbox"/> nein <input type="checkbox"/> nicht notwendig, da bereits bekannt

Bitte beantworten Sie folgende Fragen.

1. Was ist Ihnen am GlucoTab System während der Evaluation **positiv** in Erinnerung geblieben bzw. welche **negativen** Eindrücke konnten sie während der Aufgaben sammeln.

2. Gibt es Dinge die Ihnen Im Web-Frontend Besser oder Schlechter als am Tablet gefallen?

3. Finden sie die Verwendung eines Kontextmenüs im der Verordnungstabelle praktisch?

4. Wie gefällt ihnen die Darstellung und Bedienung des Therapieverlaufs (der Graph)?
Haben sie Verbesserungsvorschläge?

5. Ist der Bildschirm für die Therapieanpassung übersichtlich? Sind alle benötigten
Informationen vorhanden?

6. Wie gefällt Ihnen das Design der Benutzeroberfläche? Z.B. Farbgestaltung, Buttons,
Aufbau der Bildschirme? Haben sie Änderungsvorschläge?

7. Ist die Suchfunktion (Insuline/nicht Insuline) übersichtlich genug um schnell die benötigten Medikamente zu finden? Fehlt etwas / können Ansichten oder Funktionen eingespart werden?

8. Welche Funktionen fehlen Ihnen als Arzt noch im Web-Frontend?

9. Sonstiges

Bitte führen Sie folgende Benutzerszenarien durch.

Initialisierung

- 1) GlucoTab-Web-Frontend R4.0 mit Simulationsdaten starten
- 2) Pausieren der Medikation Novomix bei Patient Stefan Schmidt
- 3) Web-Frontend eingeloggt bereitstellen
- 4) Kurze mündliche Einschulung der Teilnehmer in die Handhabung (siehe oben)

Durchführung:

- 1) Der Interviewer händigt der interviewten Person die Beschreibung der einzelnen Aufgaben auf separaten Kärtchen aus.
- 2) Die interviewte Person führt die beschriebenen Aufgaben aus.
- 3) Der Interviewer dokumentiert anhand des folgenden Bogens das Ergebnis.

Task	MOF ID	UC Beschreibung	Beschreibung der Aufgabe (für den Benutzer)	Vorgehensweise	Ergebnis des Tasks
1	MOF08	Verständnis Medikationsplan	Versuchen Sie herauszufinden welcher Therapieplan (Medikamente und Dosierung) bei „Bernhard Müller“ in den nächsten Tagen verordnet ist.	Benutzer wählt Patient in KIS Mock aus und startet Therapieanpassung Richtige Interpretation des Medikationsplans	<input type="checkbox"/> erfolgreich <input type="checkbox"/> zum Teil erfolgreich <input type="checkbox"/> nicht erfolgreich
			Wann (zu welchen Tageszeiten) sind Blutzuckermessungen beim Patienten verordnet, welche Grenzwerte für die Blutzuckerkonzentration sind verordnet?	Benutzer erkennt das Morgen/Mittag/Abend BZ-Messungen verordnet sind Sowie die Grenzwerte für die Blutzuckerkonzentration	<input type="checkbox"/> erfolgreich <input type="checkbox"/> zum Teil erfolgreich <input type="checkbox"/> nicht erfolgreich
			Welche Medikation war in der Vortherapie verordnet?	Benutzer erkennt die Medikationen der Vortherapie (<i>NOVORAPID 100 Einheiten/ml - Inj-Lsg / NOVOMIX 30 Penfill 100 Einheiten/ml / GLUCOPHAGE - Filmtabletten 500 mg</i>)	<input type="checkbox"/> erfolgreich <input type="checkbox"/> zum Teil erfolgreich <input type="checkbox"/> nicht erfolgreich
			Bitte erklären Sie anhand des BZ-Profiles wie hoch der BZ-Wert am gestrigen Tag am Morgen war.	Patient nennt den Blutzuckerwert von <i>184 mg/dl</i>	<input type="checkbox"/> erfolgreich <input type="checkbox"/> zum Teil erfolgreich <input type="checkbox"/> nicht erfolgreich

