

Konzeption und Entwicklung einer intuitiven Modellierungssprache für digitale Therapien mittels Chatbots

Masterarbeit
von

Luisa Christine Andre

Lehrstuhl für Pervasive Computing Systeme/TECO
Institut für Telematik
Fakultät für Informatik

Erstgutachter:
Zweitgutachter:
Betreuer:

Prof. Dr. Michael Beigl
Prof. Dr.-Ing. Rainer Stiefelhagen
Dr. Anja Exler

Projektzeitraum: 01.03.2019 – 31.08.2019

Ich versichere wahrheitsgemäß, die Arbeit selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde.

Karlsruhe, den _____ date

Zusammenfassung

Zusammenfass
(Deutsch)

Abstract

Zusammenfass
(Englisch)

Contents

1	Einleitung	1
1.1	Problemstellung und Zielsetzung	3
1.2	Umfeld	3
1.3	Methodisches Vorgehen	3
1.4	Gliederung	4
2	Grundlagen	5
2.1	TherapyBuilder	5
2.1.1	Forscher-Plattform	6
2.1.2	Therapeuten-Plattform	7
2.1.3	Backend	7
2.1.4	App	7
2.2	Rahmenbedingungen von Psychotherapien	7
2.3	Chatbots	7
2.4	Stand der Technik	9
2.4.1	Grafische Programmiersprachen	9
2.4.1.1	Chatbot-Plattformen	9
2.4.1.2	Allgemeine grafische Programmiersprachen	10
2.4.2	Auszeichnungssprachen	10
2.4.3	Experience Sampling	11
2.4.4	Fazit	12
3	Konzeption	15
3.1	Anforderungsanalyse	15
3.2	Ausarbeitung verschiedener Konzepte	15
4	Entwicklung der Modellierungssprache	17
5	Evaluation	19
5.1	Deskriptive Statistik	19
5.2	Beschreibung der Ergebnisse	19
5.2.1	Konstruktionsprinzip versus Konfigurationsprinzip	19
5.2.1.1	Konstruktionsprinzip	19
5.2.1.2	Konfigurationsprinzip	20
5.2.1.3	Gegenüberstellung	20
5.2.2	Sprünge versus Sichtbarkeitsregeln	20
5.2.2.1	Sprünge	20
5.2.2.2	Sichtbarkeitsregeln	20
5.2.2.3	Gegenüberstellung	20

6	Ergebnisse	31
6.1	Deskriptive Statistik	31
6.2	Beschreibung der Ergebnisse	31
6.2.1	Konstruktionsprinzip versus Konfigurationsprinzip	31
6.2.1.1	Konstruktionsprinzip	31
6.2.1.2	Konfigurationsprinzip	32
6.2.1.3	Gegenüberstellung	32
6.2.2	Sprünge versus Sichtbarkeitsregeln	32
6.2.2.1	Sprünge	32
6.2.2.2	Sichtbarkeitsregeln	32
6.2.2.3	Gegenüberstellung	32
6.3	Kritische Reflexion	32
7	Ausblick	41
	Bibliography	43

1. Einleitung

Sie geben Auskunft über das Wetter (vgl. [17]), nehmen Bestellungen entgegen (vgl. [20]) oder wirken als Coach (vgl. [38]) - Chatbots werden bereits vielseitig im Alltag eingesetzt. Auch die Psychologie profitiert von diesen Entwicklungen. 1966 entwickelte Joseph Weizenbaum mit *ELIZA* den ersten Chatbot. *ELIZA* sollte seinem menschlichen Gesprächspartner das Gefühl geben, dass dieser mit einem Psychiater über eine Chatoberfläche kommuniziert. Entwickelt wurde *ELIZA* allerdings nicht mit der Absicht Psychotherapie zugänglich zu machen, sondern um ein Modell zur maschinellen Verarbeitung von natürlicher Sprache zu implementieren (vgl. [36]). Was mit Joseph Weizenbaums *ELIZA* begann, brachte mit der Entwicklung der Forschung und Technik schließlich einige Chatbots, wie *Wysa* (vgl. [38]), *Woebot* (vgl. [37]) und *Tess* (vgl. [33]), im Bereich der psychischen Gesundheit hervor. Sie stellen heutzutage verschiedene Methoden der kognitiven Verhaltenstherapie bereit, die Nutzern helfen können, deren Introspektion zu verbessern. Dabei wirken sie wie ein Coach der jederzeit erreichbar ist (vgl. [15]).

In den 1960-ern hatten nur wenige Zugang zu Computern. Durch ihre Bauweise benötigten diese nicht nur viel Platz, sie waren zu dieser Zeit auch sehr kostspielig (vgl. [8]). Die Technik hat sich allerdings über die Jahrzehnte hinweg stark verändert. Nicht nur wurden Computer erschwinglich und haben eine deutlich größere Rechenleistung, sie begleiten uns mittlerweile auch in Form eines Tablets oder Laptops als Personal Computer durch den Alltag. Seit Apple ihr erstes Smartphone *iPhone* im Jahr 2007 einführte, eröffneten sich durch diese Geräte noch weitere technische Möglichkeiten. Smartphones entwickelten sich zu kleinen, handlichen Geräten die nahezu in jeder Tasche Platz finden (vgl. [9]). Außerdem beinhalten die Geräte heutzutage verschiedene Sensoren, haben Zugriff auf eine Vielzahl von Anwendungen und können sich mit dem Internet verbinden (vgl. [9][1]). Die Handlichkeit und Vielzahl an mitgebrachten Funktionen führte dazu, dass im Jahre 2018 allein in Deutschland 22,74 Millionen Smartphones verkauft wurden (vgl. [39]). Statistiken der *Bitkom Research* ermittelten, dass im Jahr 2017 78 Prozent der Deutschen ein Smartphone verwendeten (vgl. [30]).

Entwickler nutzen die technischen Vorteile der Smartphones und Personal Computer. So begleitet *Woebot* Menschen mit Depressionen oder inneren Unruhen mit

Techniken aus der kognitiven Verhaltenstherapie als Selbsthilfe durch den Alltag (vgl. [15]). Der Nutzer kann dabei auswählen, ob dieser über eine *Iphone*-App, *Android*-App oder via *Facebook Messenger* mit *Woebot* kommunizieren möchte (vgl. [37]). Letzteres ist auf jedem browserfähigen Gerät nutzbar.

Eine Studie der Stanford School of Medicine untersuchte den Einsatz des Chatbots *Woebot* hinsichtlich seiner Realisierbarkeit, Nutzerakzeptanz und die vorläufige Wirksamkeit des bereitgestellten Selbsthilfeprogramms. Verglichen wurden dabei zwei Gruppen. Eine dieser Gruppen, bestehend aus 31 Probanden, erhielt Zugriff auf *Woebot*. Die zweite Gruppe, bestehend aus 25 Probanden, erhielt Zugriff auf das Ebook *Depression* des *National Institute of Mental Health*. Die Studiendauer wurde auf zwei Wochen festgelegt. Nach Ablauf der Studie zeigte sich, dass die Mehrheit der *Woebot*-Gruppe beinahe täglich den Chatbot nutzte. Auch konnte bei der Nutzung des *Woebots* im Vergleich zur Nutzung des Ebooks eine größere Zufriedenheit festgestellt werden. Außerdem ließ sich bei dieser Gruppe ein signifikanter, positiver Einfluss hinsichtlich ihrer Depressionsbewältigung und dem Umgang mit inneren Unruhen messen (vgl. [15]).

Eine weitere Studie testete den Einfluss eines virtuellen Akteurs auf das Nutzerverhalten innerhalb eines klinischen Interviews. In dieser Studie wurden 145 Probanden in zwei Gruppen eingeteilt. 57 dieser Probanden führten einen Dialog mit einem virtuellen Akteur, der von einem Menschen gesteuert wurde. Die restlichen 88 Probanden unterhielten sich mit einem virtuellen Akteur, der mittels künstlicher Intelligenz kommunizierte. Das jeweilige Setting der Gruppen war allen Probanden bekannt. Gemessen wurde, unter anderem anhand eines Fragebogens, die Furcht vor negativer Bewertung (FNE), das Selbstdarstellungsverhalten (IM), die Nutzbarkeit des Systems (SU) sowie die Selbsttäuschung der Probanden (SD). Die Ergebnisse zeigten auf, dass signifikante Unterschiede zwischen den Gruppen gemessen werden konnten. So wurde festgestellt, dass Probanden, die Dialoge mit der künstlichen Intelligenz führte, einen niedrigeren FNE und IM Wert aufweisen (vgl. [18]).

Diese Ergebnisse zeigen auf, dass Chatbots im Bereich der Psychologie und Psychotherapie nützliche Werkzeuge sein können. Allerdings ist das Entwickeln solcher Chatbots für Psychologen noch immer eine Hürde. Zwar gibt es zahlreiche Baukästen zur Entwicklung von Chatbots die keine tiefgreifenden Programmierkenntnisse benötigen. Diese sind jedoch überwiegend auf den Bereich des Marketings ausgerichtet, weshalb sie in ihrem Funktionsumfang meist eingeschränkt sind. Baukästen die mehr Funktionalität bieten, benötigen lange Einarbeitungszeit und Expertenwissen in Bezug auf ihre Programmierung. Eine einfache und schnelle Umsetzung ist daher oft nicht möglich. Auch die Entwicklung eines eigenen Produktes birgt für Psychologen und Softwareunternehmen Probleme. So scheitert die Umsetzung unter anderem an Kommunikationshürden zwischen Entwicklern und Psychologen. Aber auch die komplexen Anforderungen des Medizinproduktegesetzes (MPG), die medizinische Produkte für die Herstellung oder Einführung in den Europäischen Wirtschaftsraum zu erfüllen haben, stellen eine Hürde dar (vgl. [22]).

Das Unternehmen *movisens GmbH* entwickelt derzeit das Projekt *TherapyBuilder* welches Psychologen und Psychotherapeuten die Möglichkeit bieten soll, Chatbots für Studien sowie zur Therapiebegleitung einzusetzen. Im Rahmen dieser Masterarbeit wird für das Projekt *TherapyBuilder* ein Modellierungsansatz *TMA* (Therapy Modelling Approach) entwickelt. Ziel dieses *TMA* ist es, Psychologen die Autonomie

zu geben, ohne Expertenwissen Chatbots zu erstellen, um diese in Studien und therapiebegleitend einzusetzen.

1.1 Problemstellung und Zielsetzung

Ziel der Arbeit ist die Konzeption einer Therapiemodellierungsansatz (*TMA*). Diese soll es erlauben, auch technisch wenig versierten Psychologen ihre Therapieideen in einer Art und Weise zu formulieren, die von einer Maschine verarbeitet und ausgeführt werden kann. Dadurch entfällt der hohe und fehleranfällige Abstimmungsaufwand zwischen Forschern und Entwicklern.

Durch den Einsatz der *TMA* sollen *MPG* konforme Anwendungen mit einem Chatbot UI entstehen, welche eine für den Patienten vertraute, dialogähnliche Kommunikation ermöglicht. Dies erlaubt es eine stärkere persönliche Bindung zwischen App und dem Patienten herzustellen, was den Therapieerfolg unterstützen soll.

In der Arbeit gilt es vor allem die komplexen Konfigurationsmöglichkeiten der Domäne einer digitalen Therapie funktional abzubilden. Durch eine Befragung der Anwendergruppen soll Domänenwissen erarbeitet werden und im Folgenden die *TMA* und dessen grafische Repräsentation iterativ entworfen werden. Dabei gilt ein hohes Augenmerk der Usability, um sicherzustellen, dass der Aufwand zur Therapieentwicklung und Studiendurchführung nicht größer ist, als derzeitige Methoden zur Therapieentwicklung und Studiendurchführung.

Es werden zwei Modellierungsansätze entwickelt. Einer dieser Ansätze wird in Form eines komplexen Mockups umgesetzt. Ein weiterer in Form eines Prototypen, welcher aus der Anpassung eines Experience Sampling Tools resultiert. Anschließend erfolgt eine Evaluation der Entwürfe in einer Vergleichsstudie.

1.2 Umfeld

Die Arbeit findet im Rahmen des Forschungsprojektes *TherapyBuilder*, der Firma *movisens GmbH*, statt. Ziel des Vorhabens ist die Entwicklung eines Softwaretools, mit dessen Hilfe Anwender (medizinisch therapeutische Experten) prototypische aber studientaugliche digitale Therapiesysteme innerhalb weniger Tage mit minimalem finanziellem Einsatz und ohne Programmierkenntnisse erstellen können. Dies soll eine Evaluation der Wirksamkeit von Methoden und Therapieansätzen in kürzester Zeit ermöglichen.

1.3 Methodisches Vorgehen

Zur Konzeptionierung der Modellierungsansätze müssen zunächst mehrere Punkte betrachtet werden. Da sich die Zielgruppe aus medizinisch therapeutischen Experten zusammensetzt, gilt es herauszufinden, welche technischen Rahmenbedingungen für diese gegeben ist. Hierfür muss zunächst die Gruppe der medizinisch therapeutischen Experten genauer definiert werden. Anschließend werden die Anforderungen für studientaugliche digitale Therapiesysteme ermittelt. Hierfür werden verschiedene Studien betrachtet, die bereits in Form eines Chatbots umgesetzt werden können. Auf dieser Basis werden verschiedene Konzepte ausgearbeitet. Unter anderem werden für die Konzeptionierung Stilmittel betrachtet, die in verschiedenen

Technologien eingesetzt werden. Diese Technologien werden in Kapitel 2.4 *Stand der Technik* erörtert. Die entwickelten Konzepte werden prototypisch umgesetzt und abschließend in einer explorativen, qualitativen Studie evaluiert. Die Studie wird dabei so aufgebaut, dass die Probanden verschiedene Konzepte austesten und später Konzepte, die gleiche Informationen umsetzen, miteinander vergleichen und bewerten. Basierend auf den Ergebnissen wird eine Einschätzung abgegeben, welche Konzepte zur Umsetzung der späteren *TherapyBuilder*-Plattform weiter entwickelt werden und welche weiteren Anpassungen sinnvoll sein könnten. Der Ablauf wird in folgender Grafik nochmals verdeutlicht.

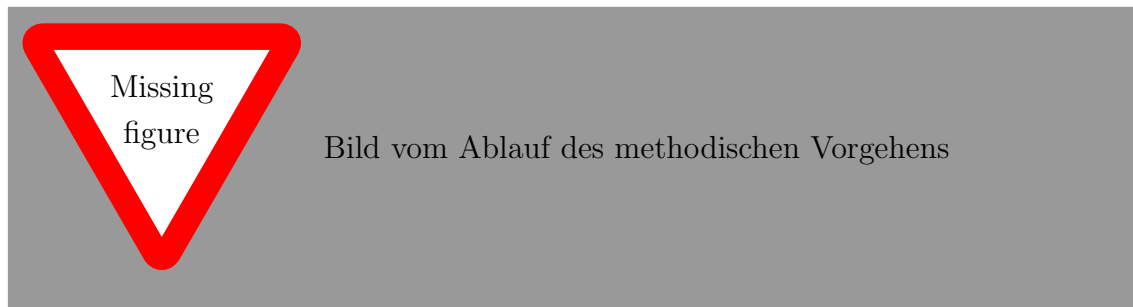


Figure 1.1: Ablauf des methodischen Vorgehens.

1.4 Gliederung

Im folgenden wird die Gliederung der Arbeit vorgestellt. Zunächst werden in 2 grundlegende Definitionen und Rahmenbedingungen erläutert. Dies beinhaltet die Einführung von Begriffen und ihren Definitionen, die im Laufe dieser Arbeit verwendet werden. Aber auch die Rahmenbedingungen von Psychotherapien. Die Rahmenbedingungen setzen sich aus grundlegenden technischen Anforderungen sowie häufig verwendeten Stilmitteln zusammen, die in Psychotherapien eingesetzt werden.

Kapitel 3 betrachtet den Aktuellen Stand der Technik. Hier werden verschiedene Konzepte bewertet, die für eine Umsetzung eines Therapiemodellierungsansatz zur Modellierung von Therapien mit Chatbots in Frage kommen. Dies beinhaltet die Betrachtung von grafischen Programmiersprachen, Auszeichnungssprachen sowie verschiedene Experience Sampling Tools.

Anschließend beginnt in Kapitel 3 die Konzeption verschiedener Therapiemodellierungsansätze. Zur Ausarbeitung der Therapiemodellierungsansätze wird zunächst eine Anforderungsanalyse aufgestellt. Diese betrachtet verschiedene bereits vorhandene Studien, die ein Dialog-Ähnliches Format verwenden. Anschließend werden die daraus resultierenden Konzepte beschrieben und evaluiert.

Kapitel 4 befasst sich mit der Entwicklung verschiedener Konzepte, die später im Modellierungsansatz eingesetzt werden könnten. Beschrieben werden die Konzepte an sich sowie deren Umsetzung. Anschließend wird das Studiendesign beschrieben und die Studienergebnisse vorgestellt und evaluiert.

Abschließend werden in Kapitel ?? die Ergebnisse dieser Arbeit zusammengefasst und bewertet. Diese Ergebnisse bieten die Grundlage für den Ausblick auf die weitere Umsetzung des Therapiemodellierungsansatz.

