

**Gottfried Wilhelm
Leibniz Universität Hannover
Fakultät für Elektrotechnik und Informatik
Institut für Praktische Informatik
Fachgebiet Software Engineering**

Erhebung und Validierung von testbaren Anforderungen durch Visionvideos

**Elicitation and validation of testable Requirement through
Visionsvideo**

Masterarbeit

im Studiengang Informatik

von

Roger Arnold FOKAM PIAM

Prüfer: Prof. Dr. rer. nat. Kurt Schneider

Zweitprüfer: Prof. Dr. Michael Rohs

Betreuer: M. Sc. Jianwei Shi

Hannover, 06.03.2023

Erklärung der Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbständig und ohne fremde Hilfe verfasst und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel verwendet habe. Die Arbeit hat in gleicher oder ähnlicher Form noch keinem anderen Prüfungsamt vorgelegen.

Hannover, den 06.03.2023

Roger Arnold FOKAM PIAM

Kurzzusammenfassung

Zur Visualisierung der Anforderungen werden Visionsvideos durch den Anforderungsingenieur erstellt. Trotz dieser Visionsvideos ist es für Kunden oft schwierig zu verstehen, wie sie die Funktionen des Produkts nutzen können. Es ist ebenfalls schwierig für Tester, Tests aus diesen Visionsvideos abzuleiten. In dieser Arbeit wird das Ziel verfolgt, die in den Visionsvideos visualisierten Anforderungen zu erheben und im Falle von Unklarheiten zu interpretieren. Hierzu werden die Visionsvideos mit Hilfe einer einfachen Sprache, Gherkin, textuell beschrieben. Aus diesen Beschreibungen werden die Anforderungen abgeleitet und in vage und testbare Anforderungen unterteilt.

In Rahmen dieser Arbeit wurde eine Studie mit zwei Gruppen von potenziellen Nutzern der Produkte, die in den Visionsvideos visualisiert wurden, durchgeführt. Die erste Gruppe wurde gebeten, sich zunächst die Visionsvideos anzuschauen und anschließend die Gherkin-Texte zu lesen. Die zweite Gruppe hingegen durfte sich ausschließlich die Visionsvideos ansehen. Die Ergebnisse der Studie zeigen, dass es einen Unterschied zwischen der Anzahl der Anforderungen gibt, die von den Studienteilnehmern mit Hilfe von Gherkin-Texten formuliert wurden, im Vergleich zu denen, die ohne Gherkin-Texte formuliert wurden. Die Gherkin-Texte ermöglichten es den Kunden auch, unklare Anforderungen zu interpretieren.

Abstract

Elicitation and validation of testable Requirement through Vision Video

To visualize the requirements, vision videos are created by the requirements engineer. Despite these vision videos, it is often difficult for customers to understand how to use the product's features. It is also difficult for testers to derive tests from these vision videos. The aim of this work is to identify and interpret the requirements visualized in the vision videos, particularly in cases of ambiguity. For this purpose, the vision videos are textually described using a simple language called Gherkin. From these descriptions, requirements are derived and classified into vague and testable requirements.

In the context of this work, a study was conducted with two groups of potential users of the products visualized in the vision videos. The first group was asked to watch the vision videos first and then read the Gherkin texts. The second group, on the other hand, was only allowed to watch the vision videos. The results of the study show that there is a difference in the number of requirements formulated by the study participants using Gherkin texts compared to those formulated without Gherkin texts. The Gherkin texts also enabled customers to interpret ambiguous requirements.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Ziel der Arbeit	2
1.2	Lösungsansatz	2
1.3	Ergebnisse der Arbeit	3
1.4	Struktur der Arbeit	4
2	Grundlagen	5
2.1	Requirements Engineering	5
2.1.1	Anforderungsanalyse	7
2.1.2	Anforderungsmanagements	8
2.2	Visionsvideos	10
2.3	Erhebung von Anforderungen	11
2.3.1	Aufgaben bei der Erhebung	11
2.3.2	Techniken zur Erhebung von Anforderungen	11
2.4	Textuelle Beschreibung von Anforderungen	14
2.4.1	Vorteile von Gherkin	14
2.4.2	Nachteile von Gherkin	15
2.4.3	Syntax von Gherkin	15
2.5	Interpretation von Anforderungen	18
2.5.1	Nachfragen	18
2.5.2	Einholen von Expertenwissen	19
2.5.3	Anforderungsreviews	19
2.5.4	Klassifikation	20
3	Verwandte Arbeiten	21
3.1	Visionsvideos in Requirements Engineering	21
3.2	Videos in der Anforderungsanalyse	22
3.2.1	Anforderungserhebung	22
3.2.2	Anforderungsvalidierung	23
3.3	Abgrenzungen	23