Task	MOF ID	UC Beschreibung	Aufgabenstellung (f. Benutzer)	Vorgehensweise	Ergebnis des Tasks
			Verlassen sie den Therapieanpassungsbildschirm mit Abbrechen		
2	MOF 28 MOF 61 MOF 62	Enrolment „Freie Therapie (Übernahme Vortherapie)	Schreiben Sie „Georg Weber“ in „Eigene Therapie“ des Systems ein. Zuhause waren „INSUMAN Rapid 100 I.E./ml - Inj-Lsg“ (10 IE Morgens, und 10 IE Abends), sowie „Metformin Arcana 850 mg“ (1 Tablette in der Früh und 1 am Abend) verordnet. Bitte übernehmen Sie diese Vortherapie als Therapie bei ihrem stationären Aufenthalt im KH.	Benutzer wählt in der Patientenliste „Georg Weber“ aus und startet durch leichten klick das Enrolment. Benutzer gibt geforderte Vortherapie ein (optimal: verwendet er die Suche um die Medikamente zu finden). Benutzer wählt Vortherapie übernehmen und drückt weiter. Benutzer bestätigt die Therapie. Er wählt keine begleitende Steroidtherapie aus und drückt verordnen.	<input type="checkbox"/> erfolgreich <input type="checkbox"/> zum Teil erfolgreich <input type="checkbox"/> nicht erfolgreich
3	MOF 31 MOF 42 MOF33	Enrolment „Freie Therapie“ (Änderung Therapie)	Da „INSUMAN Rapid 100 I.E./ml - Inj-Lsg“ zur Zeit nicht lagernd ist, ändern Sie die Verordnung auf „NOVORAPID 100 Einheiten/ml - Inj-Lsg“ - 8 IE Morgens und 8IE Abends	Benutzer setzt Verordnung „INSUMAN Rapid“ ab. Benutzer wählt Insulin „NOVORAPID 100 Einheiten/ml - Inj-Lsg“ mit angegebener Verordnung und bestätigt die Änderung.	<input type="checkbox"/> erfolgreich <input type="checkbox"/> zum Teil erfolgreich <input type="checkbox"/> nicht erfolgreich
			Ändern Sie bitte zusätzlich die BZ-Messfrequenz auf (Morgen, Mittag, Abend). und den Zielbereich auf 120 mg/dl bis 160 mg/dl Schließen Sie die Einschreibung bzw. den Therapiestart von Georg Weber ab.	Benutzer ändert in den Therapieeinstellungen die Messfrequenz und den Zielbereich	<input type="checkbox"/> erfolgreich <input type="checkbox"/> zum Teil erfolgreich <input type="checkbox"/> nicht erfolgreich
4	MOF 18 MOF 19 MOF 20 MOF 21 MOF 23 MOF 24 MOF 25 MOF 28 MOF 39 MOF 53	Enrolment „Basis/Bolus DSS“	Bernd Koch wurde stationär bei Ihnen aufgenommen. Es ist geplant, Hrn. Koch in das GlucoTab System einzuschreiben. Er soll „Basis/Bolus DSS“ Therapie erhalten. Hr. Koch wiegt 107 kg. Er hatte keine Vortherapie. Schreiben Sie Hr. Koch in das System ein und starten Sie „Basis/Bolus DSS“. Verwenden Sie die Entscheidungsunterstützung um einen initialen Insulin-Tagesbedarf zu berechnen.	Benutzer wählt Patient Bernd Koch im KIS aus und klickt darauf um die Einschreibung zu starten. Benutzer wählt „Basis/Bolus DSS“ als Therapie aus. Benutzer gibt den Kreatinin-Wert und das Gewicht ein und übernimmt die vorgeschlagene Tagesdosis sowie die Aufteilung.	<input type="checkbox"/> erfolgreich <input type="checkbox"/> zum Teil erfolgreich <input type="checkbox"/> nicht erfolgreich