2. Grundlagen

Zunächst wird das Projekt *TherapyBuilder* der Firma *movisens GmbH* näher erläutert. Dies dient zum besseren Verständnis der Forschungsfrage und porträtiert die Rahmenbedingungen dieser Arbeit. Anschließend wird die Expertengruppe näher definiert. Des Weiteren werden grundlegende Rahmenbedingungen ermittelt, die für Psychotherapien gelten. Abschließend wird die Definition eines Chatbots erläutert und erklärt, was unter einem Chatbot im Rahmen dieser Arbeit verstanden wird.

2.1 TherapyBuilder

Das Projekt *TherapyBuilder* entstand im Rahmen verschiedener Überlegungen der Firma *movisens GmbH*. Die Firma *movisens GmbH* bietet Produkte und Dienstleistungen für ambulantes Assessment in der Forschung an. Unter ambulantem Assessment wird hierbei das Erfassen von Daten einer untersuchten Person im Alltag verstanden. Ambulant bedeutet in diesem Zusammenhang, dass sich die Personen in ihrem natürlichen Umfeld im Alltag befinden. Für die Datenerfassung ist kein vorbestimmter Ort notwendig, an dem sich der Proband stationär einfinden muss. Diese Erfassung kann über verschiedene Methoden geschehen. Eine Möglichkeit ist das Tracken von Patientenverhalten über verschiedene Sensoren. Diese können beispielsweise am Patienten selbst angebracht sein. Im Laufe der Datenerhebung zeichnen die Sensoren die benötigten Daten auf. Eine weitere Methode ist die *Experience Sampling Method (EMS)*. Bei dieser Methode führt die zu untersuchende Person ein Tagebuch zur gezielten Selbstbeobachtung. Diese Methode kann heute leicht mit Hilfe von Smartphones umgesetzt werden.[2]

Das Produkt *movisensXS* der Firma *movisens GmbH* bietet bereits verschiedene Funktionen zur Umsetzung von Datenerhebung via *EMS*. Mit diesem Tool können Forscher Fragebögen erstellen und sie auf den Smartphones ihrer Probanden ausführen lassen. Einige der bereits erstellten Fragebögen verwenden Dialoge in Form von Anleitungen, Fragen von Seiten des Forschers und den darauf folgenden Antworten des Probanden. Diese Fragebögen könnten von den Vorteilen eines Chatbots profitieren. So könnten die Nutzer eine höhere Zufriedenheit bei der Nutzung empfinden und eine größere Bereitschaft zeigen, die Fragebögen auszufüllen und Anleitungen durchzuführen (vgl. [15]).

Da die derzeitige *EMS*-Plattform *movisensXS* noch keine Chatbot-ähnliche Ausgabe unterstützt, wurde das Projekt *TherapyBuilder* gestartet. Ziel des *TherapyBuilder*-Projekts ist die Entwicklung eines Softwaretools, mit dessen Hilfe Anwender (medizinisch-therapeutische Experten) prototypische aber studientaugliche digitale Therapiesysteme erstellen können. Dies soll innerhalb weniger Tage mit minimalem finanziellen Einsatz und ohne Programmierkenntnisse realisierbar sein, um so die Wirksamkeit von Methoden und Therapieansätze in kürzester Zeit evaluieren zu können. Dabei soll das Projekt *TherapyBuilder* derart gestaltet sein, dass der Anwender sich nicht um eine Medizinproduktegesetz (*MPG*)-konforme Softwareentwicklung kümmern muss.

Das Projekt setzt sich aus den vier Komponenten *Forscher-Plattform*, *Therapeuten-Plattform*, *Backend* und *App* zusammen. Die Komponenten werden nachfolgend näher beschrieben.



Figure 2.1: Architektur des *TherapyBuilders*

2.1.1 Forscher-Plattform

Auf dieser Plattform wird die Umsetzung eines studientauglichen digitalen Therapiesystems realisiert. Unter einem studientauglichen digitalen Therapiesystem wird ein System verstanden, welches zur Umsetzung von Therapien eingesetzt werden kann. Die Therapien werden in diesem Fall digital, beispielsweise über ein Smartphone, Tablet oder Computer, bereitgestellt. Die Therapie kann dabei in Form einer Studie evaluiert werden.

Die Forscher-Plattform erlaubt Forschern eine digitale Therapie anzulegen. Diese kann später über die Therapeuten-Plattform der entsprechenden Zielgruppe zugänglich gemacht werden. Eine Therapie setzt sich dabei aus verschiedenen Therapiemodulen zusammen. Ein Therapiemodul ist gleichbedeutend mit der Umsetzung einer Therapiemethode. Der Aufbau des Therapiemoduls lässt sich wie folgt versinnbildlichen.

Das Therapiemodul folgt einer Art Skript (*engl. Script*). In diesem Skript wird festgelegt wann und auf welche Weise ein Dialog zwischen Anwender und Chatbot gestartet wird. Innerhalb des Skripts gibt es verschiedene Handlungsstränge (*engl. Storylines*). Ein Handlungsstrang beschreibt einen Dialog zwischen Nutzer und Chatbot und dessen Verlauf. Der Handlungsstrang kann neben einem normalen

Dialog auch aus verschiedenen Aktionen (*engl. Actions*) bestehen, die dem Nutzer bereitgestellt werden um Tagebuch über verschiedene Verhaltensweisen zu führen. Auch kann ein Handlungsstrang Übungen beinhalten, die der Nutzer durchführt. Diese Übungen können dem Nutzer als Werkzeug in einer Art Werkzeugkasten (*engl. Toolbox*) bereitgestellt werden. Somit kann der Nutzer jederzeit auf diese Übungen zugreifen.

2.1.2 Therapeuten-Plattform

2.1.3 Backend

2.1.4 App

2.2 Rahmenbedingungen von Psychotherapien

- Intervention
- Just in Time
- Psychologischer Psychotherapeut
- Experience Sampling

2.3 Chatbots

- Gabler: Chatbots oder Chatterbots sind Dialogsysteme mit natürlichsprachlichen Fähigkeiten textueller oder auditiver Art. Sie werden, oft in Kombination mit statischen oder animierten Avataren, auf Websites oder in Instant-Messaging-Systemen verwendet, wo sie die Produkte und Dienstleistungen ihrer Betreiber erklären und bewerben respektive sich um Anliegen der Interessenten und Kunden kümmern.
 1. Begriff: Chatbots oder Chatterbots sind Dialogsysteme mit natürlichsprachlichen Fähigkeiten textueller oder auditiver Art. Sie werden, oft in Kombination mit statischen oder animierten Avataren, auf Websites oder in Instant-Messaging-Systemen verwendet, wo sie die Produkte und Services ihrer Betreiber erklären und bewerben respektive sich um Anliegen der Interessenten und Kunden kümmern – oder einfach dem Amüsement dienen. In sozialen Medien treten Social Bots auf, die wiederum als Chatbots fungieren können.
 2. Ziele und Merkmale: Ein Chatbot untersucht die Eingaben der Benutzer und gibt Antworten und (Rück-)Fragen aus, unter Anwendung von Routinen und Regeln. In Verbindung mit Suchmaschinen, Thesauri und Ontologien sowie mithilfe der Künstlichen Intelligenz (KI) wird er zu einem breit abgestützten und einsetzbaren System. Ebenfalls unter seinen Begriff fallen Programme, die im Chat neue Gäste begrüßen, die Unterhaltung in Gang bringen sowie für die Einhaltung der Chatiquette (einer speziellen Netiquette) sorgen und beispielsweise unerwünschte Benutzer kicken.
 3. Kritik und Ausblick: Chatbots waren um die Jahrtausendwende ein Hype und wurden 15 Jahre später wieder zu einem, allerdings unter neuen Voraussetzungen, wenn man an die Entwicklungen in der KI und die Überlegungen in der

Ethik denkt. In der Maschinenethik werden Chatbots entwickelt, die moralisch adäquat agieren und reagieren, etwa Probleme des Gesprächspartners erkennen, eine Notfallnummer herausgeben oder ausdrücklich die Wahrheit sagen. Sie kann ebenso Lügenmaschinen als Artefakte hervorbringen, die sie dann untersucht, um wiederum Erkenntnisse in Bezug auf verlässliche und vertrauenswürdige Maschinen zu gewinnen. Die Informationsethik diskutiert die Auswirkungen des Einsatzes von Chatbots, u.a. mit Blick auf die persönliche und informationelle Autonomie. Die Wirtschaftsethik ist relevant hinsichtlich der Unterstützung und Ersetzung von Arbeitskräften

- Verweis auf Quelle von Jürgen

2.4 Stand der Technik

Für die Bearbeitung der Forschungsfrage werden verschiedene Technologien bewertet. Derzeit existiert noch keine bekannte Sprache, die speziell zur Modellierung von Therapien mit Chatbots entwickelt wurden. Aus diesem Grund werden nachfolgend Ansätze betrachtet, die für eine Umsetzung eines *TMA* in Frage kommen. Zunächst werden Plattformen beleuchtet, die eine Erstellung von Chatbots ermöglichen, ohne weitere Programmierkenntnisse zu benötigen.

2.4.1 Grafische Programmiersprachen

Diese Art der Programmiersprache bedient sich visueller Elemente um Programmstrukturen verständlich abzubilden. Die visuellen Elemente können auf eine bestimmte Domäne zugeschnitten sein (vgl. [35]) oder beschränken sich auf die Visualisierung gängiger Programmanweisungen (vgl. [5]). In den folgenden Abschnitten werden Chatbot-Plattformen und allgemeine grafische Programmiersprachen betrachtet.

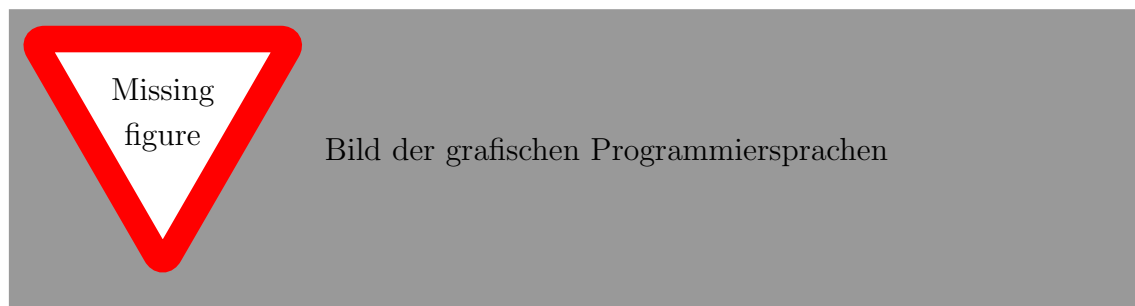


Figure 2.2: Unterschrift

2.4.1.1 Chatbot-Plattformen

Der Konversationsfluss der Chatbots wird auf den Plattformen, wie *Dialogflow* und *IBM Watson* unter anderem als eine Art Baum, ähnlich zur bekannten Ordnerstruktur unter Windows Betriebssystemen, angelegt und dargestellt (vgl. [14] [21]). Chatbot-Plattformen, wie *ManyChat*, *Converse.ai* und *Chatfuel* verwenden Diagramme zur Darstellung eines Chatverlaufes (vgl. [12] [24]) oder Blocksysteme mit Referenzen auf nachfolgende Blöcke (vgl. [10]). Andere Plattformen nutzen keine Darstellung des Verlaufes (vgl. [7]). Im Beispiel von *Botsify* oder *Recast.ai* werden nur Verhaltensweisen angelegt, die durch bestimmte Nutzereingaben getriggert werden.

Auch in der Handhabung der Nutzereingaben gibt es verschiedene Ansätze. So bieten manche Plattformen die Möglichkeit Antworten für den Nutzer des Chatbots vorzugeben (vgl. [10] [24]). Andere hingegen verwenden natürliche Sprachverarbeitung um Schlagwörter einzutrainieren. Der Ersteller des Chatbots legt fest, wie der Chatbot auf die entsprechenden Schlagwörter reagiert (vgl. [7]. [14] [21]). Die Chatbot-Plattform *Chatfuel* verwendet beide Ansätze. So können hier Antworten vordefiniert oder Schlagwörter festgelegt werden (vgl. [10]).

Damit Nutzerdaten abgespeichert und verarbeitet werden können, bieten einige Plattformen Variablen an. Dort können unter anderem Nutzernamen sowie Aufenthaltsort des Nutzers gespeichert und weiterverwendet werden. Der Nutzer kann auf bereits definierte Variablen zurückgreifen oder eigene anlegen (vgl. [10] [12] [14] [21] [24]).



Figure 2.3: Unterschrift

2.4.1.2 Allgemeine grafische Programmiersprachen

Neben dem Einsatz von grafischer Programmierung von Chatbots, gibt es noch weitere Domänen die ebenfalls die grafische Programmierung verwenden. Die grafische Programmiersprache *Labview* beispielsweise, konzentriert sich auf die Domäne System-, Steuer- und Regelungstechnik (vgl. [35]). Programmiert wird, indem Elemente miteinander kombiniert werden, die als Schaltzeichen aus der Elektrotechnik bekannt sind. Nach diesem Prinzip arbeiten auch die Editoren *Matlab Simulink* und *Choreograph* (vgl. [11] [29]).

Ist keine domänenspezifische grafische Programmiersprache gewünscht oder bekannt, ist es dennoch möglich ohne tiefgreifende Programmierkenntnisse Programme zu entwickeln. Ermöglicht wird dies durch Programmiersprachen, die Programmanweisungen durch Diagramme oder Blöcke visualisieren. Für Diagramme werden unter anderem Zustandsdiagramme oder eine Form von Flussdiagrammen verwendet (vgl. [32] [13] [27]). Durch diese Vorgehensweisen lassen sich Schleifen oder bedingte Anweisungen leicht erkennen. Eine Visualisierung mit Blöcken hingegen folgt dem Steckprinzip. So können Anweisungen in Blockform nebeneinander wie untereinander angeordnet werden. Schleifen oder Bedingungen werden durch Blöcke dargestellt, die andere Blöcke beinhalten. Diese Blöcke stellen Anweisungen dar, die innerhalb dieser Schleife oder jeweiligen Bedingung ausgeführt werden (vgl. [6] [25] [31] [41]). *Lego Mindstorms NXT* verbindet das Steckprinzip der Blöcke mit domänenspezifischen Elementen der Lego Mindstorms Bausätze. Insbesondere Schleifen und Bedingungen werden als eine Art Blocksystem genutzt (vgl. [25]).

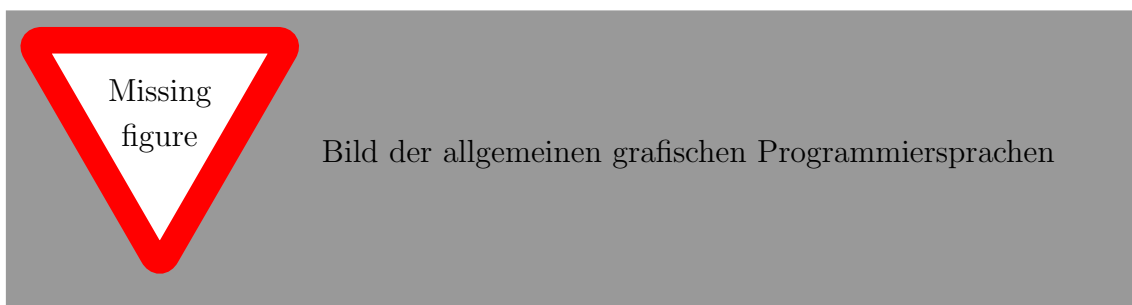


Figure 2.4: Unterschrift

2.4.2 Auszeichnungssprachen

Eine weitere Möglichkeit zur komplexen Programmierung sind die sogenannten vereinfachten Auszeichnungssprachen. Diese arbeiten mit Text der anhand einfacher

Befehle formatiert und strukturiert wird. So kann anhand eines vorangehenden Symbols Text als Überschrift definiert werden. Insbesondere *Markdown* verwendet Sonderzeichen um Textabschnitte zu formatieren und strukturieren (vgl. [16]).

Auch *YAML* nutzt Sonderzeichen, um Listen und größere Mengen von Daten zu beschreiben (vgl. [34]). *BBCode* hingegen verwendet einfache Anweisungen die mit eckigen Klammern eingeleitet und abgeschlossen werden. Die Anweisung selbst wird in Form eines Buchstabens angegeben (vgl. [3]).

HTML ist die geläufigste Auszeichnungssprache. Diese wird zur Strukturierung von Websites benötigt. Dabei können verschiedene Bereiche definiert und deren Inhalte strukturiert werden. *HTML* hat die Fähigkeit durch die Verwendung von Tags komplexe Inhalte, wie Texte, Bilder, Listen und Tabellen zu strukturieren und zu formatieren. Die Tags werden mit spitzen Klammern gekennzeichnet. Im Rahmen einer Studie wurde *HTML* eingesetzt um Ambulante Assessment Protokolle zu erstellen, die sowohl vom Menschen lesbar als auch vom Computer ausführbar sind (vgl. [4]).



Figure 2.5: Unterschrift

2.4.3 Experience Sampling

Psychotherapeuten und Psychologen haben die Möglichkeit anhand bestimmter Experience Sampling Software Fragebögen für mobile Geräte zu entwickeln (vgl. [26]). Hierbei werden auch Lösungen angeboten, die Auszeichnungssprachen verwenden. Die Software *Experience Sampler* verwendet die Auszeichnungssprache *JSON*, aufbauend auf *YAML*, um Fragen, Anzeige- sowie Eingabeformate zu definieren (vgl. [26]). *MyExperience* verwendet einen ähnlichen Ansatz. Als Auszeichnungssprache zur Entwicklung der Fragebögen wird hier auf *XML* zurückgegriffen (vgl. [42]).