4	Konzept und Umsetzung	25
4.1	Detaillierte Beschreibung der Prozesse zur Erhebung von Anforderungen	25
4.1.1	Schritt 1: Visionsvideo anschauen	26
4.1.2	Schritt 2: Visionsvideo segmentieren	26
4.1.3	Schritt 3: Feature und Szenarien ableiten	27
4.1.4	Schritt 4a: Klare Anforderungen auflisten	28
4.1.5	Schritt 4b: Unklarheiten markieren	30
4.1.6	Schritt 5: Unklare Anforderungen auflisten	31
4.2	Anforderungsgrad	31
4.3	Unterscheidung von Anforderungen	33
4.3.1	Testbare Anforderungen	33
4.3.2	Vage Anforderungen	34
4.4	Interpretation	34
5	Vorgehen bei der Benutzerstudie	35
5.1	Vorbereitung mit GQM	36
5.1.1	Ziele:	36
5.1.2	Forschungsfragen	37
5.1.3	Metriken	37
5.2	Kontext	38
5.3	Hypothesen	38
5.4	Variable	40
5.5	Probandenauswahl	40
5.6	Experiment Design	41
5.7	Instrumentierung	42
5.8	Methode und Ablauf	43
5.9	Threats to Validity	45
5.9.1	Konklusion Validität	45
5.9.2	Interne Validität	46
5.9.3	Konstruktvalidität	46
5.9.4	Externe Validität	46
6	Ergebnisse und Interpretation	47
6.1	Vorbereitung der Studie	47
6.1.1	Visionsvideoseauswahl	47
6.1.2	Demographische Daten	48
6.1.3	Gherkin-Schulung	49
6.2	Überprüfung der Hypothesen	51
6.2.1	$H_{0,2,2}$: Es gibt keinen Unterschied zwischen dem Betrachten von Visionsvideos mit Hilfe von Gherkin-Texten und dem Betrachten ausschließlich von Visionsvideos in Bezug auf die Variable Anzahl von Anforderungen.	51

6.2.2	$H_{0,1,1}$: Es gibt keine Anforderungen, die in den gezeigten Gherkin-Texten richtig beschrieben werden müssen.	56
6.2.3	$H_{0,2,1}$: Es gibt keine Anforderungen, die in den gezeigten Gherkin-Texten vervollständigt werden müssen.	59
7	Diskussionen	63
7.1	Antwort auf die Forschungsfragen	63
7.2	Formulierung von Anforderungen	64
7.2.1	Einfluss vom Betrachten von Gherkin-Texten	64
7.2.2	Einfluss von User story	65
7.2.3	Einfluss der Anzahl der Anforderung	65
7.3	Begrenzung der Arbeit	66
7.3.1	Requirement Engineering	66
7.3.2	Visionsvideo	66
7.3.3	Textuelle Beschreibung von Anforderungen	67
7.3.4	Benutzerstudie	67
8	Zusammenfassung und Ausblick	69
8.1	Zusammenfassung	69
8.2	Ausblick	69
A		71
A.1	Inhalt der CD	71
A.2	Kurze Beschreibung der Visionsvideos	71
A.2.1	„CVparser“	71
A.2.2	„Intelligent Coach“	71
A.2.3	„iKühlschrank“	72
A.2.4	„MobileMeetingMaster“	72
A.2.5	„Watertracer“	72
A.3	Umfrage	72
B	Feature-Dateien mit Szenarien pro Visionsvideos	75
B.1	iKühlschrank	75
B.2	Watertracer	78
B.3	CVParse	80
B.4	IntelligentCoach	82
B.5	MobileMeetingMaster	84
C	Erhobenen Anforderungen	87
D	Begleitdokumente der Studie	91
D.1	Information zur Studie Visionsvideos	91
D.2	Einverständniserklärung	92
D.3	Probandeninformation	93

D.4	Aufgabenstellung	94
D.5	Fragebogen	95
D.6	Gherkin-Schulung	95
E	Ergebnisse der Studie	99
E.1	Frage 2 für Gruppe ohne Gherkin-Texte	99
E.1.1	Visionsvideo IntelligentCoach	99
E.1.2	Visionsvideo CVParser	100
E.2	Frage 3 für Gruppe mit Gherkin-Texte	100
E.2.1	Visionsvideo CVParser	101
E.2.2	Visionsvideo IntelligentCoach	102