	MOF 55 MOF 56 MOF 62		Der Kreatinin-Wert beträgt 2.3. Starten sie die Basis/Bolus-Therapie.		
			Außerdem ist geplant das Hr. Koch morgens ½ Tablette und mittags 1 Tablette „Amaryl 2mg“ einnimmt. Bitte nehmen Sie Amaryl mit in die Therapie auf. Sie beschließen dass die berechnete Tagesdosis zu hoch gewählt ist. Vermindern Sie daher die Tagesdosis so, dass nur 10 IE Basis-Insulin vorgesehen sind. Ändern Sie zusätzlich das Korrekturschema auf „sensitive Insulinresistenz“. Schließen Sie die Einschreibung bzw. den Therapiestart von Bernd Koch ab.	Benutzer wählt zusätzlich als Medikation Amaryl 2mg → aktiviert ½ Einheiten → wählt Dosierungen richtig → verordnet das Medikament. Benutzer öffnet Basis-Bolus Einstellungsdialog und vermindert Tagesdosis auf 20 oder 21; Benutzer wählt sensitives Korrekturschema aus und bestätigt Eingabe	<input type="checkbox"/> erfolgreich <input type="checkbox"/> zum Teil erfolgreich <input type="checkbox"/> nicht erfolgreich
5	MOF 31 MOF 47 MOF 50 MOF 51 MOF 53 MOF 55 MOF 56	Verordnung ergänzen	Herr Weber soll ab übermorgen zusätzlich zu Metformin auch ein Repaglinid (0,5 mg) zu sich nehmen. Ergänzen Sie die Therapie um diese Medikation (zu 1Stück Früh/Mittag/Abend). Erhöhen Sie aber dafür die Dosierung von NOVORAPID 100 am Abend auf 15 IE nur morgen (heute und übermorgen wieder 8 IE am Abend)	Benutzer wählt Herr Weber in der Patientenliste aus. Benutzer klickt auf Therapieanpassung und wählt „Andere Medikation hinzufügen“. Benutzer sucht Repaglinid in der Medikationsdatenbank und dosiert entsprechend der Angabe. Benutzer wählt die Verabreichung entsprechend der Angabe aus und verordnet das Medikament. Benutzer klickt auf Humalog Verordnung morgen → bearbeiten → erhöht Abendosis auf 15 → wählt Gültigkeitsdauer „nur heute“	<input type="checkbox"/> erfolgreich <input type="checkbox"/> zum Teil erfolgreich <input type="checkbox"/> nicht erfolgreich
			Ab Übermorgen soll das verordnete Metformin vorerst pausiert werden. Geben Sie die Pausierung in das System ein.	Benutzer klickt auf Metformin Verordnung übermorgen → Benutzer aktiviert Pausierung	<input type="checkbox"/> erfolgreich <input type="checkbox"/> zum Teil erfolgreich <input type="checkbox"/> nicht erfolgreich
	MOF 31	Weiters wird beim Patienten eine begleitende Steroidtherapie durchgeführt Schließen Sie die Therapieanpassung ab.	Benutzer aktiviert die Scheckbox für die Steroidtherapie und schließt die Therapieanpassung ab.	<input type="checkbox"/> erfolgreich <input type="checkbox"/> zum Teil erfolgreich <input type="checkbox"/> nicht erfolgreich	