Ein weiteres Projekt zur Erstellung von Experience Sampling ist *Jeeves*. Fragebögen werden mit diesem Programm über eine grafische Programmiersprache definiert. Verwendet wird hauptsächlich die grafische Programmierung mit Blöcken. Über eine weitere Oberfläche werden die Eingabeformate der Antworten festgelegt. So ist es möglich Formate wie Likert Skala, Checkboxes, Radiobuttons, Ortsabfragen und weitere zu verwenden (vgl. [28]).

Die Experience Sampling Software *movisensXS* nutzt Diagramme zur Beschreibung des Ablaufs eines Fragebogens. Diese werden nach einem Puzzle-Prinzip angeordnet. Die Fragen selbst, sowie Formate der Antworten, werden separat angelegt und können später im Diagramm ausgewählt werden (vgl. [40]).



Figure 2.6: Unterschrift

2.4.4 Fazit

Zwar bieten Chatbot-Plattformen bereits einige Funktionen, allerdings fokussieren diese sich vornehmlich auf Marketing, Vertrieb und Support (vgl. [10] [12] [23]). Dies kann die Umsetzung einer komplexen Therapie erschweren. Übliche Elemente, wie visuelle Analogskalen und Likert-Skalen, die häufig in der Psychologie Verwendung finden, können nur schwer oder gar nicht umgesetzt werden. Die Darstellungsformen der Konversationen verschiedener Chatbot-Plattformen haben unterschiedliche Vor- und Nachteile. Bäume und Diagramme bieten eine visuelle und leicht verständliche Übersicht eines Konversationsablaufs. Je größer und komplexer dieser Ablauf allerdings wird, umso unübersichtlicher wird eine Konversation. Bei großen Bäumen und Diagrammen werden in der Gesamtansicht die einzelnen Komponenten und Schriften zu klein und somit schwer lesbar für das menschliche Auge. Ist die Funktion eines hinein- und herauszoomens implementiert, erschwert sich das Verorten der Komponenten im Gesamtsystem (vgl. [19]). Blocksysteme bieten zusätzlich die visuelle Darstellung von Bedingungen, aber auch hier ist ein großer Konversationsablauf schnell unübersichtlich (vgl. [19]). Viele Elemente, die in Chatbot-Plattformen eingesetzt werden, könnten für einen Therapiemodellierungsansatz nützlich sein, da diese leicht nachvollziehbar und leicht in der Handhabung sind. Allerdings ist keine der bisherigen Chatbot-Plattformen derzeit geeignet, um eine komplexe Therapie umzusetzen ohne lange Einarbeitungszeit oder Einschränkungen in der Gestaltung.

Auch eine Umsetzung mit den sogenannten grafischen Programmiersprachen wäre möglich um eigene Chatbots zu entwickeln. Überwiegend gibt es diese für spezielle Domänen wie Elektrotechnik. Das Baukastenprinzip ist hier besonders interessant, da es verschiedene Funktionen visuell darstellt und später in Code übersetzt. Die visuelle Darstellung ist leicht verständlich und schnell zu Erlernen. *Blockly* bedient sich diesem Prinzip, um verschiedene Arten der Programmanweisungen verständlicher darzustellen. Allerdings erhält in dieser Form der Umsetzung ein komplexeres Programm oder System die zuvor genannten Probleme (vgl. [19]). Es gibt ebenfalls noch keine grafische Programmiersprache, die zur Umsetzung eines Therapiemodellierungsansatzes geeignet wäre.

Für das Beschreiben einer Konversation wäre auch die Nutzung einer Auszeichnungssprache möglich. Das Schreiben eines Konversationsfluss wirkt hier sehr natürlich, da es dem Chatten nahe kommt. Aber auch hier kann man leicht die Übersicht verlieren, da Verzweigungen in Konversationen nicht entsprechend dargestellt werden können, wie es beispielsweise bei Diagrammen möglich ist. Auch liefern nicht alle Auszeichnungssprachen den Umfang von Funktionen um komplexe Therapien darzustellen. Auch die Syntax und Fehlersuche wird zeitaufwändig sofern das

genutzte Programm zur Beschreibung keine oder eine rudimentäre Fehlerbehandlung mit sich bringt.

Im Bereich des Experience Samplings werden bereits verschiedene Ansätze verwendet, die eine grafische Programmierung oder das Verwenden einer Auszeichnungssprache integrieren. Hier liegt der Fokus auf der Entwicklung von Fragebögen die zu verschiedenen Zeiten getriggert werden. Derzeit gibt es keine Experience Sampling Software die Therapien gezielt in Form von Konversationen umsetzt. Jedoch können Fragebögen ein wichtiges Stilmittel einer Therapie darstellen.

Aufgrund der verschiedenen Vor- und Nachteile der vorgestellten Ansätze, wäre eine Kombination verschiedener Ansätze denkbar.

3. Konzeption

3.1 Anforderungsanalyse

Nur Fokus auf Funktionale Anforderungen

- Konversationen: Arten (Fragebogen, JIT Intervention, getriggerte Intervention, On-demand Intervention, Self-Monitoring, Notfallkoffer)
- Fragebogen (Offene und geschlossene Fragen)
- Eingabeformate der Nutzer (Text, Likert-Skala,...)

3.2 Ausarbeitung verschiedener Konzepte

4. Entwicklung der Modellierungssprache

5. Evaluation

5.1 Deskriptive Statistik

5.2 Beschreibung der Ergebnisse

5.2.1 Konstruktionsprinzip versus Konfigurationsprinzip

5.2.1.1 Konstruktionsprinzip

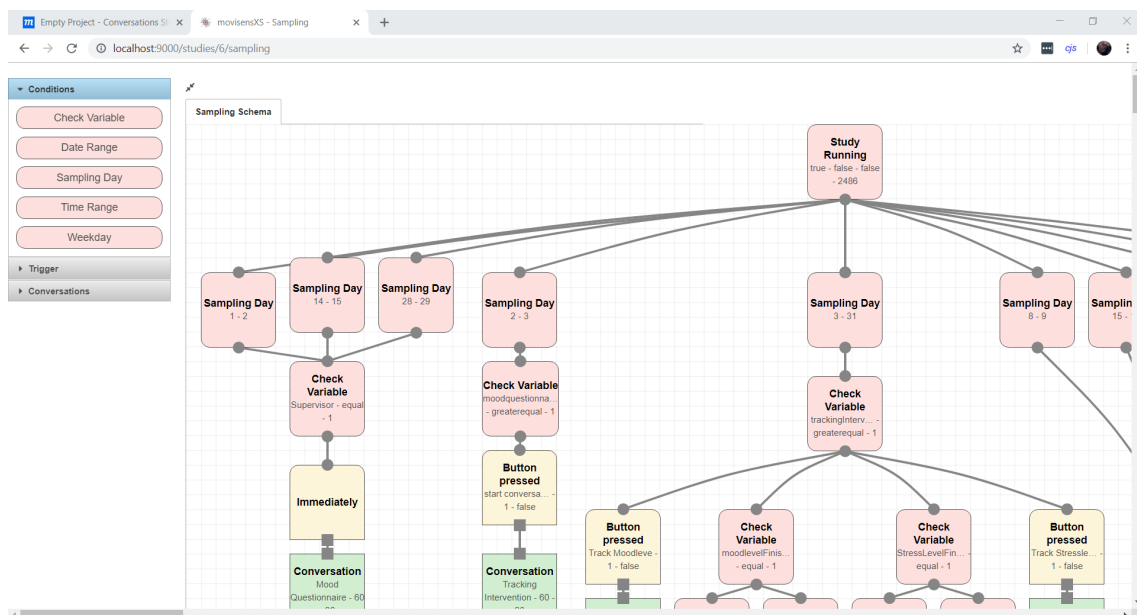


Figure 5.1: Architektur des *konstruktion*

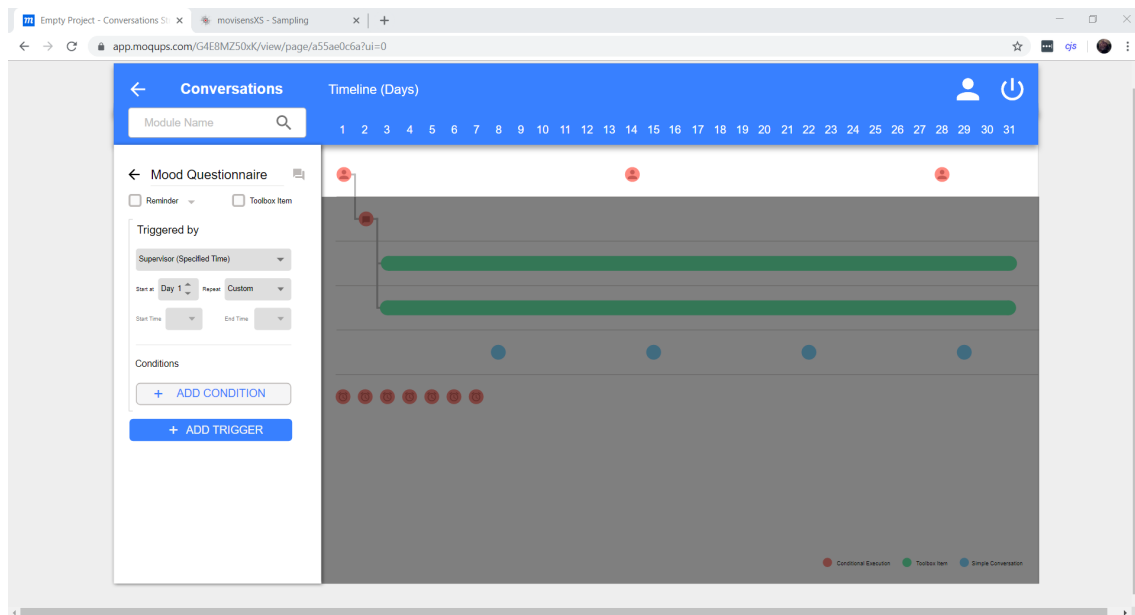


Figure 5.2: Architektur des *konfiguration*

5.2.1.2 Konfigurationsprinzip

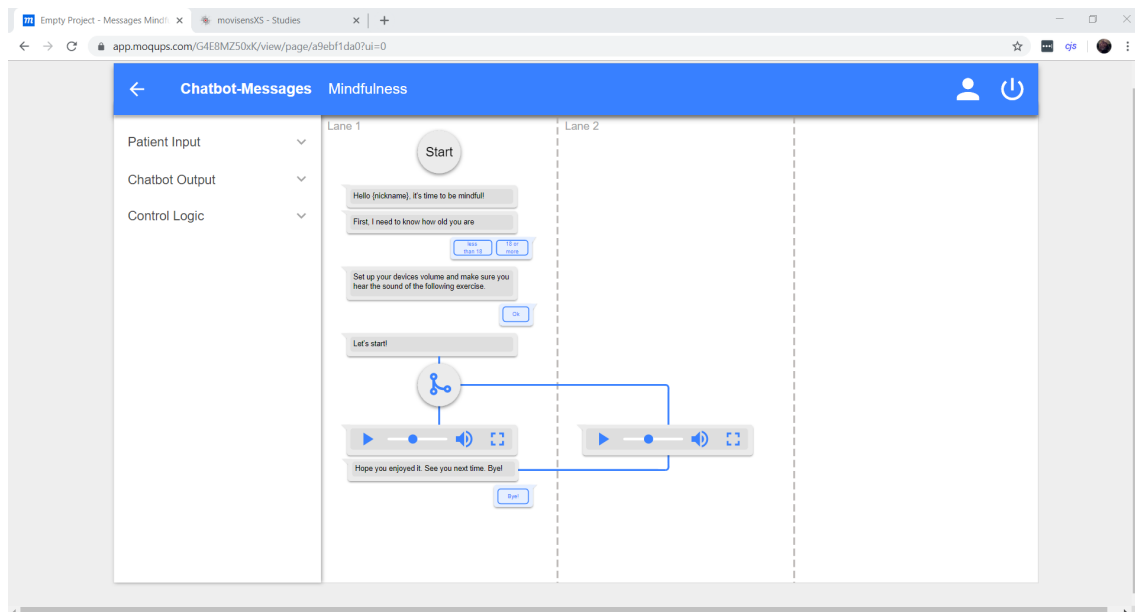
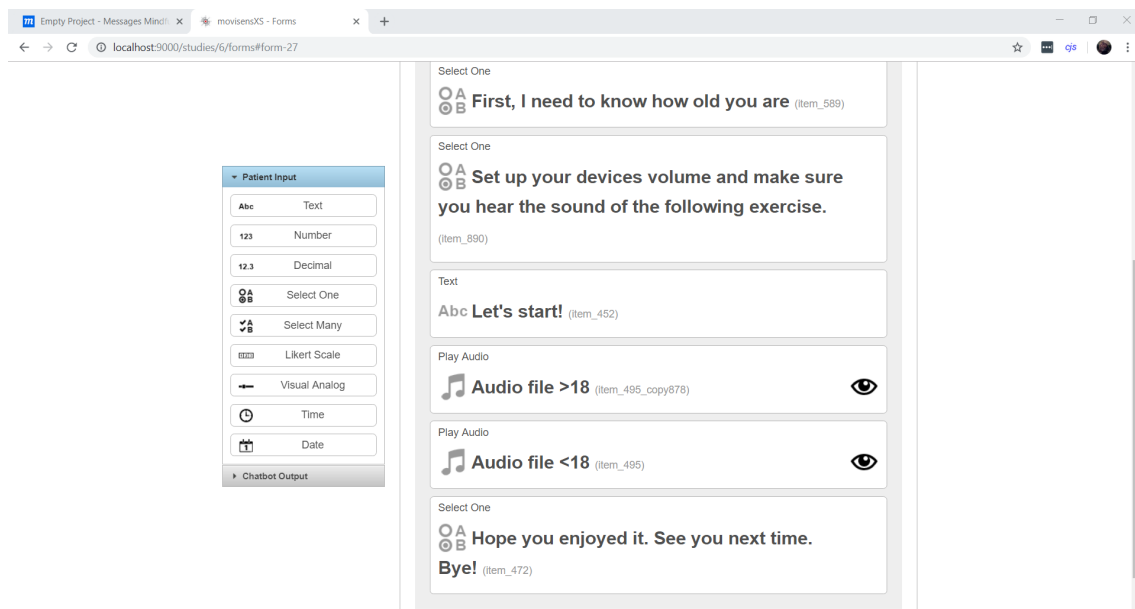
5.2.1.3 Gegenüberstellung

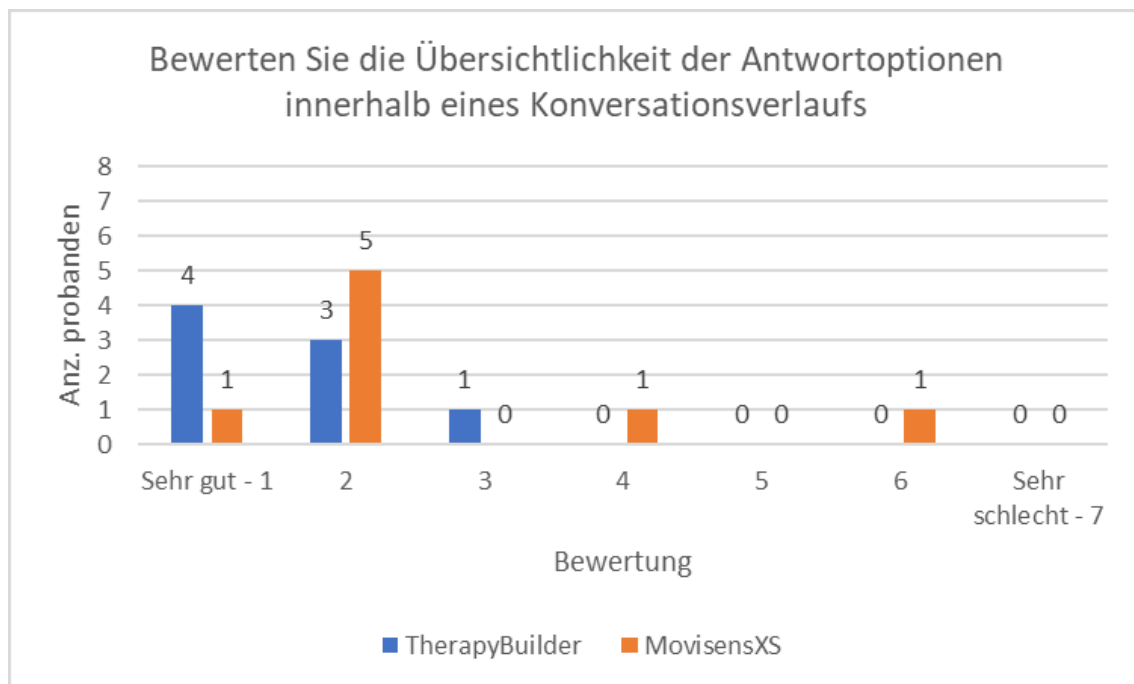
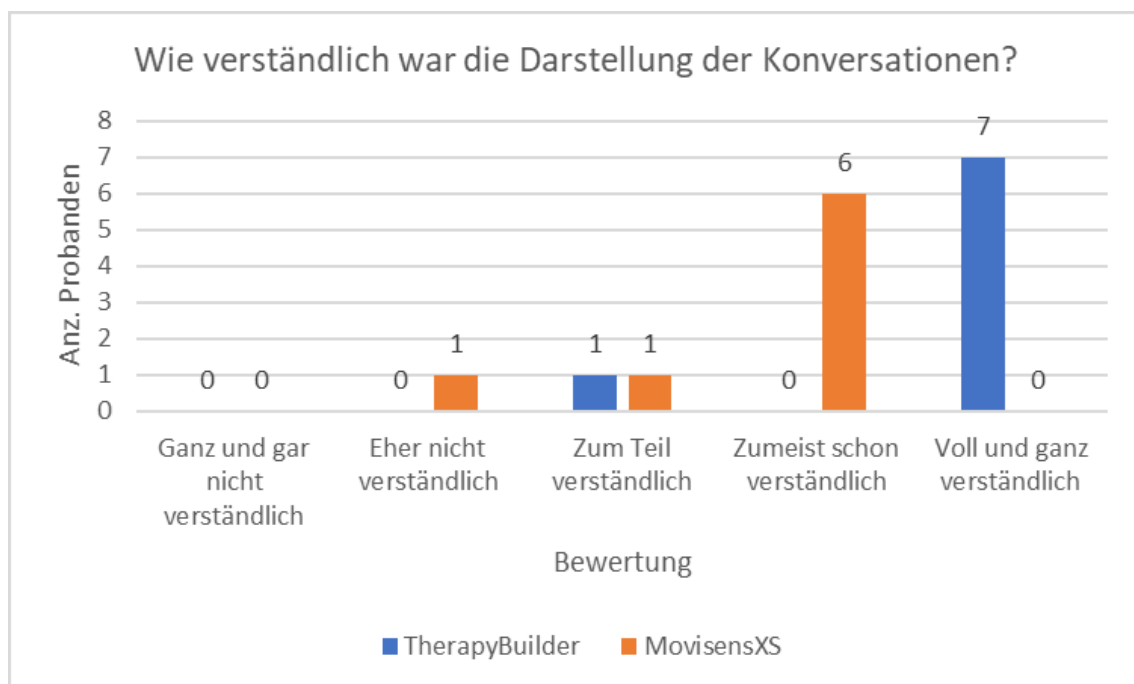
5.2.2 Sprünge versus Sichtbarkeitsregeln