Kapitel 1

Einleitung

Requirement Engineering [31] ist ein systematisches Vorgehen, um Anforderungen an ein System, ein Produkt oder eine Software zu analysieren und zu verwalten. Zur Analyse von Anforderungen [65] gehören auch die Erhebung und die Validierung. Die Anforderungen werden mittels Methoden erhoben, dokumentiert, verifiziert und vom Kunden validiert. Anforderungsingenieure erstellen also Visionsvideos [39], um Anforderungen zu visualisieren. Diese Anforderungen werden danach von Kunden validiert. Ein Visionsvideo zeigt die Vision eines künftigen Produkts. Nachdem Visionsvideos erstellt wurden, kommt es vor, dass Anforderungen allerdings oft implizit dargestellt sind. Die Visionsvideos sind da, aber die Anforderungen sind nicht testbar und die Kunden können die Funktionalität der Software nicht benutzen. Deswegen sollen Anforderungen durch Visionsvideos erhoben, beschrieben und validiert werden.

Für die Erhebung gibt es Methoden wie Interviews, Workshop und Apprenticing [62]. Das Interview ist eine dialogische Gesprächsform. Bei einem Workshop debattiert eine Gruppe von Menschen ausführlich über ein spezifisches Thema und tauscht ihre Ansichten aus [59]. Nagel et al. [53] haben diese Methoden analysiert und das Problem war es, entweder ausreichende Zeit für Interviews zu finden oder einen Termin für Workshop zu finden, der allen Teilnehmern passt. Probleme bei Treffen mit vielen Teilnehmern ist es wahrscheinlich, dass einige Teilnehmer aktiver sind als andere. Außerdem kann es sein, dass die Teilnehmer den Inhalt des Videos nicht richtig verstehen. Eine weitere Diskussion, zumindest per E-Mail, ist erforderlich. Bei Apprenticing [62] erlernt der Anforderungsingenieur die empirischen Daten mittels Beobachtung zu erheben und sie selbstständig durchzuführen. Das Apprenticing wäre auch durch Anwendung von Visionsvideos kompliziert, da es keine zu beobachtenden Personen gibt, die reflektieren und erklären können, was sie tun.

Es gibt verschiedene Techniken, um die erhobenen Anforderungen zu beschreiben und zu validieren. Eine Möglichkeit sind verhaltensorientierte

Dokumentationstechniken, wie beispielsweise universell einsetzbare Modellierungssprachen. Diese Techniken können dazu beitragen, Software-Teile und andere Systeme spezifizieren, konstruieren, dokumentieren und visualisieren zu können. Jedoch werden die Anforderungen nicht textuell beschrieben. Sie müssen für die Validierung noch vereinfacht werden, weil die Kunden die Diagramme oft nicht verstehen. Außerdem können testbare Anforderungen nicht von vagen Anforderungen unterschieden werden. Deswegen ist eine natürliche domänenspezifische Sprache [52] (englisch domain-specific language, kurz DSL) empfohlen, um alle Probleme der Domäne und nicht, was außerhalb der Domäne liegt, darstellen zu können. Die meisten dieser Art Sprachen basieren auf dem Programmieren. Sie sind nicht für Tester geeignet und definieren keine Testfälle, um das Verhalten der Software zu verifizieren. Die zur Verfügung gestellten Visionsvideos und die textuelle Beschreibung der Anforderungen sind bereits Methoden zur Validierung [14]. Aus diesem Grund ist es notwendig, eine Methode für Validierung mit Visionsvideos und textuellen Beschreibungen zu konzipieren.

1.1 Ziel der Arbeit

Das Hauptziel dieser Arbeit ist unter anderem der Einsatz von Visionsvideos in einem agilen Kontext [31], um Anforderungen zu erheben und zu validieren. In dieser Arbeit soll herausgefunden werden, wie Anforderungen durch Visionsvideos mit und ohne textuelle Beschreibungen in einfacher Sprache besser von den Kunden geklärt und validiert werden können. In einer einfachen Erklärung wird das erste Ziel darin bestehen, ein Verfahren zu finden, das die Erhebung von Anforderungen ermöglicht, und dieses Verfahren anzuwenden. Das zweite Ziel dieser Arbeit ist, eine Vergleichsstudie über die Fähigkeit potenzieller Kunden, die Anforderungen in Visionsvideos mit und ohne Hilfe dieses Verfahrens zu verstehen und zu validieren, durchzuführen. Diese Ziele dienen dazu, die oben dargestellten Abgrenzungen zu vermeiden.