6	MOF 43 MOF 44	Verordnung absetzen Pausierung aufheben	Bei ihrer Visite beschließen Sie, dass bei Patient Stefan Schmidt „NOVORAPID 100 Einheiten/ml - Inj-Lsg“ ab sofort abgesetzt werden soll. Bitte führen Sie diese Aktion am System aus. Dafür soll aber die ab morgen aktive Pausierung der Mixtard 30 Verordnung aufgehoben werden. Führen Sie die Aufhebung der Pausierung im System durch.	Benutzer wählt Hr. Schmidt in der Patientenliste aus. Benutzer klickt auf Medikament (heutiger Tag) und setzt es mit „Verordnung absetzen“ ab. Benutzer klickt auf Mixtard 30 Verordnung morgen Benutzer wählt „Verordnung fortsetzen“ aus oder Bearbeitet Verordnung vor Aufhebung	<input type="checkbox"/> erfolgreich <input type="checkbox"/> zum Teil erfolgreich <input type="checkbox"/> nicht erfolgreich
Task	MOF ID	UC Beschreibung	Aufgabenstellung (f. Benutzer) (für den Benutzer)	Vorgehensweise	Ergebnis des Tasks
7	MOF 45 MOF 46 MOF 48 MOF 49 MOF 54	Verordnung wieder Aufnehmen / Korrekturschema definieren	Alles wieder zurück! Nehmen Sie das eben abgesetzte „NOVORAPID 100 Einheiten/ml - Inj-Lsg“ wieder in den Medikationsplan auf (ab heute). Ändern Sie dabei das Korrekturschema auf das vordefinierte Schema „sensitiver Patient“. Ändern Sie das Korrekturschema dahingehend, dass mit 3 IU gestartet wird und erst bei ≥ 160 mg/dl überhaupt korrigiert wird. Schließen Sie die Therapieanpassung bei Herrn Schmidt ab	Benutzer klickt auf Insuman Rapid Verordnung in Tabelle mit abgesetzten Medikamenten Benutzer wählt Option zum Wiederaufnehmen Benutzer wählt voreingestelltes Korrekturschema „sensitiver Patient“ aus Benutzer drückt bei Korrekturschema auf „Ändern/Details“ und ändert es entsprechend der Angabe. Benutzer bestätigt die Änderung und verordnet die Therapie.	<input type="checkbox"/> erfolgreich <input type="checkbox"/> zum Teil erfolgreich <input type="checkbox"/> nicht erfolgreich
8	MOF 18 MOF 19 MOF 20 MOF 21 MOF 22	Therapiewechsel	Wechseln sie bei Hrn. Schmidt von der „eigenen Therapie“ auf die „Basis/Bolus DSS“ Therapie. Lassen Sie sich die Tagesdosis (Kreatinin = 1.6) vom System ausrechnen und verordnen Sie die Therapie.	Benutzer bleibt bei Patient Schmidt und wechselt zum Hauptbildschirm. Benutzer klickt auf Therapieanpassung. Benutzer klickt auf Therapie und wechselt zu „GlucoTab Basis/Bolus“. Benutzer übernimmt das Gewicht und gibt den Kreatininwert ein. Benutzer bestätigt die berechnete Tagesdosis und verordnet die Therapie.	<input type="checkbox"/> erfolgreich <input type="checkbox"/> zum Teil erfolgreich <input type="checkbox"/> nicht erfolgreich
9	MOF 57 MOF 58 MOF 60	Einmalgabe hinzunehmen	Ergänzen Sie die Therapie von Fr. Fischer um eine Einmalverordnung von Amaryl 2mg (1 Tablette morgen um 17:00).	Benutzer wechselt zu Patientin Fischer. Benutzer wählt Therapieanpassung. Benutzer wählt „Andere Medikation hinzufügen“. Benutzer sucht Medikament entsprechend Angabe und gibt Dosierung ein. Benutzer verordnet Medikation.	<input type="checkbox"/> erfolgreich <input type="checkbox"/> zum Teil erfolgreich <input type="checkbox"/> nicht erfolgreich

Tabelle 9: Matrix der Aufgaben und Durchführungsschritte für den Benutzertest

11.4 Detaillierergebnisse heuristische Evaluation

Die Detaillierergebnisse der heuristischen Evaluation befinden sich im digitalen Anhang im Dokument: *HE_Ergebnis.xlsx*

11.5 Detaillierergebnisse Usability-Test

Tabelle 10: Auswertung der Gebrauchstauglichkeit nach PROMEDIKS

Aufgabe	Teilaufgabe/Hauptbedienfunktion	B1	B2	B3	B4	B5	BS	RT
Verständnis Therapieanpassungsbildschirm								
Identifizierung Medikationsplan	1	Öffnen der Therapieanpassungsansicht für richtigen Patienten	1	1	1	1	1	1
	2	Identifikation der verordneten Medikamente und Dosierungen	1	1	1	1	1	3
Zeitpunkt Blutzuckermessungen Grenzwerte für Blutzuckerspiegel	3	Erkennen der korrekten BZ-Messfrequenz	3	3	2	1	1	2
	4	Erkennen der Grenzwerte für die Blutzuckerkonzentration	1	1	1	1	1	2
Identifizierung Vortherapie	5	Auswahl des Vortherapie-Tabs	1	1	1	1	1	1
	6	Identifikation der Medikation der Vortherapie	1	1	1	1	1	1
Blutzuckerwert des Vortages	7	Auswahl des Therapieverlauf-Tabs	1	1	1	1	1	1
	8	Identifikation des korrekten Datums und Messewertes im Blutzuckerlaufdiagramm	1	1	3	3	3	3
	9	Beenden des Therapieanpassungsbildschirmes	1	1	1	1	1	1
Einschreiben "Freie Therapie"								
Einschreiben eigene Therapie mit Vortherapie	10	Öffnen der Therapieanpassungsansicht für richtigen Patienten	1	1	1	1	1	1
	11	Betätigen von "Andere Medikation hinzufügen"	1	1	1	1	1	2
	12	Wählen des korrekten Medikaments mittels der Medikamentensuche	1	1	1	2	1	3
	13	Korrekte Doseingabe für das gewählte Medikament und Bestätigung der Verordnung	1	1	1	1	1	3
	14	Betätigen von "Insulin hinzufügen"	1	1	1	1	1	2
	15	Wählen des korrekten Medikaments mittels der Medikamentensuche	1	1	1	1	1	3
	16	Korrekte Doseingabe für das gewählte Medikament und Bestätigung der Verordnung	1	1	1	1	1	3
	17	Bestätigen der Vortherapie	1	1	1	1	1	1
	18	Auswahl der "Vortherapie fortsetzen"-Therapieoption	1	1	2	3	1	2