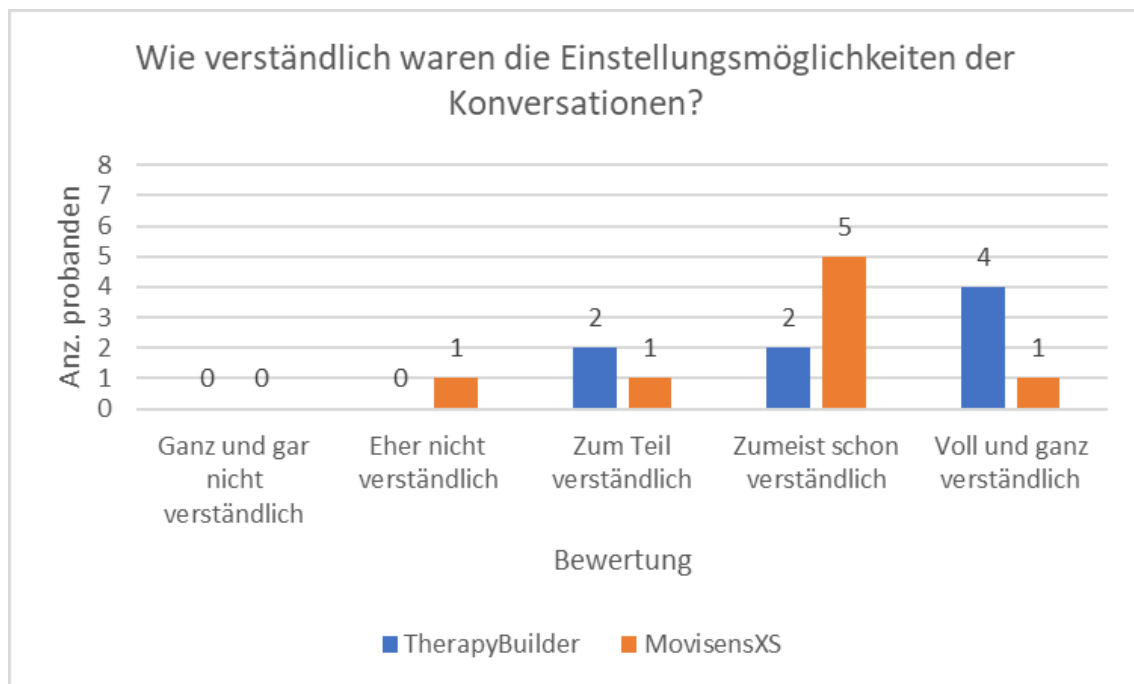
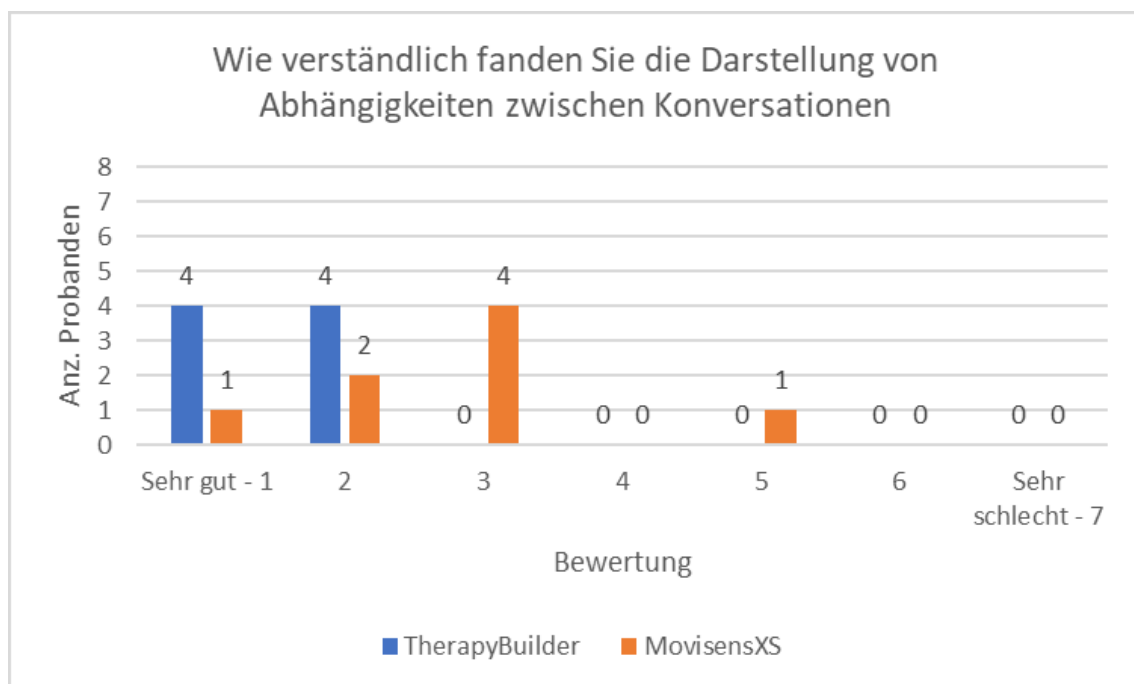
5.2.2.1 Sprünge

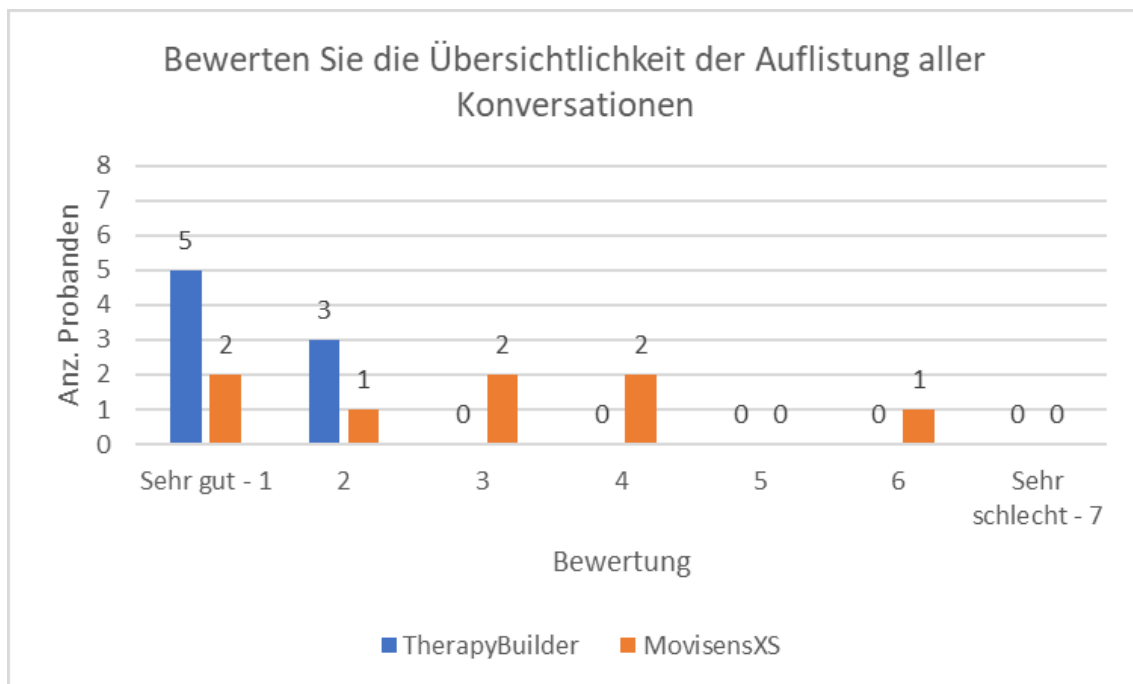
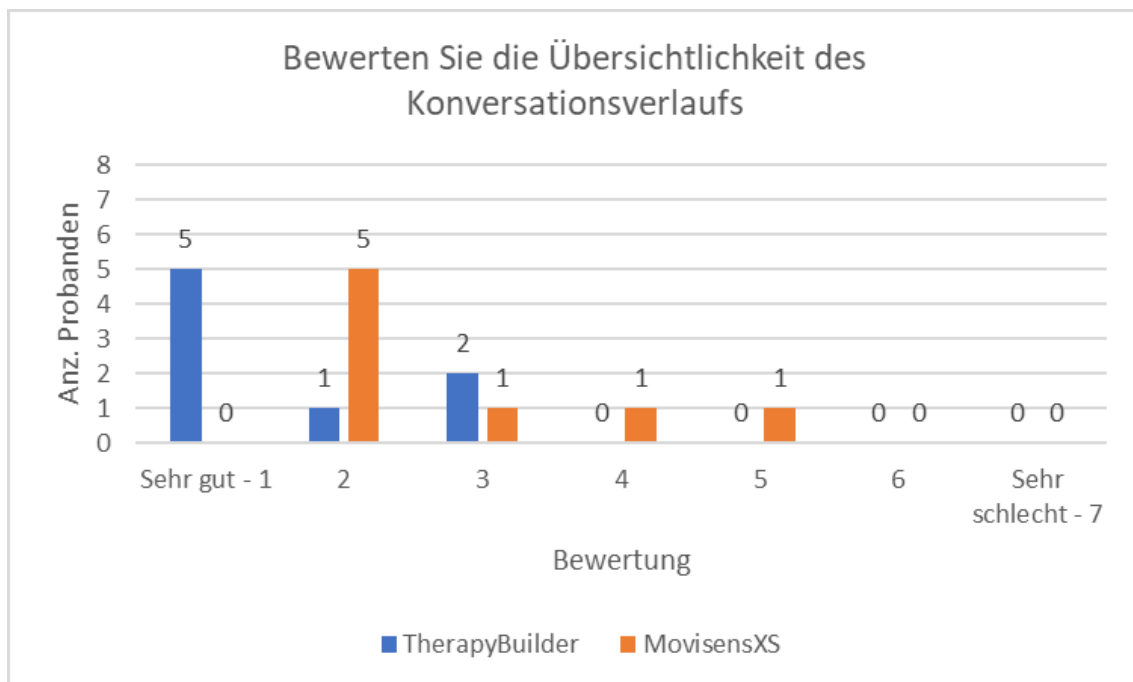
5.2.2.2 Sichtbarkeitsregeln

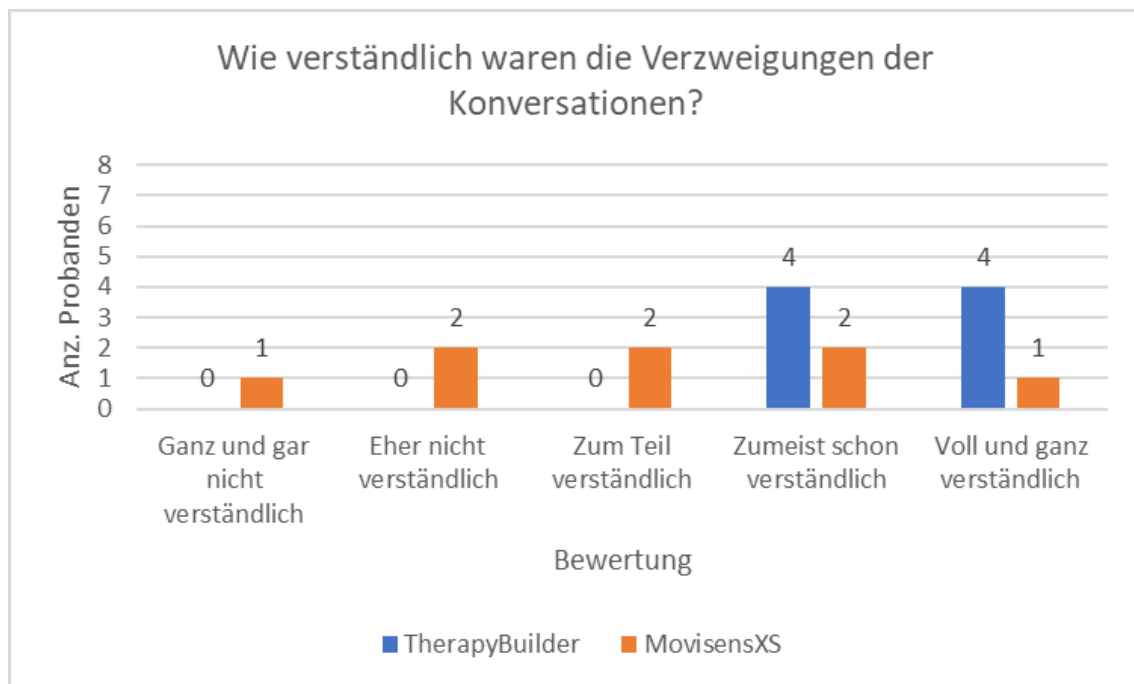
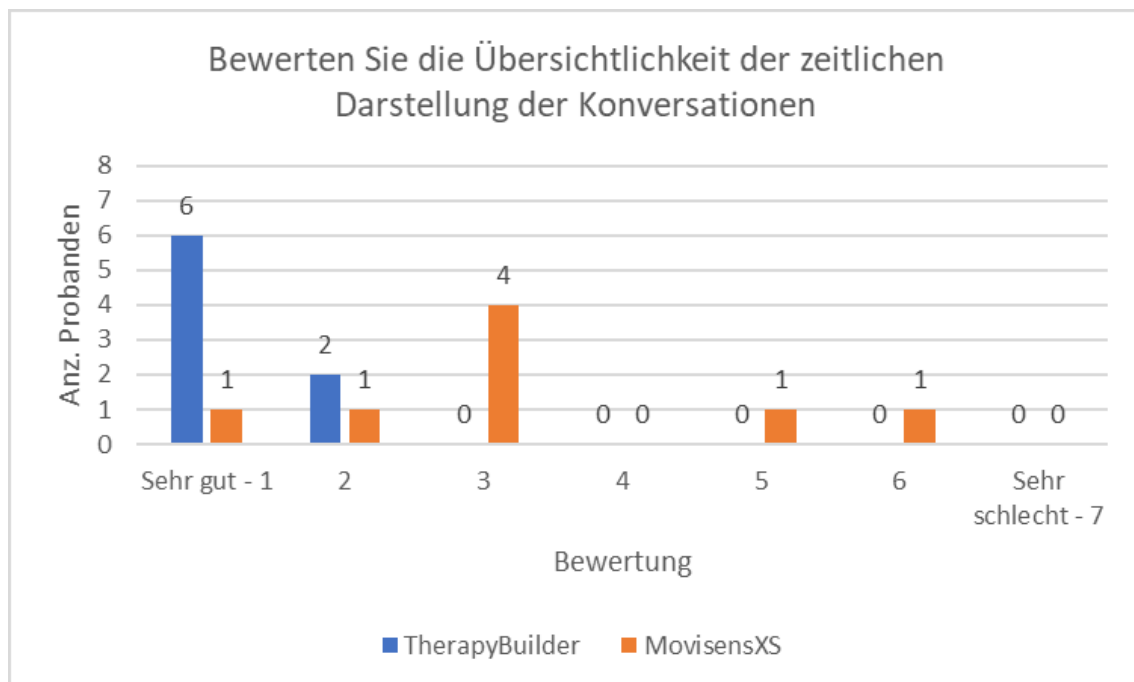
5.2.2.3 Gegenüberstellung