1.2 Lösungsansatz

Die Lösung für diese Aufgabe besteht darin, sich die zur Verfügung gestellten Visionsvideos genau anzusehen und die Anforderungen zu extrahieren. Dafür wird eine Methode zur textuellen Beschreibung von Anforderungen unter Verwendung einer domänenspezifischen Sprache [52] für die Beschreibung des Verhaltens angewendet. Diese Sprache wird es auch ermöglichen, zwischen testbaren und vagen Anforderungen zu unterscheiden. Die gefundenen Anforderungen müssen vom Kunden validiert werden. Eine Benutzerstudie, die den Mittelpunkt dieser Arbeit bildet, wird dann entworfen, durchgeführt und evaluiert. Dabei werden Visionsvideos und textuelle Beschreibungen

verwendet, um die Fähigkeit der Kunden zu vergleichen, die Anforderungen der Visionsvideos mit und ohne textuelle Beschreibungen zu validieren. Dieser Lösungsansatz ist besonders gut, weil die Kunden ab diesem Zeitpunkt die Anforderungen besser verstehen und validieren können. Dadurch wird eine klare Kommunikation über Voraussetzungen und Schritte der Funktionalitäten zwischen Anforderungsingenieur, Kunden, und Tester vereinfacht. Die Anforderungen können ebenso getestet werden

1.3 Ergebnisse der Arbeit

Das Hauptergebnis dieser Arbeit liegt in der Tatsache, dass Anforderungen durch textuelle Beschreibungen erhoben werden können und die Kunden Anforderungen durch Visionsvideos mithilfe einer Benutzerstudie validieren. Aus diesem Grund zeigt eine textuelle Beschreibung von Anforderungen die Art und Weise, wie mit Visionsvideos umgegangen werden kann. Die Evaluation der Benutzerstudie zeigt, dass die Kunden sich mit den Anforderungen auseinandergesetzt haben.

Anpassungen der Arbeit

In dieser Masterarbeit habe ich folgenden Anpassungen zur Ausschreibung mit Einwilligung vom ersten Prüfer und Betreuer

Nach etwa zwei Monaten Arbeit stellte ich fest, dass die erhobenen Anforderungen Unklarheiten enthielten und daher nicht validiert werden konnten. Diese Anforderungen müssen zuerst interpretiert werden. Nach Absprache mit dem Betreuer und dem ersten Prüfer beschloss ich, mich auf die Interpretationsphase zu beschränken. Aus diesem Grund wurden die folgenden Änderungen vorgenommen:

1. Der Titel der Masterarbeit wird sich ändern und lautet nun: Erhebung und Interpretation von testbaren Anforderungen durch Visionsvideos.
2. Die Benutzerstudie soll eine vergleichende Studie von zwei Gruppen von Probanden durchführen. Die erste Gruppe besteht aus den Visionsvideos und den textuellen Beschreibungen, die zweite Gruppe ausschließlich aus den Visionsvideos.

Diese Anpassungen werden ab dem nächsten Teil berücksichtigt, außer in Kapitel 3 der verwandten Arbeiten.

1.4 Struktur der Arbeit

Diese Arbeit ist folgendermaßen aufgebaut. In Kapitel 2 wird auf die Grundlagen der Requirement Engineering und Visionsvideos eingegangen. Es werden grundlegende Definitionen eingeführt und erklärt sowie Erhebung- und Validierungsmethoden vorgestellt. In Kapitel 3 wird auf thematisch verwandte Arbeiten im Bereich Visionsvideos in der Anforderungsanalyse eingegangen und wie sich diese Arbeit von anderer Literatur unterscheidet. Kapitel 4 befasst sich mit dem Konzept und Umsetzung. Hier werden die Anforderungen textuell beschrieben, erhoben und aufgelistet. Es wird auch vagen Anforderungen von testbaren Anforderungen unterschieden. Es folgt ein Vorgehen bei der Benutzerstudie (Entwurf, Durchführung und Evaluierung) in Kapitel 5. Anschließend beschäftigt sich das Kapitel 6 mit den Resultaten, der Interpretation, der Analyse der Studie und das Kapitel 7 mit den Diskussionen rund um die Studie. Das letzte Kapitel resümiert schließlich diese Arbeit und gibt einen Ausblick auf mögliche weitere Forschungsprojekte.

Kapitel 2

Grundlagen

Dieses Kapitel erläutert die Konzepte, Verfahren und technologischen Grundlagen, die zum Grundverständnis folgender Kapitel wichtig sind. Es wird zunächst die Frage beantwortet, wie Requirements Engineering definiert wird. Dazu werden die verschiedenen Tätigkeiten und Funktionsweisen des Requirements Engineering (RE) ausführlich beleuchtet. Darauf basierend werden Visionsvideos und ihre Anwendung im RE-Prozess erläutert. Schließlich werden verschiedene Methoden zur Ermittlung und Interpretation einer Anforderung betrachtet.