Aufgabe	Teilaufgabe/Hauptbedienfunktion	B1	B2	B3	B4	B5	BS	RT
---------	---------------------------------	----	----	----	----	----	----	----

Anpassung "Freie Therapie"									
Änderung Intervallmedikation	19	Aufrufen des Kontextmenüs in der virtuellen Fieberkurve	2	2	1	2	2	2	3
	20	Absetzen der gewünschten Medikamentenverordnung	1	1	1	2	3	3	3
	21	Betätigen von "Insulin hinzufügen"	1	1	1	1	1	1	2
	22	Wählen des korrekten Medikaments mittels der Medikamentensuche	1	1	1	1	1	1	3
	23	Korrekte Doseingabe für das gewählte Medikament	1	1	1	1	1	1	3
	24	Bestätigen der neuen Medikamentenverordnung	1	1	1	1	1	1	3
Ändern der Blutzuckermessfrequenz	25	Ändern der Blutzuckermessfrequenz über den entsprechenden Dialog	1	1	1	1	1	1	2
	26	Ändern der Blutzuckergrenzwerte über den entsprechenden Dialog	1	1	1	1	1	1	2
	27	Speichern und Verlassen des Therapieplanes durch betätigen des "Verordnen"-Buttons	1	1	1	1	1	1	3
Einschreiben und Anpassen "Basis/Bolus-Therapie"									
Einschreiben in Basis/Bolus-Therapie mit Entscheidungsunterstützung	28	Öffnen der Therapieanpassungsansicht für richtigen Patienten	1	1	1	1	1	1	1
	29	Bestätigen der leeren Vortherapie	1	1	1	1	1	1	2
	30	Auswahl der "Basis/Bolus"-Therapieoption	1	1	1	1	1	1	3
	31	Eingabe des Gewichts und des Kreatininwertes im Dafür vorgesehenen Dialogfenster	2	1	1	2	1	2	3
	32	Berechnen der initialen Tagesdosis durch Betätigen des "Tagesdosis berechnen"-Buttons	1	1	1	1	1	1	2
	33	Bestätigen der "Basis/Bolus"-Therapie	1	1	1	1	1	1	2
Anpassung der Basis/Bolus-Therapie	34	Betätigen von "Andere Medikation hinzufügen"	1	1	1	1	1	1	2
	35	Wählen des korrekten Medikaments mittels der Medikamentensuche	1	1	1	1	1	1	3
	36	Aktivieren der "Halbe Einheiten" Funktion	3	2	3	1	1	3	2
	37	Korrekte Doseingabe für das gewählte Medikament	2	1	3	1	1	3	3
	38	Bestätigen der neuen Medikamentenverordnung	1	1	1	1	1	1	3
	39	Erkennen und bestätigen des Warnhinweises für gefährliche Medikamentenkombinationen	1	1	1	1	1	1	3
	40	Öffnen des "Basis/Bolus"-Anpassungsdialoges	1	1	1	2	1	1	3
	41	Änderung der Tagesdosis	1	2	1	2	1	2	3
	42	Änderung des Korrekturschemas	1	1	1	1	1	1	2
	43	Bestätigen der Änderung der "Basis/Bolus"-Therapieanpassung	1	1	1	1	1	1	3
	44	Speichern und Verlassen des Therapieplanes durch betätigen des "Verordnen"-Buttons	1	1	1	1	1	1	3
Aufgabe		Teilaufgabe/Hauptbedienfunktion	B1	B2	B3	B4	B5	BS	RT