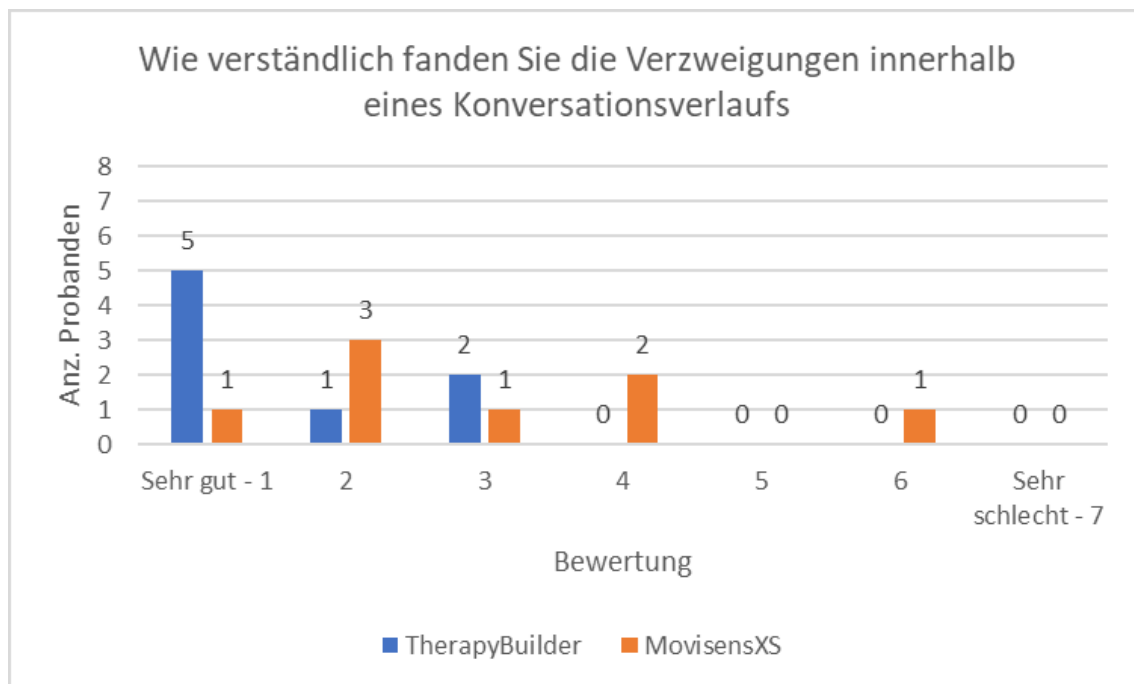
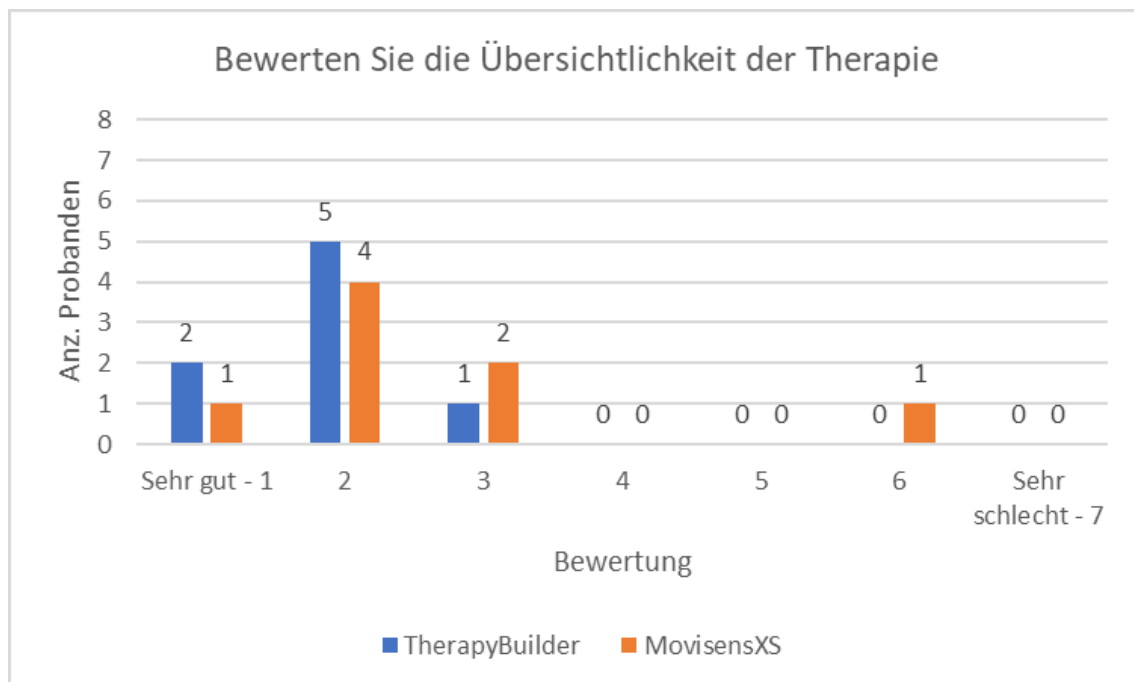
Figure 5.3: Architektur des *spruenge*Figure 5.4: Architektur des *sichtbarkeit*

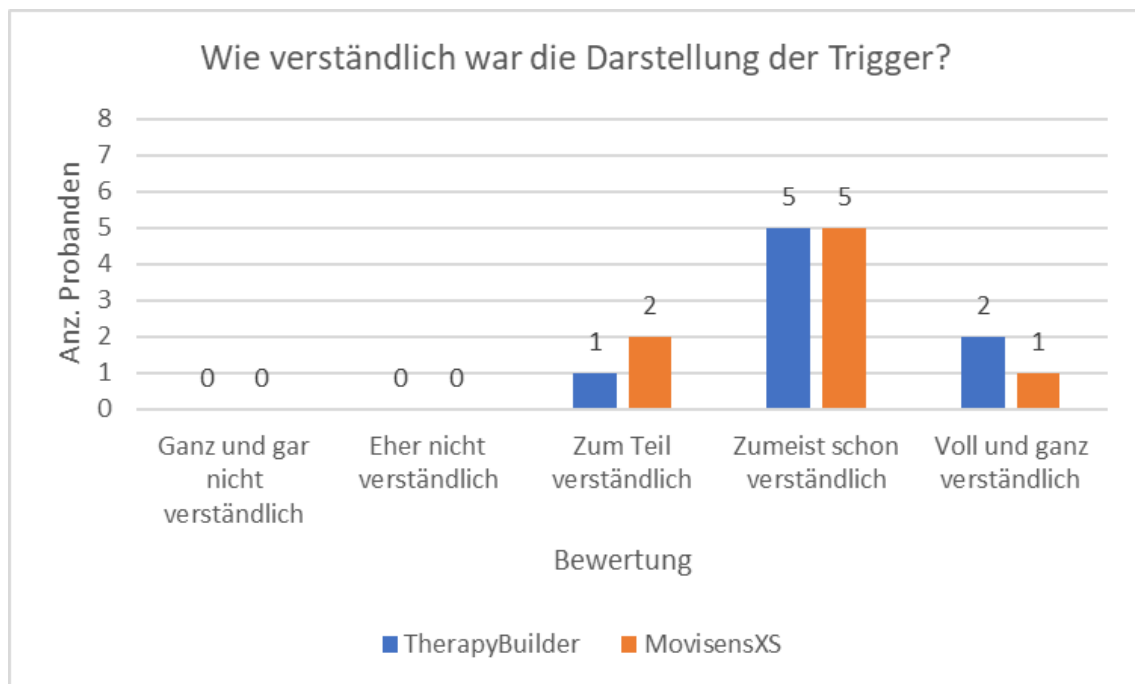
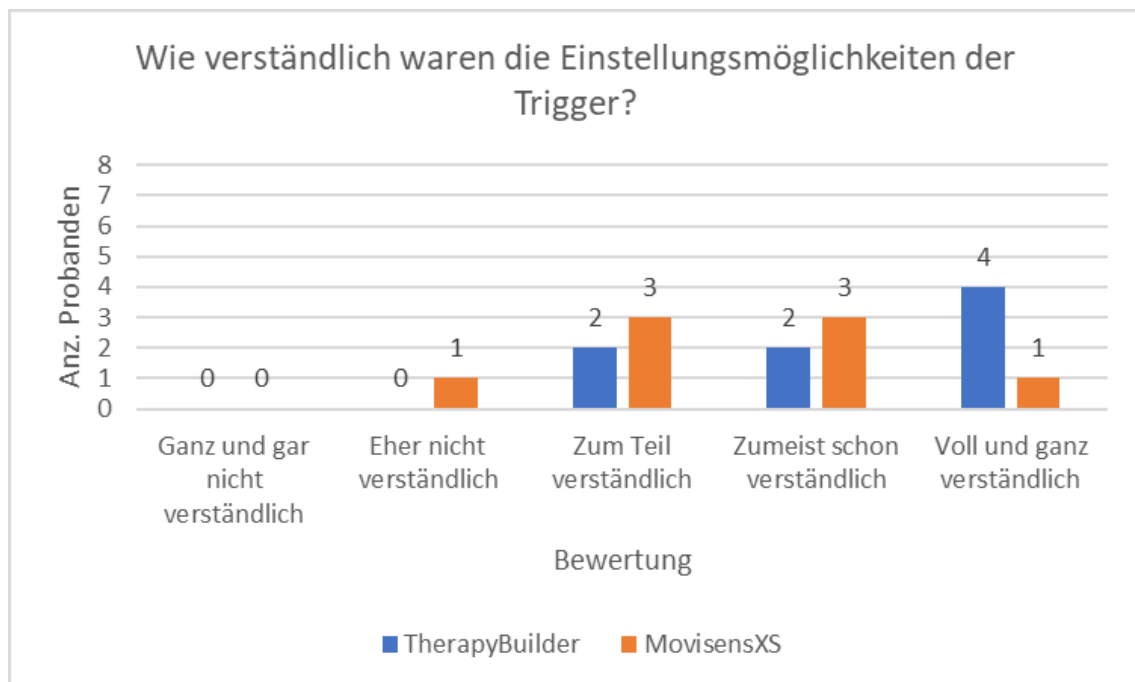
Figure 5.5: Architektur des *sichtbarkeit*Figure 5.6: Architektur des *konfiguration*

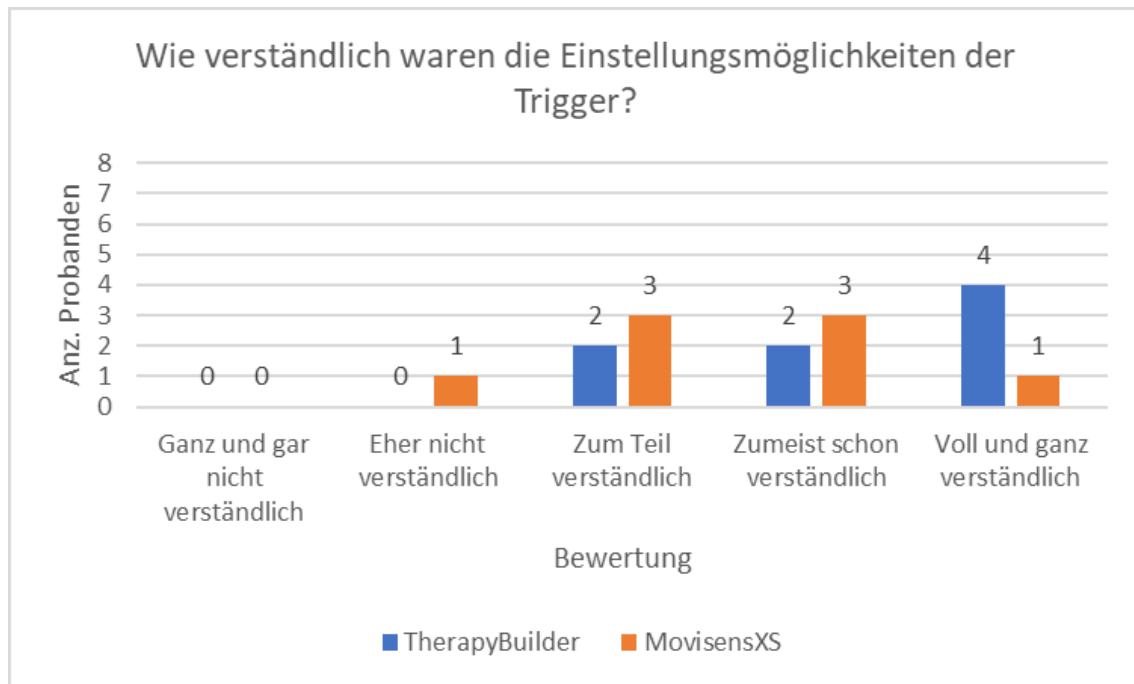
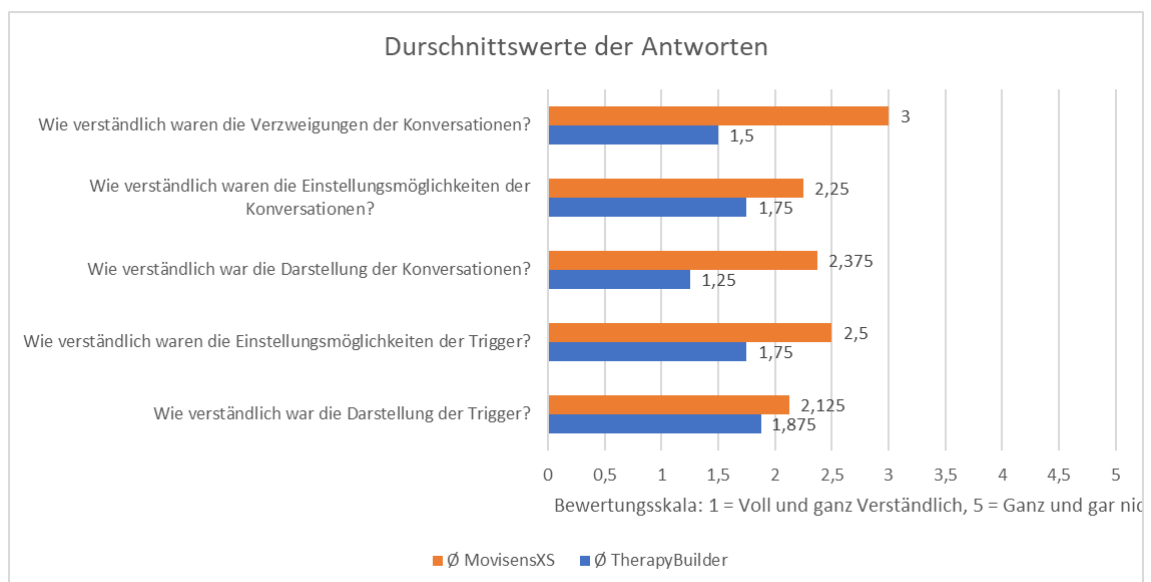
Figure 5.7: Architektur des *konfiguration*Figure 5.8: Architektur des *konfiguration*

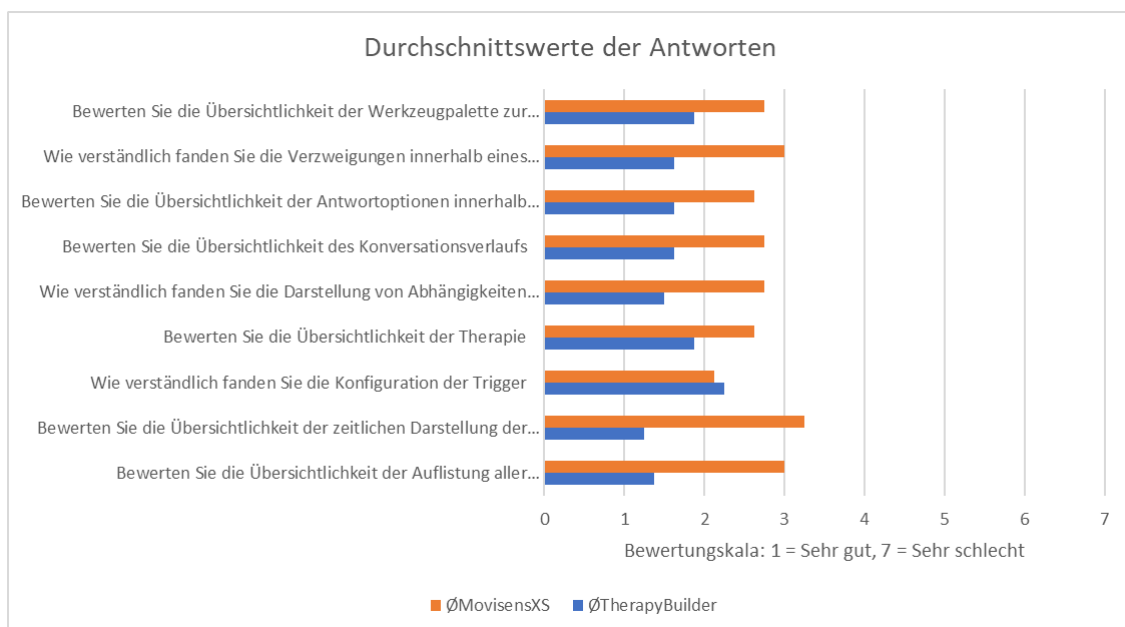
Figure 5.9: Architektur des *konfiguration*Figure 5.10: Architektur des *konfiguration*

Figure 5.11: Architektur des *konfiguration*Figure 5.12: Architektur des *konfiguration*

Figure 5.13: Architektur des *konfiguration*Figure 5.14: Architektur des *konfiguration*

Figure 5.15: Architektur des *konfiguration*Figure 5.16: Architektur des *konfiguration*