2.1 Requirements Engineering

Der Fokus dieser Arbeit liegt auf dem Requirements Engineering, welches als erster Meilenstein im Software-Engineering unmittelbar nach Projektstart beginnt. Um das Konzept des Requirements Engineering zu verstehen, ist es notwendig, sich einen allgemeinen Überblick zu verschaffen und es zu definieren. In dieser Arbeit wird die Definition des **International Requirements Engineering Board (IREB)** [33] angewendet und wie folgt beschrieben: [58]:

Das Requirements Engineering umfasst einen strukturierten und methodischen Ansatz zur Festlegung und Verwaltung von Anforderungen. Dabei werden folgende Ziele verfolgt:

- Das Verständnis relevanter Anforderungen durch die Stakeholder zu erlangen, Konsens über die Anforderungen zu erzielen, sie gemäß vorgegebenen Standards zu dokumentieren und systematisch zu verwalten.
- Die Wünsche und Bedürfnisse der Stakeholder zu verstehen, zu dokumentieren und die Anforderungen zu spezifizieren und zu verwalten, um das Risiko zu minimieren, dass das System nicht den Erwartungen der Stakeholder entspricht.

Auf Grundlage dieser Begriffsdefinition lassen sich im Requirements Engineering zwei verschiedene Teilbereiche voneinander unterscheiden. Der erste davon ist die Anforderungsanalyse, die sich mit der Anforderungsspezifikation befasst. Der zweite nennt sich Anforderungsmanagement und befasst sich mit dem Änderungsmanagement und Tracking von Anforderungen. Die genannten Zielsetzungen betreffen diese beiden unterschiedlichen Teilbereiche. Sie gewährleisten eine weitere Aufteilung dieser beiden Teilspektre auf Teilaufgaben, die die zugehörigen Zielsetzungen des Requirements Engineering erfüllen. Auch andere Arbeiten [31, 19] haben sich mit dieser Art der Unterteilung des Requirements Engineering beschäftigt.

Börger et al. [8] haben verschiedene Phasen mit den beiden Teilbereichen in einem Referenzmodell verknüpft. Abbildung 2.1 zeigt dieses Referenzmodell des Requirements Engineering, das in die Bereiche Anforderungsanalyse und Anforderungsmanagement sowie deren sieben Unterbereiche gegliedert ist. Dieser systematische Ansatz von Börger et al. steigert gleichzeitig die Produktqualität und die Entwicklerproduktivität. Auf diese Weise werden die Kosten für die (Software) Entwicklung und die Arbeit an ungenutzten Funktionen und Komponenten verringert [20]. Der RE-Prozess ist hier ein inkrementelles und iteratives Vorgehen, das nach dem initialen Vorgehen parallel zu den anderen Schritten des Entwicklungsprozesses ablaufen kann [54]. In den folgenden Abschnitten werden die beiden Bereiche des RE-Prozesses und die damit verbundenen Aktivitäten genauer betrachtet.