Anpassung des verordneten Therapieplanes									
Änderung der Medikation in der eigenen Therapie	45	Öffnen der Therapieanpassungsansicht für richtigen Patienten	1	1	1	1	1	1	1
	46	Betätigen von "Andere Medikation hinzufügen"	1	1	1	1	1	1	2
	47	Wählen des korrekten Medikaments mittels der Medikamentensuche	1	1	1	1	1	1	3
	48	Korrekte Doseingabe für das gewählte Medikament	1	1	1	1	1	1	3
	49	Bestätigen der neuen Medikamentenverordnung	1	1	1	1	1	1	3
	50	Aufrufen des Kontextmenues in der virtuellen Fieberkurve	1	1	1	1	1	1	3
	51	Aufrufen des Bearbeiten-Dialoges im Kontext-Menü	1	1	1	1	1	1	3
	52	Auswahl der Option nur aktueller Tag	2	2	1	1	1	2	2
	53	Dosianpassung der Verordnung nur am morgigen Tag	1	1	1	1	1	1	1
	54	Bestätigung der geänderten Medikamentenverordnung	1	1	1	1	1	1	3
Pausieren einer Medikation	55	Aufrufen des Kontextmenues in der virtuellen Fieberkurve	1	1	1	1	1	1	3
	56	Pausierung der gewünschten Verordnung zwei Tage in der Zukunft	1	1	2	1	1	1	2
Verordnung Steroidtherapie	57	Auswählen der Option "Begleitende Steroidtherapie"	1	1	1	1	1	1	1
	58	Speichern und Verlassen des Therapieplanes durch betätigen des "Verordnen"-Buttons	1	1	1	1	1	1	3
Verordnung absetzen / Pausierung aufheben									
Absetzen einer Medikamentenverordnung	59	Öffnen der Therapieanpassungsansicht für richtigen Patienten	1	1	1	1	1	1	1
	60	Aufrufen des Kontextm. in der virtu. Fieberkurve	1	1	1	1	1	1	3
	61	Absetzen der gewünschten Medikamentenverordnung	1	1	1	1	1	1	2
Pausieren einer Medikamentenverordnung	62	Aufrufen des Kontextm. in der virtu. Fieberkurve	1	1	1	1	1	1	3
	63	Aufheben der Pausierung der Medikamentenverordnung	1	1	1	1	1	1	1
Wiederaufnahme von abgesetzter Verordnung / Änderung Korrekturschema									
Wiederaufnehmen einer Verordnung	64	Auswahl der wiederaufzunehmenden Verordnung	2	1	1	1	1	1	1
	65	Bestätigen der geänderten Medikamentenverordnung	1	1	1	1	1	1	3
	66	Wiederaufnahme der abgesetzten Verordnung	1	1	1	1	1	1	1
Korrekturschema anpassen	67	Ändern des Korrekturschemas	1	1	1	1	1	1	2
	68	Bestätigen der geänderten Medikamentenverordnung	1	1	1	1	1	1	3
	69	Speichern und Verlassen des Therapieplanes durch betätigen des "Verordnen"-Buttons	1	1	1	1	1	1	3
Aufgabe		Teilaufgabe/Hauptbedienfunktion	B1	B2	B3	B4	B5	BS	RT