Figure 5.17: Architektur des *konfiguration*Figure 5.18: Architektur des *konfiguration*

Figure 5.19: Architektur des *konfiguration*

6. Ergebnisse

6.1 Deskriptive Statistik

6.2 Beschreibung der Ergebnisse

6.2.1 Konstruktionsprinzip versus Konfigurationsprinzip

6.2.1.1 Konstruktionsprinzip

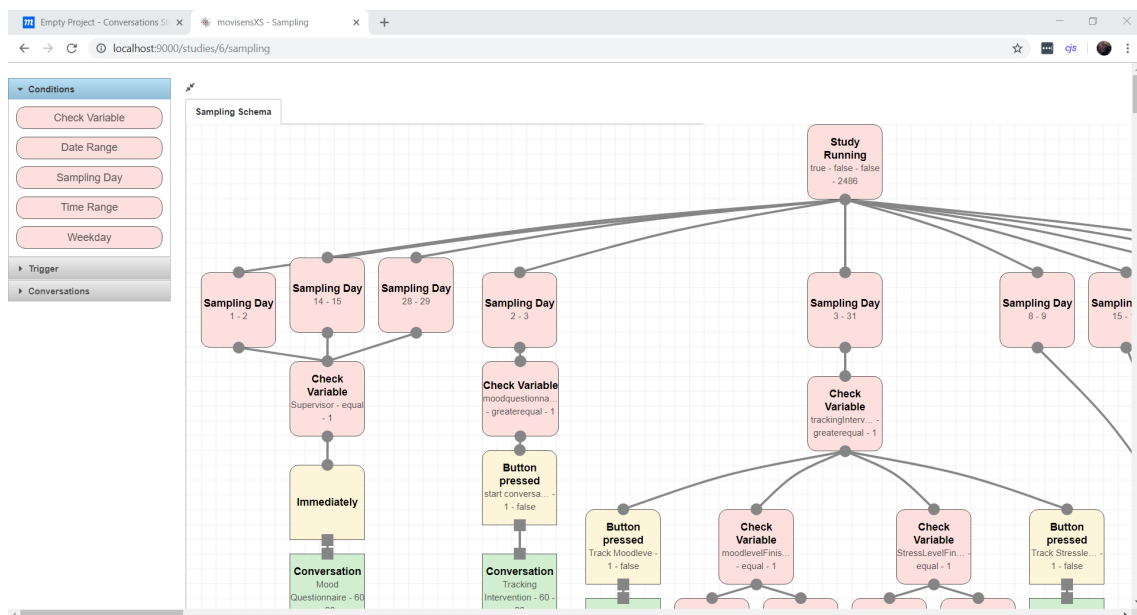
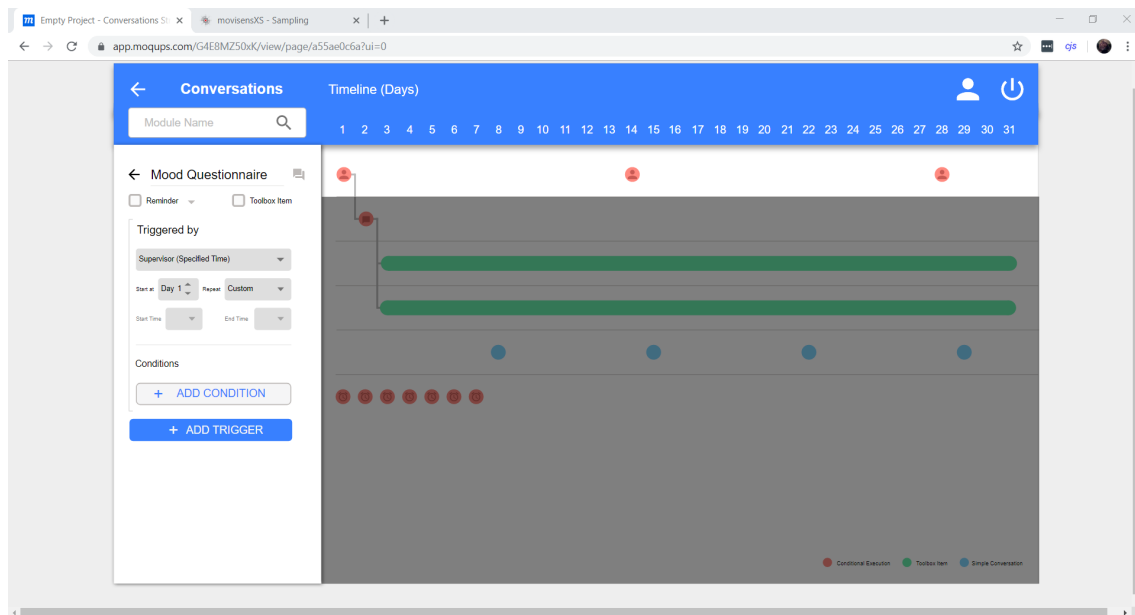
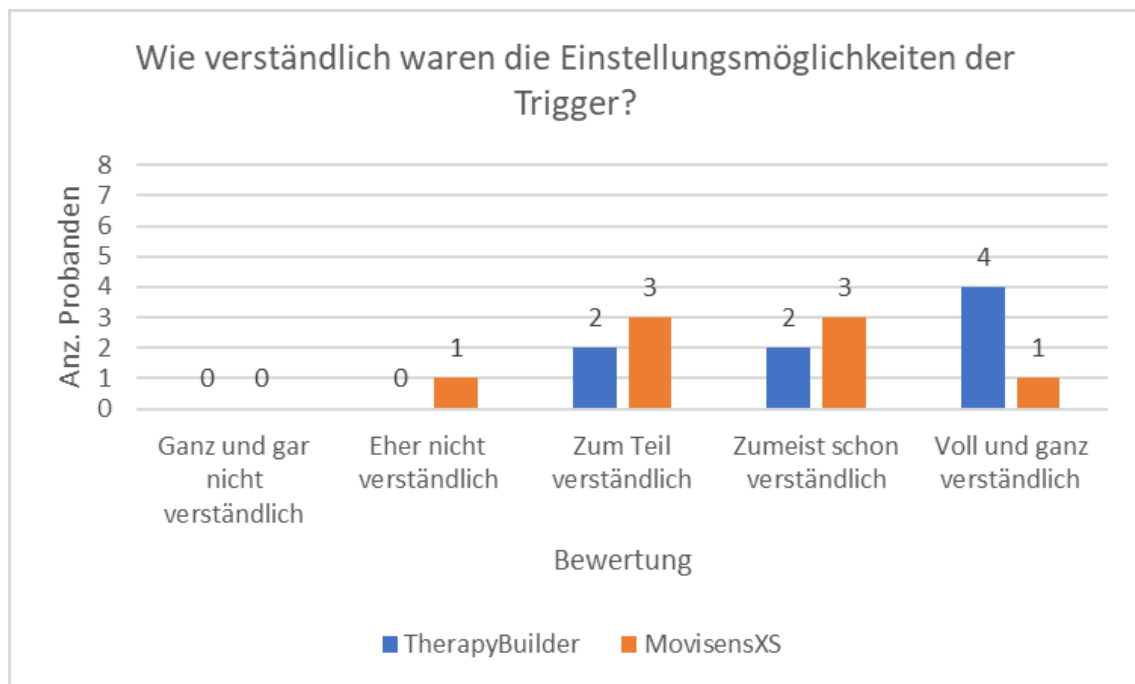


Figure 6.1: Architektur des *konstruktion*

Figure 6.2: Architektur des *konfiguration*Figure 6.3: Architektur des *konfiguration*

6.2.1.2 Konfigurationsprinzip

6.2.1.3 Gegenüberstellung

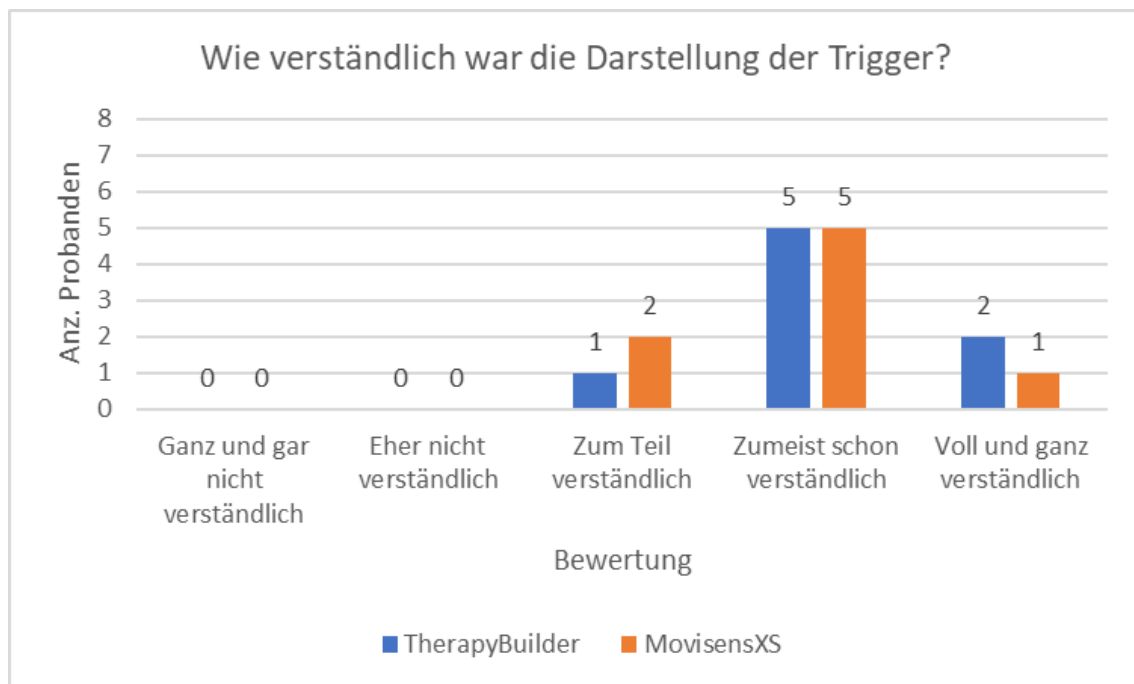
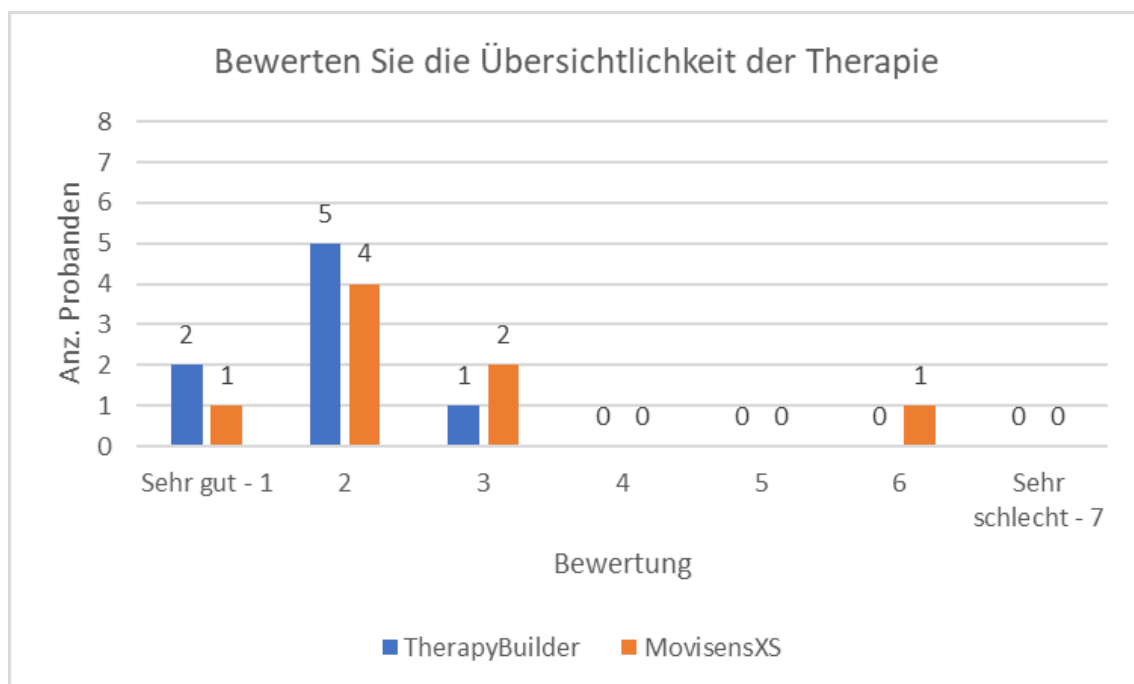
6.2.2 Sprünge versus Sichtbarkeitsregeln

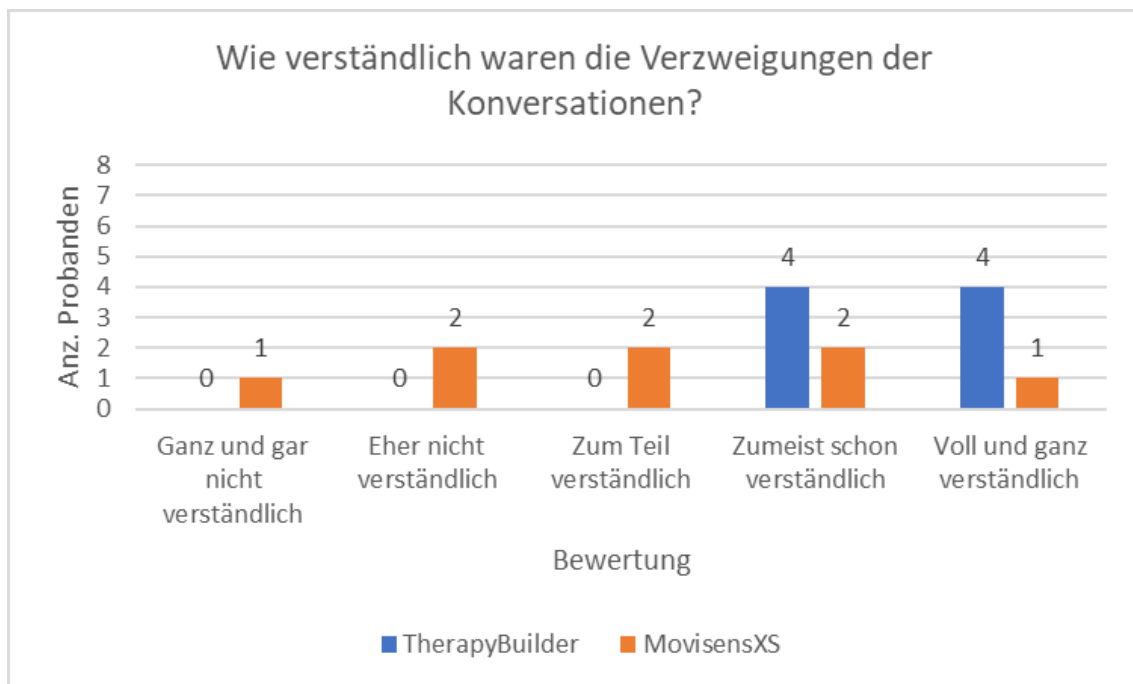
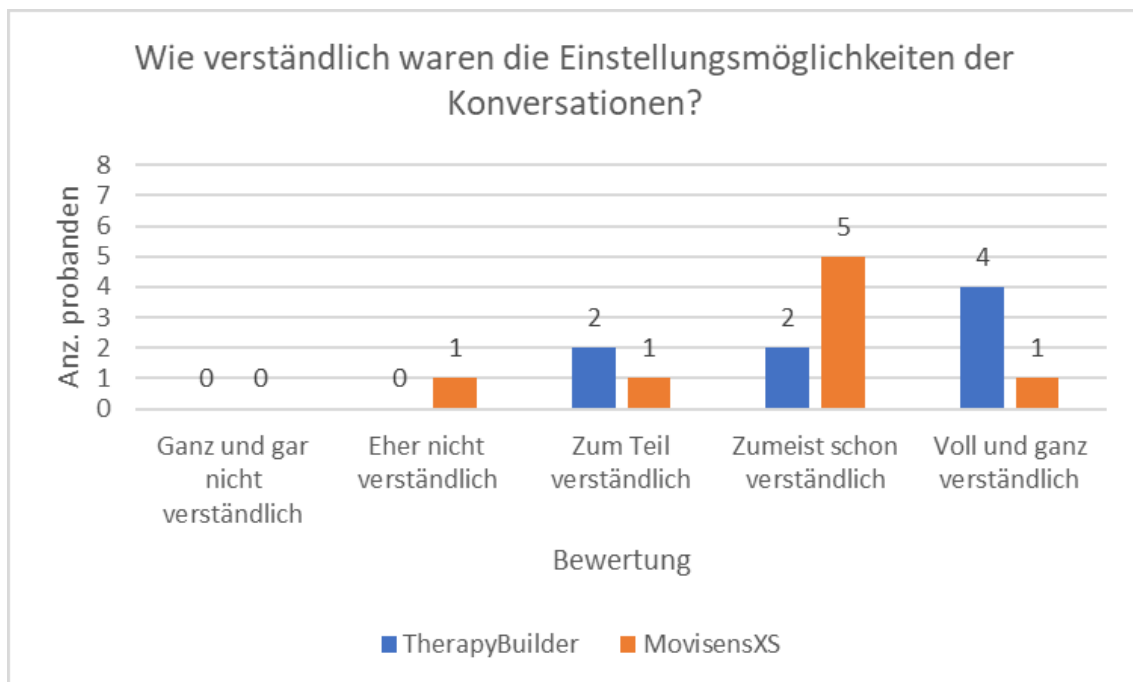
6.2.2.1 Sprünge