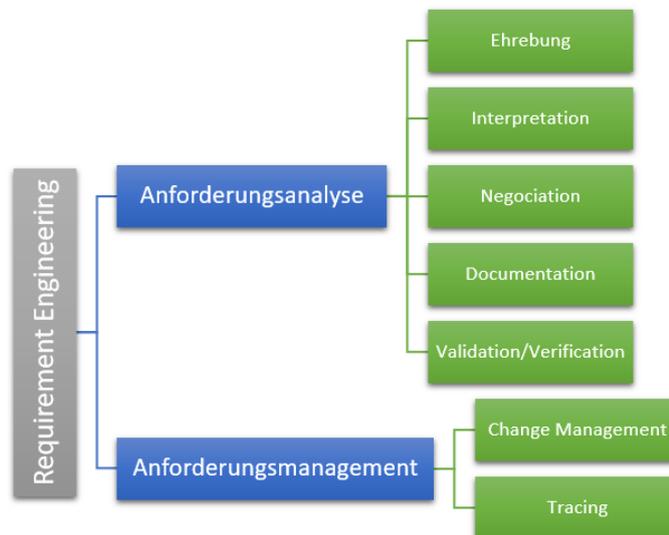


Abbildung 2.1: Das RE-Referenzmodell

2.1.1 Anforderungsanalyse

Um ein Projekt erfolgreich durchzuführen, befasst sich die Anforderungsanalyse (engl. Requirements Analysis) mit den Aktivitäten zur Ermittlung, Abstimmung, Formulierung, Dokumentation und Validierung von Anforderungen [62]. Die Kundenwünsche und -bedürfnisse zu kennen, stellt eine wesentliche Notwendigkeit dar, um daraus Anforderungen für alle relevanten Parteien abzuleiten. Gemäß dem Referenzmodell kann die Anforderungsanalyse in fünf Teilaufgaben unterteilt werden: Elicitation, Interpretation, Negotiation, Documentation und Validation/Verification [8, 19, 20, 54]. Die Teilprozesse der Anforderungsanalyse werden im Folgenden detaillierter erläutert. Diese Phasen treten im Allgemeinen zu Beginn des Entwicklungsprozesses auf.

Elicitation

Eine der bedeutendsten Aufgaben im Requirements Engineering ist die Elicitation, da sie dazu dient, die Anforderungen der Stakeholder zu identifizieren und die Systemgrenzen festzulegen. Diese Grenzen definieren den Kontext, in dem das System arbeiten wird. Das Hauptziel der Elicitation besteht darin, ein besseres Verständnis für die Anforderungen des Projekts zu erlangen. Die Identifizierung der Stakeholder ist ein wichtiger erster Schritt in diesem Prozess, da sie oft zu Beginn des Projekts noch nicht klar definiert sind. Die Festlegung der Stakeholder ermöglicht es, Rohanforderungen zu erfassen, welche abstrakte und unstrukturierte Anforderungen an das System darstellen. Es gibt verschiedene Methoden, um Rohanforderungen zu erfassen, wie beispielsweise Interviews, Workshops oder Fragebögen. Es ist auch wichtig, andere relevante Informationsquellen zu berücksichtigen, wie zum Beispiel die Umgebung des Systems und mögliche vorherige Systeme sowie vorhandene Dokumentation [62].

Interpretation

Im Rahmen der Interpretation werden die Rohanforderungen, die während der Elicitation erfasst wurden, genauer analysiert. Das Ziel der Interpretation ist es, die tatsächlichen Anforderungen zu bestimmen und sie in eine strukturierte Form zu bringen, beispielsweise durch die Klassifizierung nach Funktionalität. Es gibt zwei Arten von Anforderungen: funktionale und nicht-funktionale. Funktionale Anforderungen beschreiben, was das Produkt tun soll, während nicht-funktionale Anforderungen zusätzlich die Qualität der Funktionen des Produkts bestimmen, wie zum Beispiel die optimale Reaktionszeit des Systems. Die Beziehungen zwischen den Anforderungen werden erkannt und die Anforderungen werden konkretisiert.

Negotiation

In der Phase der Negotiation werden nicht nur Abhängigkeiten zwischen den Anforderungen identifiziert, sondern auch Inkonsistenzen aufgedeckt und behoben. Anschließend erfolgt eine Priorisierung der Anforderungen. Die verwendeten Methoden und Techniken ähneln denen, die bei der Elicitation eingesetzt werden. Das übergeordnete Ziel der Negotiation besteht darin, einen Kompromiss für eventuell auftretende Konflikte zu finden und sicherzustellen, dass das Ergebnis des Projekts im Einklang mit den Bedürfnissen und Meinungen aller beteiligten Parteien steht.

Dokumentation

Die Dokumentation dient dazu, die gefundenen Anforderungen gemäß den festgelegten Standards zu dokumentieren. Bei der Dokumentationsarbeit werden die Anforderungen schriftlich festgehalten. Zusätzlich müssen alle Entscheidungen und Begründungen dokumentiert werden. Dazu werden sie zunächst definiert, nach menschlichen Bedürfnissen gruppiert und schließlich durch relevante Merkmale definiert. Die Dokumentation sollte Anforderungen verbinden, sowie den Kontext und die Ziele enthalten. Neben den Anforderungen werden auch die Beteiligten, ihre Rollen und Ziele sowie der Kontext ihrer Anforderungen festgehalten. Die schriftliche Ergebnissicherung sollte für alle Personen verständlich formuliert sein, um spätere Missverständnisse zu vermeiden und neue Mitarbeiter besser einarbeiten zu können.