Therapiewechsel								
Therapiewechsel	70	Öffnen der Therapieanpassungsansicht für richtigen Patienten	1	1	1	1	1	1
	71	Start des Therapiewechsel-Dialoges	1	1	1	1	1	2
	72	Auswahl der "Basis/Bolus"-Therapieoption	1	1	1	1	1	2
	73	Eingabe des Gewichts und des Kreatininwertes im dafür vorgesehenen Dialogfenster	1	1	1	1	1	2
	74	Berechnen der initialen Tagesdosis durch Betätigen des "Tagesdosis berechnen"-Buttons	1	1	1	1	1	2
	75	Bestätigen der "Basis/Bolus"-Therapie	1	1	1	1	1	2
	76	Speichern und Verlassen des Therapieplanes durch betätigen des "Verordnen"-Buttons	1	1	1	1	1	3
Verordnung einer Einmalgabe								
Hinzufügen einer Einmalgabe	77	Öffnen der Therapieanpassungsansicht für richtigen Patienten	1	1	1	1	1	1
	78	Betätigen von "Andere Medikation hinzufügen"	1	1	1	1	1	2
	79	Wählen des korrekten Medikaments mittels der Medikamentensuche	1	1	1	1	1	3
	80	Ändern der Verordnungsoption "Einmalgabe"	3	1	2	1	1	2
	81	Dosianpassung und Anpassung der Verabreichungszeit	3	1	2	1	1	3
	82	Bestätigen der neuen Medikamentenverordnung	1	1	1	1	1	3
	83	Speichern und Verlassen des Therapieplanes durch betätigen des "Verordnen"-Buttons	1	1	1	1	1	3

B1-B5 ... Benutzerergebnis 1-5; BS ... Bewertungsstufe; RT ... Relevanz der Teilhandlung

Tabelle 11: Detaillierergebnisse bei der Auswertung nach Gebrauchstauglichkeitszielen

Aufgabe	Gebrauchstauglichkeitsziele	Ergebnis	TZ	GZ
Öffnen der Therapieanpassungsansicht	Die Therapieanpassungsansicht wird für den gewünschten Patienten geöffnet	100%	J	J
Beenden der Therapieanpassungsansicht	Alle Probanden beenden die Therapieanpassungsansicht über "Abbrechen"	100%	J	J
	Alle Probanden speichern die Verordnung und beenden die Therapieanpassungsansicht über "Verordnen"	100%	J	
Therapieparameter Identifizieren	Identifizieren der Blutzuckergrenzwerte	100%	J	N
	Identifizieren der zusätzlichen Steroidtherapie	100%	J	
	Identifizieren der Frequenz und Zeitpunkte der Blutzuckerwerte	40%	N	
Aufgabe	Gebrauchstauglichkeitsziele	Ergebnis	TZ	GZ

Medikationsplan interpretieren	Alle Probanden interpretieren den verordneten Medikationsplan in der "Freien Therapie" richtig.	100%	J	J
	Alle Probanden interpretieren den verordneten Medikationsplan in der "Basis/Bolus - Therapie" richtig.	100%	J	
Identifikation Vortherapie	Das Register für die Vortherapie wird geöffnet.	100%	J	J
	Die Medikation der Vortherapie wird erkannt.	100%	J	
Blutzuckerverlaufsdiagramm	Im Blutzuckerverlaufsdiagramm kann problemlos navigiert werden.	100%	J	N
	Der Messwert beim gesuchten Zeitpunkt wird richtig identifiziert.	40%	N	
Einschreiben in eigene Therapie	Die Eingabe der Vortherapie funktioniert ohne Probleme.	100%	J	J
	Die Option für die "Freie Therapie" wird gewählt.	100%	J	
Einschreiben "Basis/Bolus" Therapie	Die Option für die "Basis/Bolus-Therapie" wird gewählt.	100%	J	J
	Die Werte für Gewicht und Kreatininwert werden problemlos eingegeben	100%	J	
	Die Berechnung der initialen Tagesdosis kann problemlos gestartet werden.	100%	J	
Eingabe der Vortherapie	Die gewünschte Medikation in der Vortherapie kann erstellt werden.	100%	J	J
	Die Vortherapie kann entsprechend der Vorgabe bearbeitet werden.	100%	J	
	Einzelne Einträge der Vortherapie können ohne Probleme gelöscht werden.	100%	J	
Auswahl „Vortherapie fortsetzen“ Therapieoption	Alle Probanden wählen die Option für die Fortsetzung der Vortherapie aus.	80%	N	N
Therapiewechsel	Der Therapiewechsel von Basis/Bolus auf die eigene Therapie wird ohne Probleme durchgeführt.	100%	J	J
	Der Therapiewechsel von "Freier-" auf "Basis/Bolus" Therapie wird ohne Probleme durchgeführt.	100%	J	
Erstbedienung Kontextmenu	Die Probanden starten das Kontextmenü durch Klick in das gewünschte Feld der virtuellen Fieberkurve.	20%	N	N
	Die gewünschte Aktion wird im Kontextmenu ausgewählt	100%	J	
	Das Kontextmenü kann gewünschter weise beendet werden.	100%	J	
Aufgabe	Gebrauchstauglichkeitsziele	Ergebnis	TZ	GZ