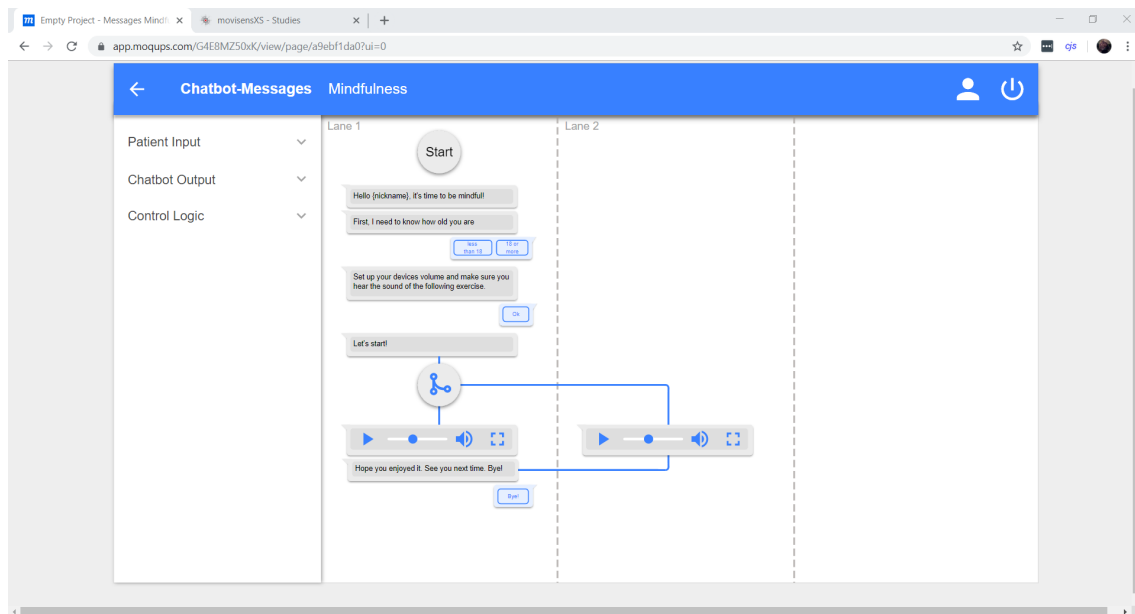
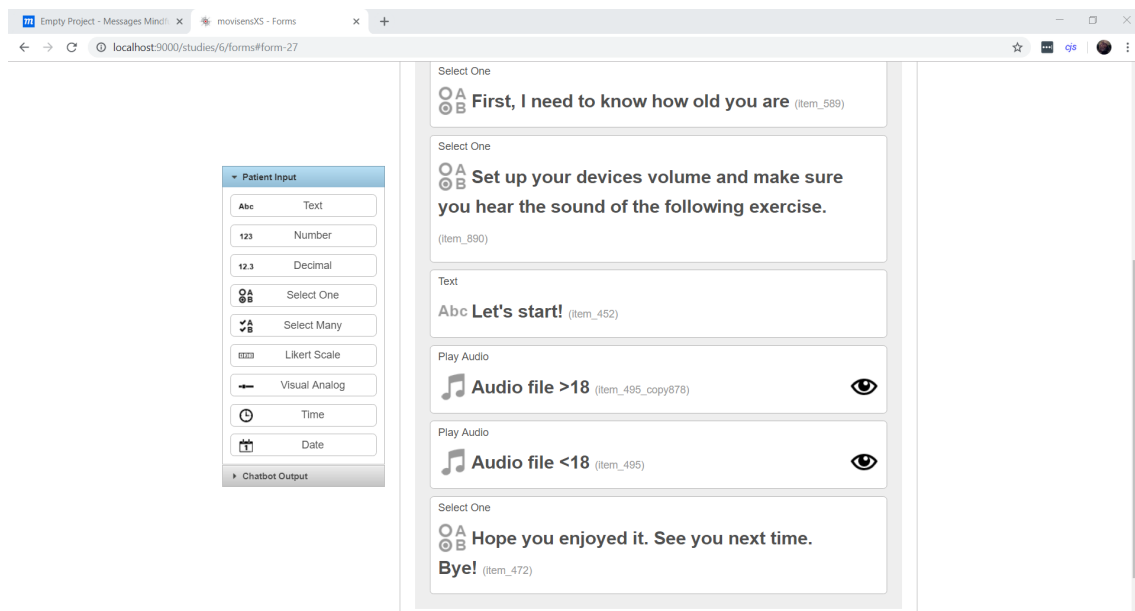
6.2.2.2 Sichtbarkeitsregeln

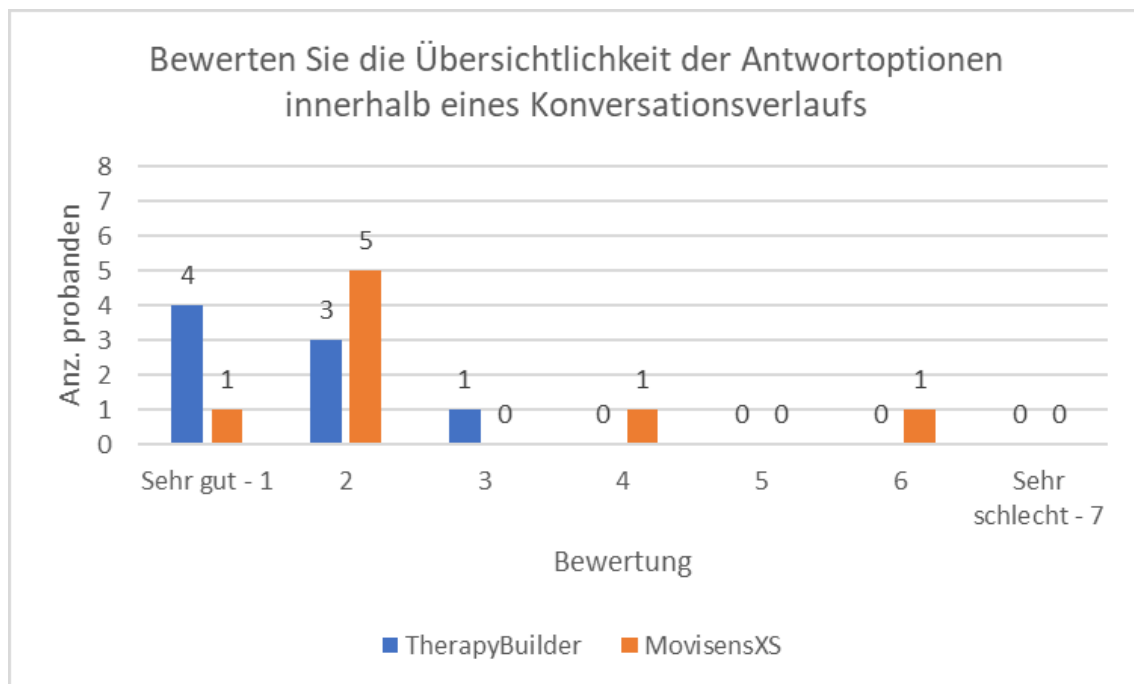
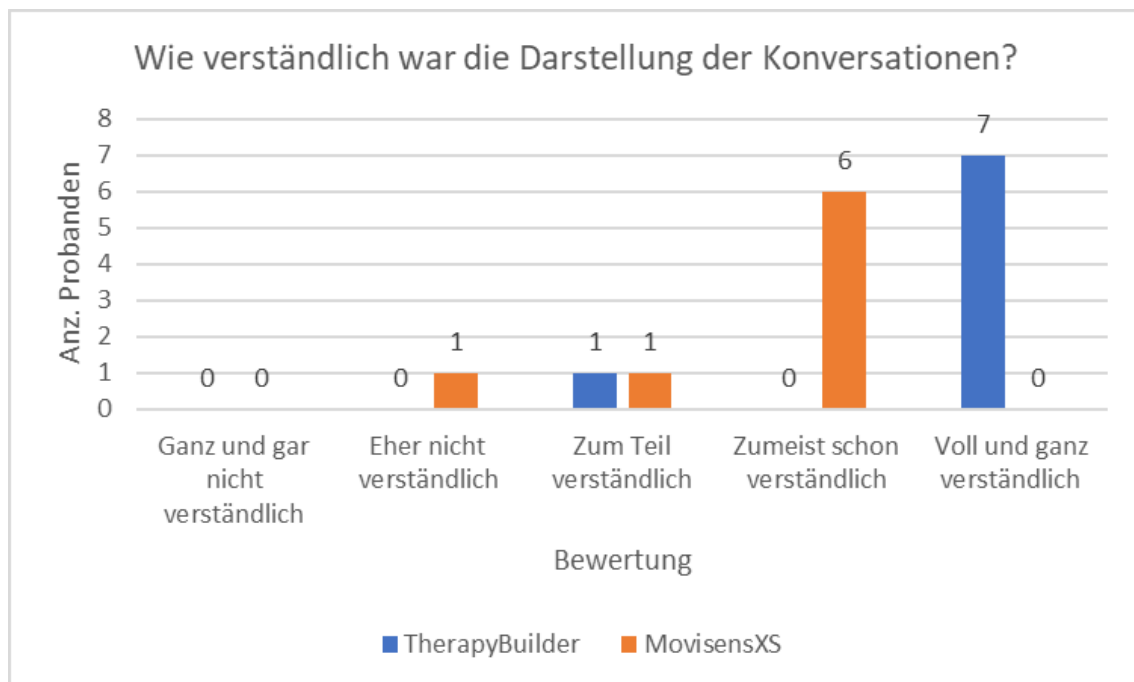
6.2.2.3 Gegenüberstellung

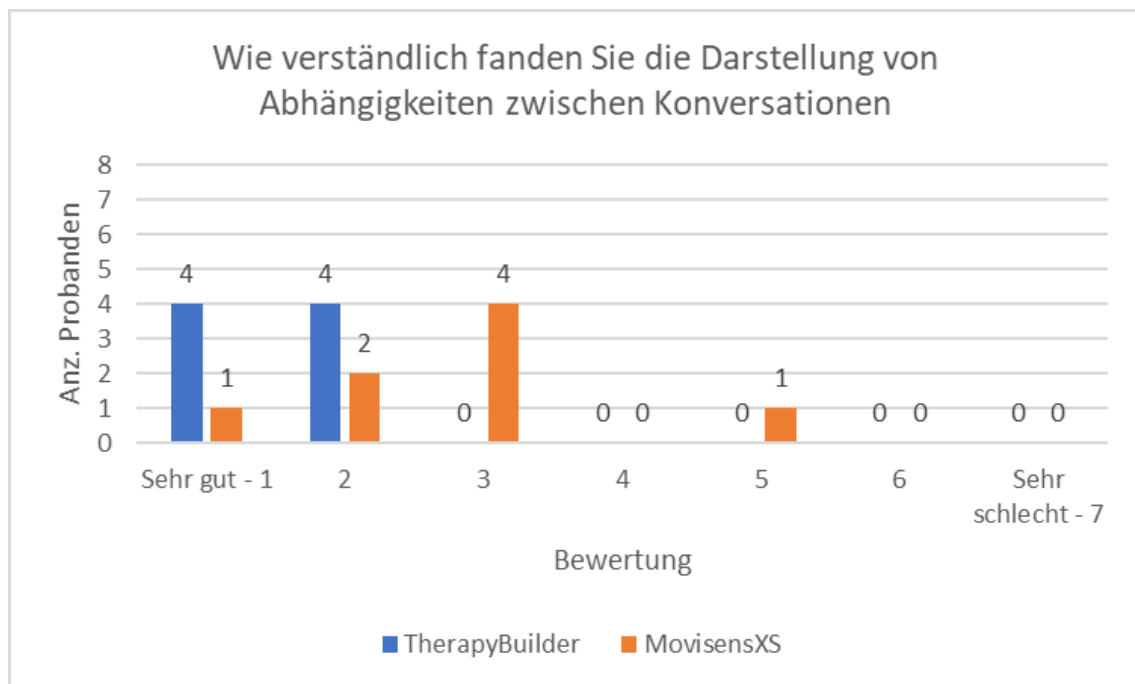
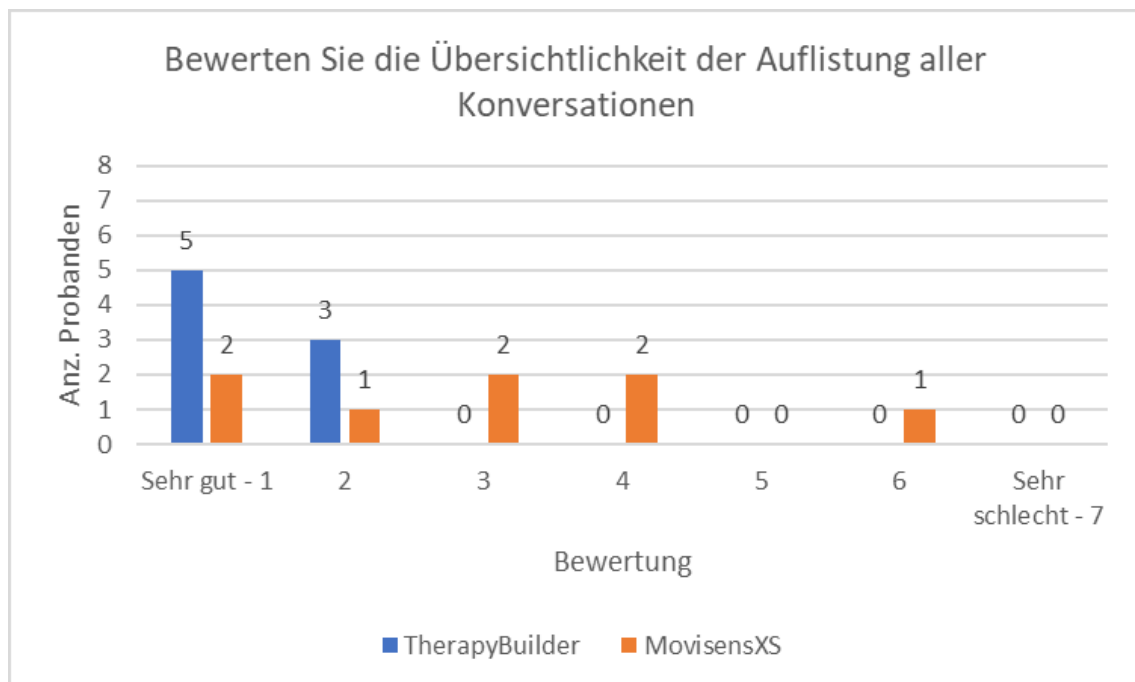
6.3 Kritische Reflexion

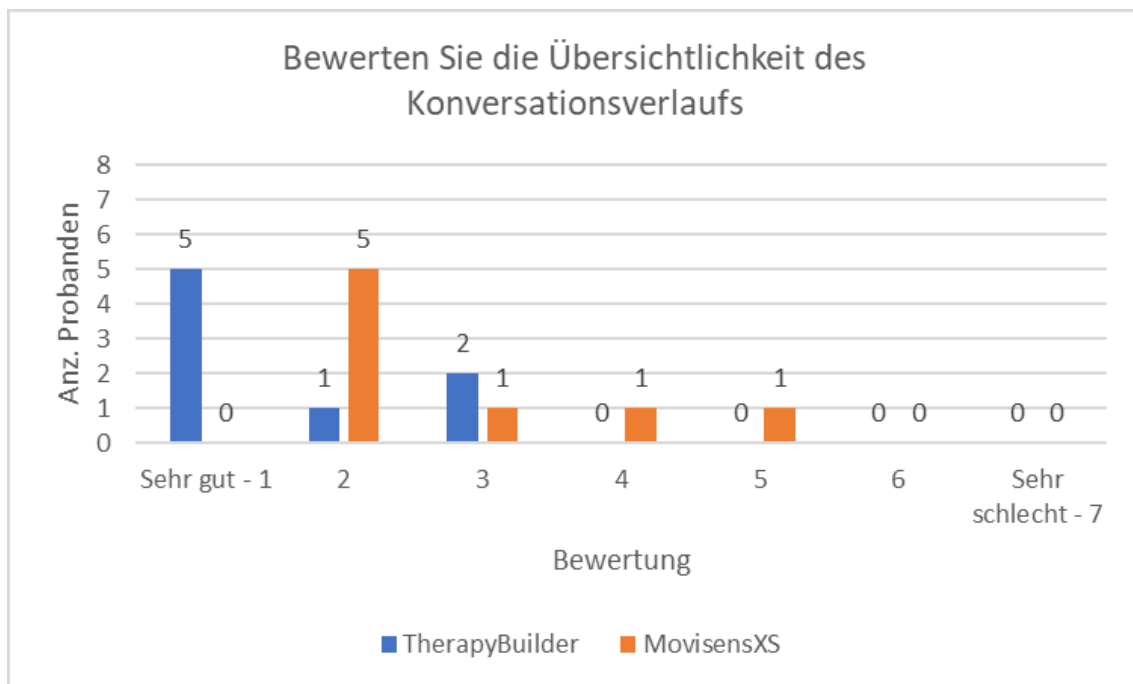
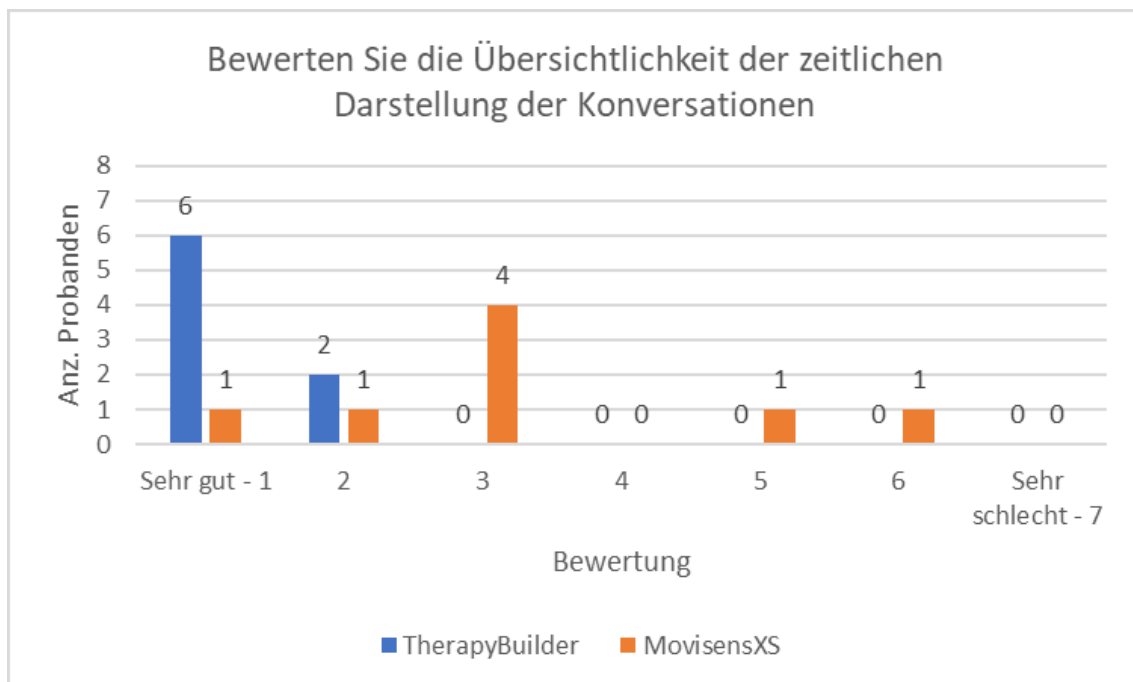
Figure 6.4: Architektur des *konfiguration*Figure 6.5: Architektur des *konfiguration*