Validation/Verifikation

Diese letzte Phase der Anforderungsanalyse zielt auf eine möglichst hohe Qualität der daraus entwickelten Anforderungen ab, damit sie eine sichere Grundlage für die Softwareentwicklung bilden. Obwohl sowohl die Validierung als auch die Verifizierung diesem Ziel dienen, müssen sie voneinander unterschieden werden. Die Validierung, Content Check genannt, prüft, ob die schriftlichen Anforderungen den tatsächlichen Anforderungen entsprechen, zum Beispiel der Kundenperspektive. Dies geschieht durch Umfragen, Interviews und Stakeholder-Befragungen [75]. Auf der anderen Seite ist die Verifizierung eine systematische Überprüfung, ob die tatsächlichen Eigenschaften des Produkts den vorab festgelegten Anforderungen entsprechen, beispielsweise in Bezug auf Kompatibilität und Benutzerfreundlichkeit.

2.1.2 Anforderungsmanagements

Der zweite Bereich des Requirements Engineering ist das Anforderungsmanagement. Das Anforderungsmanagement dient der Anforderungsverwaltung und damit der allgemeinen Unterstützung der Anforderungsanalyse.

Nach Rupp et al. [62] lassen sich die Aufgaben des Anforderungsmanagements aus zwei Grundhypothesen ableiten. Die erste Hypothese lautet:

Anforderungen ändern sich im Laufe eines Projekts/Systemlebenszyklus

Das heißt, die Beteiligten können ihre Anforderungen an das Projekt zu jeder Zeit modifizieren oder revidieren. Die Änderungen scheinen dadurch motiviert zu sein, dass sich die Stakeholder das Endprodukt zu Beginn des Projekts ausschließlich schlecht vorstellen können und im Laufe der Entwicklung eine konkretere Vorstellung davon bekommen. Die zweite Hypothese lautet:

Die Anforderungen werden im Laufe des Entwicklungsprozesses wiederverwendet

Auf der Grundlage dieser Hypothesen und des im Börger et al. [8] beschriebenen Modells kann das Anforderungsmanagement in **Änderungsmanagement** und **Tracking** unterteilt werden, wie es wird. Die genauen Begriffe und Aufgaben dieser beiden Phasen werden im Folgenden näher erläutert.

Change Management

Das Änderungsmanagement beginnt erst, nachdem die Anforderungen analysiert wurden und versucht wurde, sie zu organisieren. Es ist ein Prozess, der den Lebenszyklus aller Änderungen während der gesamten Laufzeit des Projekts verwaltet, so dass sinnvolle Änderungen mit minimaler Unterbrechung der IT-Dienste möglich sind. In dieser Phase werden alle während des Projekts aufgetretenen Änderungswünsche dokumentiert und die jeweiligen Gründe dafür festgehalten [62].

Tracing

Eine wichtige Maßnahme im Anforderungsmanagement ist das Tracing. Beim Tracing geht es darum, Änderungen an Anforderungen und deren Beziehungen zu verfolgen. Damit soll sichergestellt werden, dass die Entwicklung von Anforderungen und deren Ursachen während des gesamten Projekts immer nachvollziehbar ist. Pohl und Rupp [58] definieren Traceability als

die Fähigkeit, eine Anforderung über ihren gesamten Lebenszyklus hinweg zu verfolgen

Das Tracing kann in Pre- und Post-Tracing unterteilt werden. Das Pre-Tracing ermöglicht die Rückverfolgung einer Anforderung bis zu ihrer Quelle, falls es Änderungen gibt. Post-Tracing hingegen kann die Auswirkungen von Anforderungen auf verwandte Systeme untersuchen.

2.2 Visionsvideos

In diesem Abschnitt werden die Grundlagen des Themas Visionsvideos vorgestellt. Der Begriff Visionsvideo umfasst laut Karras et al. [38] ein konzipiertes Video, das eine Vision oder Teilvisionen eines zukünftigen Systems oder einer Funktionalität eines noch nicht existierenden Systems repräsentiert. Es geht um die Erfahrung, das zukünftige System oder die zukünftige Funktionalität in einer realen Umgebung zu verwenden. In diesem Zusammenhang ist ein Szenario eine Abfolge von Ereignissen und Aktionen.