Medikamentensuche	Das gewünschte Medikament kann aus der Liste der Insuline ausgewählt werden.	100%	J	J
	Das gewünschte Medikament kann aus der Liste der oralen Antidiabetika gewählt werden.	100%	J	
	Bei der Suche nach Substanzengruppen kann wird das Medikament der gewünschten Substanzengruppe gewählt.	100%	J	
Dosierung halbe Einheiten	Die Option für die Eingabe der halben Einheiten wird aktiviert.	40%	N	N
	Nach Aktivierung wird die gewünschte Dosis in Halben Einheiten eingegeben.	60%	N	
Verordnung Intervallmedikation	Die Dosis wird entsprechend der Vorgabe gewählt	100%	J	J
	Das Startdatum wird entsprechend der Vorgabe gewählt.	100%	J	
	Die Länge der Verordnung wird entsprechend der Vorgabe gewählt.	100%	J	
Korrekturschema Anpassen	Das gewünschte Korrekturschema wird aus der Liste für die vordefinierten Schemen gewählt.	100%	J	J
	Die Parameter der vordefinierten Schemata werden entsprechend der Vorgaben verändert.	100%	J	
Verordnung Einzelmedikation	Alle Probanden wählen die Option für die Einzelverordnung.	60%	N	N
	Alle Probanden geben den richtigen Zeitpunkt für die Gabe der Einzelverordnung an.	60%	N	
Bearbeiten Intervallmedikation	Die Dosierung der Intervallverordnung wird entsprechend der Vorgabe geändert.	100%	J	J
	Das Startdatum der geänderten Verordnung wird entsprechend gewählt.	100%	J	
Pausieren Intervallmedikation	Alle Probanden können die Verordnung entsprechend der Verordnung pausieren.	100%	J	J
Pausierung aufheben	Alle Probanden können die Pausierung der Verordnung entsprechend der Verordnung aufheben.	100%	J	J
Absetzen Intervallmedikation	Alle Probanden setzen die Verordnung entsprechend der Vorgabe ab.	60%	N	N
Wiederaufnehmen Intervallmedikation	Das richtige Medikament wird in der Tabelle für Abgesetzte Medikamente gewählt.	100%	J	J
	Das gewählte Medikament wird mithilfe des vorgesehenen Dialogs wiederaufgenommen.	100%	J	
Aufgabe	Gebrauchstauglichkeitsziele	Ergebnis	TZ	GZ

Ändern Therapieparameter	Die Anpassung der Blutzuckermessfrequenz wird problemlos durchgeführt.	100%	J	J
	Die Anpassung der Blutzuckergrenzwerte wird problemlos durchgeführt.	100%	J	
	Die Option für die begleitende Steroidtherapie wird entsprechend der Vorgabe gewählt.	100%	J	
Anpassung der Basis/Bolus-Therapie	Die Tagesdosis wird ohne Probleme angepasst.	60%	N	N
	Das gewünschte Korrekturschemawird gewählt.	100%	J	
	Das gewünschte Bolus Insulin wird aus der Liste gewählt.	100%	J	
Wahl der richtigen Medikamenten-kategorie	Die Auswahl der Kategorie "Insuline" zur Medikamentenverordnung wird richtig gewählt.	100%	J	J
	Die Auswahl der Kategorie "Andere Medikamente" zur Medikamentenverordnung wird richtig gewählt.	100%	J	
Lerneffekt Kontextmenü	Bei Wiederholtem Aufrufen des Kontextmenüs treten keine Probleme auf.	100%	J	J
Lerneffekt Therapiestart Basis/Bolus	Bei Wiederholen des Therapiestarts treten keine Probleme auf.	100%	J	J

TZ ... Teilziel, GZ ... Gesamtziel

11.6 Berechnungsbeispiel quantitative Usability

$$QU = \frac{\sum(TR_i \cdot (4 - BS_i))}{\sum(TR_i \cdot BS_{max})} \cdot 100\% = \frac{736}{813} = 90,5\%$$