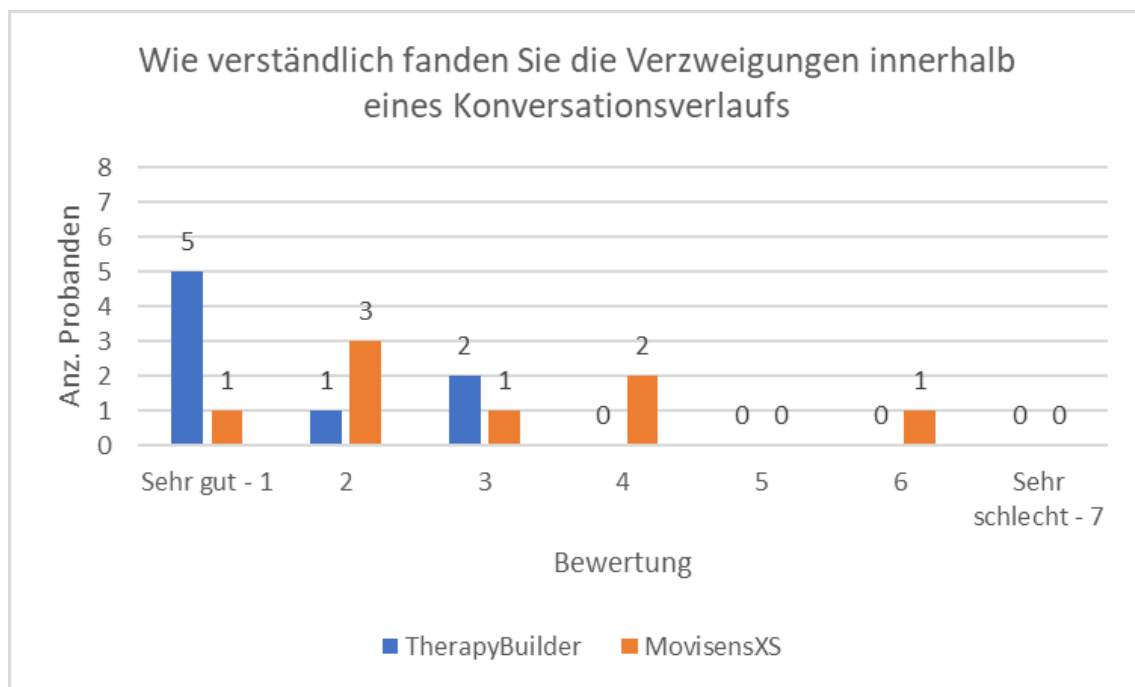
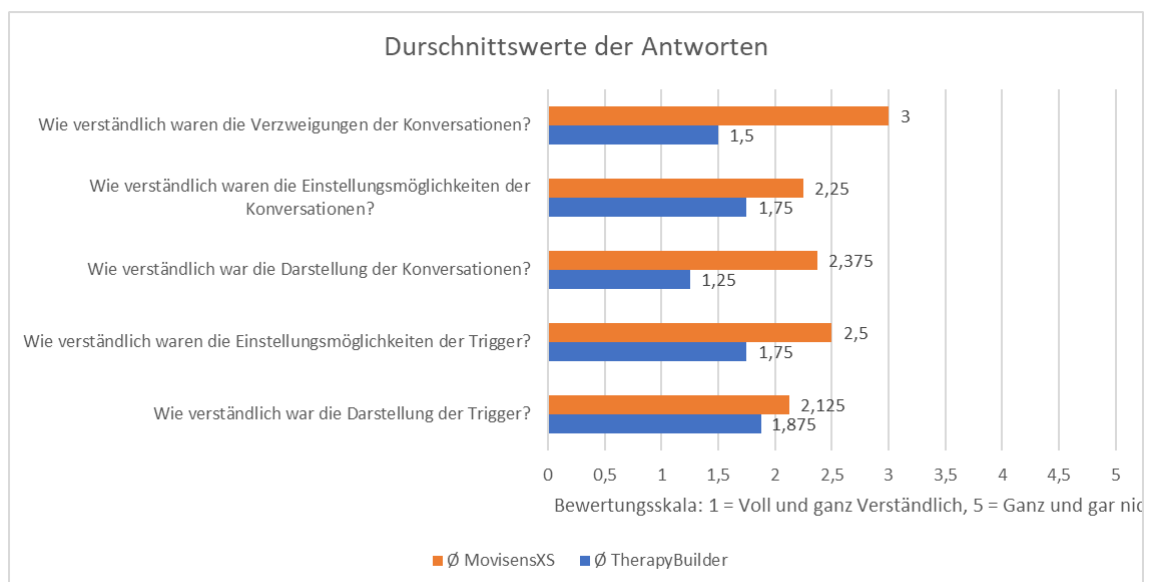
Figure 6.6: Architektur des *konfiguration*Figure 6.7: Architektur des *konfiguration*

Figure 6.8: Architektur des *spruenge*Figure 6.9: Architektur des *sichtbarkeit*

Figure 6.10: Architektur des *sichtbarkeit*Figure 6.11: Architektur des *konfiguration*

Figure 6.12: Architektur des *konfiguration*Figure 6.13: Architektur des *konfiguration*

Figure 6.14: Architektur des *konfiguration*Figure 6.15: Architektur des *konfiguration*

Figure 6.16: Architektur des *konfiguration*Figure 6.17: Architektur des *konfiguration*

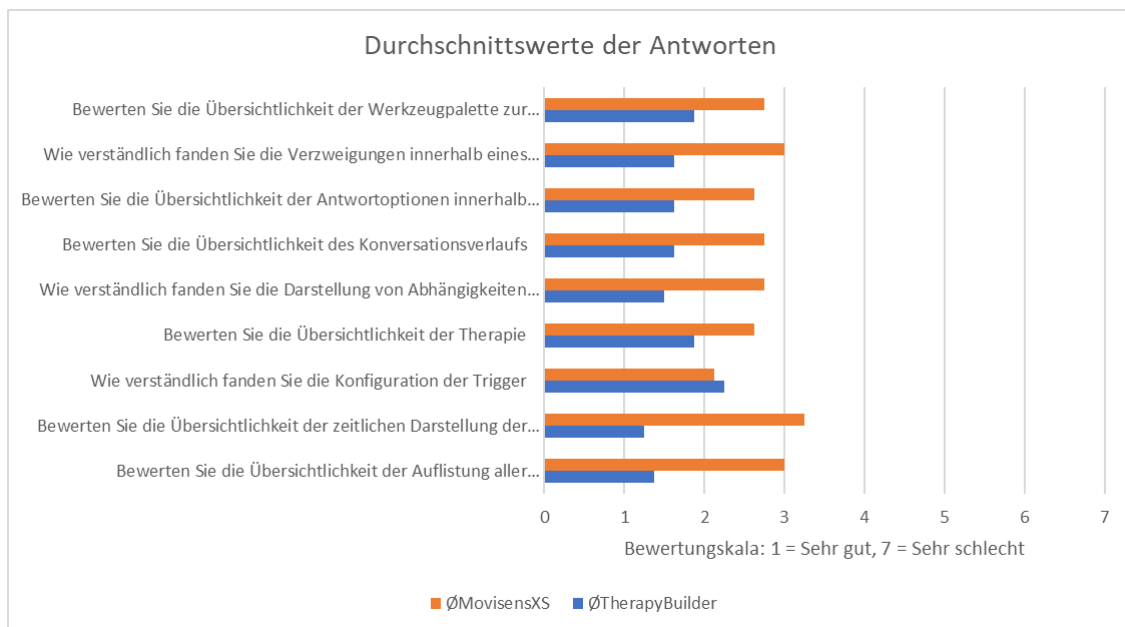


Figure 6.18: Architektur des *konfiguration*

7. Ausblick

Bibliography

- [1] *App Store - Apple (DE)*. <https://www.apple.com/de/ios/app-store/>. (Accessed on 03/05/2019).
- [2] *Assessment, ambulantes - Dorsch Lexikon der Psychologie - Verlag Hans Huber*. <https://m.portal.hogrefe.com/dorsch/assessment-ambulantes/>. (Accessed on 08/08/2019).
- [3] *BBCode.org, bbcode users guide and tricks on implementing it*. <https://www.bbcode.org/>. (Accessed on 11/28/2018).
- [4] Nikolaos Batalas et al. "Formal representation of ambulatory assessment protocols in HTML5 for human readability and computer execution". In: *Behavior Research Methods* (2018). ISSN: 15543528.
- [5] *Blockly | Google Developers*. <https://developers.google.com/blockly/>. (Accessed on 11/27/2018).
- [6] *Blockly | Google Developers*. <https://developers.google.com/blockly/>. (Accessed on 11/28/2018).
- [7] *Botsify - Create artificial intelligent chatbots without coding*. <https://botsify.com/>. (Accessed on 11/27/2018).
- [8] Martin Campbell-Kelly and William Aspray. *Computer : a history of the information machine*. 2. ed. Previous ed.: New York: Basic Books, 1996. Boulder: Westview, 2004. ISBN: 0-8133-4264-3.
- [9] Martin Campbell-Kelly and Daniel D. [author] Garcia-Swartz, eds. *From mainframes to smartphones : a history of the international computer industry*. Includes bibliographical references and index. - Print version record; The aim of this book is to provide a compact and up-to-date business and economic history of the computer industry. The reader we have in mind is someone who wants to make a quick study of the computer industry... In our quest for brevity, we have assumed that our readers are themselves information-technology users and are familiar with the everyday vocabulary of computing (such as operating systems, disk drives, and broadband), and that they are aware of recent gadgetry such as tablet computers and smartphones—Page 1. Cambridge, Massachusetts, 2015.
- [10] *Chatfuel dashboard*. <https://dashboard.chatfuel.com/#/bot/5bea8fd30ecd9f4c8c0dd458/structure/5bea8fd30ecd9f4c8c0dd466>. (Accessed on 11/27/2018).
- [11] *Choregraphe Suite — Aldebaran 2.4.3.28-r2 documentation*. <http://doc.aldebaran.com/2-4/software/choregraphe/index.html>. (Accessed on 11/28/2018).
- [12] *Converse.AI*. <http://www.converse.ai/>. (Accessed on 11/27/2018).

- [13] *DRAKON Editor*. <http://drakon-editor.sourceforge.net/>. (Accessed on 11/28/2018).
- [14] *Dialogflow*. <https://console.dialogflow.com/api-client/#/login>. (Accessed on 11/27/2018).
- [15] Kathleen Kara Fitzpatrick, Alison Darcy, and Molly Vierhile. “Delivering Cognitive Behavior Therapy to Young Adults With Symptoms of Depression and Anxiety Using a Fully Automated Conversational Agent (Woebot): A Randomized Controlled Trial”. In: *JMIR Mental Health* 4.2 (2017), e19. ISSN: 2368-7959. arXiv: 9809069v1 [arXiv:gr-qc].
- [16] *Getting Started | Markdown Guide*. <https://www.markdownguide.org/getting-started/>. (Accessed on 11/28/2018).
- [17] *Google Allo - A smart messaging app*. <https://allo.google.com/>. (Accessed on 02/27/2019).
- [18] Jonathan Gratch and Gale Lucas. “It ’ s Only a Computer : The Impact of Human-agent Interaction in Clinical Interviews”. In: Aamas (2014).
- [19] Kasper Hornbæk et al. “NAVIGATION PATTERNS AND USABILITY OF ZOOMABLE USER INTERFACES WITH AND WITHOUT”. In: (2003).
- [20] *Kassen-Schnittstelle Table Duck: Bestellen via Messenger | GASTROFIX*. <https://www.gastrofix.com/de/kassen-schnittstelle-table-duck/>. (Accessed on 02/27/2019).
- [21] *Katalog - IBM Cloud*. <https://console.bluemix.net/catalog/?category=ai&search=label:lite>. (Accessed on 11/27/2018).
- [22] *MPG - nichtamtliches Inhaltsverzeichnis*. <http://www.gesetze-im-internet.de/mpg/>. (Accessed on 05/28/2019).
- [23] *ManyChat - Startseite*. <https://www.facebook.com/ManyChat/>. (Accessed on 03/06/2019).
- [24] *ManyChat*. <https://manychat.com>. (Accessed on 11/27/2018).
- [25] *NXT Software Download - Downloads - Mindstorms LEGO.com*. <https://www.lego.com/de-de/mindstorms/downloads/nxt-software-download>. (Accessed on 11/28/2018).
- [26] *OSF | Sabrina Thai - ExperienceSampler SPSP San Antonio 2017 Draft 1 FINAL To Post.pdf*. <https://osf.io/3am7g/>. (Accessed on 11/28/2018).
- [27] *Pure Data — Pd Community Site*. <https://puredata.info/>. (Accessed on 11/28/2018).
- [28] Daniel Rough and Aaron Quigley. “Jeeves - An Experience Sampling Study Creation Tool”. In: *BCS Health Informatics Scotland (HIS)* (2017), pp. 1–10.
- [29] *Simulink - Simulation und Model-Based Design - MATLAB & Simulink*. <https://de.mathworks.com/products/simulink.html>. (Accessed on 11/28/2018).
- [30] *Smartphone-Markt: Konjunktur und Trends*. <https://www.bitkom.org/sites/default/files/file/import/Bitkom-Presskonferenz-Smartphone-Markt-Konjunktur-und-Trends-22-02-2017-Praesentation.pdf>. (Accessed on 05/28/2019).
- [31] *Snap! (Build Your Own Blocks) 4.2*. <https://snap.berkeley.edu/>. (Accessed on 11/28/2018).

- [32] *SwissEduc - Informatik - JavaKara: Einführung in Java*. <https://www.swisseduc.ch/informatik/karatojava/javakara/>. (Accessed on 11/28/2018).
- [33] *Tess: Artificial Intelligence Mental Health Support*. <https://www.x2ai.com/>. (Accessed on 02/27/2019).
- [34] *The Official YAML Web Site*. <http://yaml.org/>. (Accessed on 11/28/2018).
- [35] *Was ist LabVIEW? - National Instruments*. <http://www.ni.com/de-de/shop/labview.html>. (Accessed on 11/27/2018).
- [36] Joseph Weizenbaum. "ELIZA—A Computer Program For the Study of Natural Language Communication Between Man and Machine". In: (1966), pp. 1–7.
- [37] *Woebot - Your charming robot friend who is ready to listen, 24/7*. <https://woebot.io/>. (Accessed on 02/27/2019).
- [38] *Wysa - your 4 am friend and AI life coach*. <https://www.wysa.io/>. (Accessed on 02/27/2019).
- [39] *Zukunft der Consumer Technology – 2018*. <https://www.bitkom.org/sites/default/files/file/import/180822-CT-Studie-2018-online.pdf>. (Accessed on 03/06/2019).
- [40] *movisensXS - eXperience Sampling for Android!* <https://xs.movisens.com/>. (Accessed on 11/28/2018).
- [41] *squeakland : home of squeak etoys*. <http://www.squeakland.org/>. (Accessed on 11/28/2018).
- [42] *the MyExperience tool*. <http://myexperience.sourceforge.net/>. (Accessed on 11/28/2018).