Die Entscheidung, Visionsvideos als Kommunikationskanal für zukünftige Systeme oder Funktionalitäten zu verwenden, basiert auf verschiedenen Faktoren, darunter die Feststellung von Ambler [3], dass schriftliche Dokumentation die am wenigsten effektive Dokumentationsoption ist. Abbildung 2.2 zeigt einen Vergleich der Effektivität verschiedener Kommunikationskanäle. Dabei geht Ambler [3] davon aus, dass wichtige Informationen durch schriftliche Dokumentation verloren gehen können. Die Kommuni-

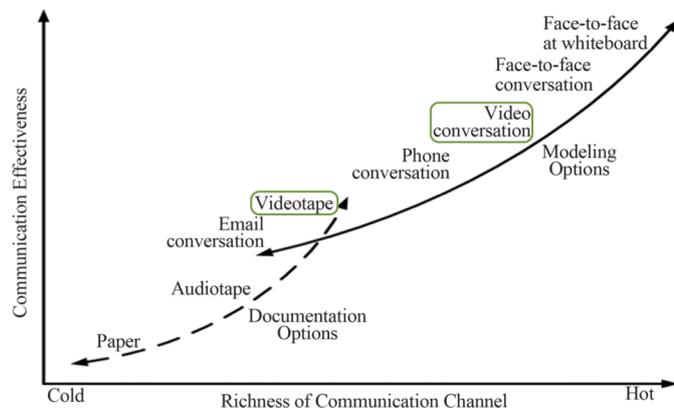


Abbildung 2.2: Darstellung der Kommunikationsarten nach [3]

kationskanäle werden in zwei Kategorien unterteilt: die Dokumentation und die Modellierung. Die Grafik zeigt, dass Video als Dokumentationsoption für die Kommunikation ein optimales Medium ist. Ebenso ist das Video-Gespräch ein effektiver Weg für die Modellierung. Diese beiden Optionen werden durch die grüne Farbmarkierung in der Abbildung besonders hervorgehoben. Zusammen repräsentieren diese Optionen den hohen Wert der Visionsvideo in Hinblick auf Vielseitigkeit und Leistungsfähigkeit.

Dennoch muss berücksichtigt werden, dass der Prozess der Erstellung und der Umgang mit Videos im Vergleich zur entsprechenden textbasierten Darstellungsform viel aufwendiger ist. Trotz der großen Vielseitigkeit und der Leistungsfähigkeit dieses Mediums existieren ebenfalls eine Reihe von negativen Aspekten, welche die Verwendung von Visionsvideos in der Praxis beeinträchtigen. Vor allem der hohe Zeitaufwand für die Produktion von Visionsvideos und die sich ständig ändernden Anforderungen sind hier zu nennen. Auch das Fehlen von Grundsätzen und Leitlinien für die Darstellung einer Produktvision in Visionsvideos macht es schwierig, aussagekräftige Videos zu erstellen. Es besteht daher häufig die Gefahr, dass nicht relevante Videoszenen erstellt werden [10].

2.3 Erhebung von Anforderungen

2.3.1 Aufgaben bei der Erhebung

Um die Anforderungen zu erheben, ist es wichtig, dass bestimmte Aufgaben erledigt werden. Die verschiedenen Aufgaben der Elicitation sind :

- **Stakeholder identifizieren** Die Stakeholder sind Personen oder Rollen (auch einer Gruppe), die von geplanten System betroffen sind und Einfluss auf Anforderungen haben sollten. Sie müssen dann identifiziert und dokumentiert werden. Sie können Tester, Betriebsinhaber, Programmierer, Benutzer, Softwarearchitekten oder Auftraggeber sein [56]...
- **Andere Quellen von Anforderungen identifizieren** Neben der Identifizierung der Stakeholder, die bereits eine der Quellen für Anforderungen sind, müssen andere Quellen der Anforderungen identifiziert werden, wie zum Beispiel Dokumente, eingeführte Abläufe und Software. Relevante Dokumente könnten Gesetze, Regelungen Visionspapiere, Entscheidungsvorlagen oder Vorarbeiten anderer Firmen sein. Zur Software zählen Altsysteme, Hardware und Prototypen.
- **Erfassen von Rohanforderungen** Aus diesen zwei oben genannten Quellen müssen dann die Rohanforderungen herausgezogen werden. Einige Techniken dafür werden in den nächsten Abschnitten detailliert.

2.3.2 Techniken zur Erhebung von Anforderungen

Faktoren wie das Kundenprofil und die Organisationsstruktur sowie die Projektart müssen berücksichtigt werden, bevor eine Technik zur Anforderungserhebung oder eine Kombination von Techniken angewendet wird. Es gibt zahlreiche Anforderungsermittlungstechniken, die es ermöglichen, wichtige Informationen von fachspezifischen Experten und Stakeholdern

zu erhalten. Die beliebtesten Techniken im Jahre 2023 sind im Folgenden aufgelistet [68]

Brainstorming

Es handelt sich um eine Gruppentechnik, die aus Repräsentanten aller Beteiligten besteht und von Business-Analysten eingesetzt wird, um viele neue Ideen zu generieren. Sie ist eine Plattform für den Austausch von Meinungen. Ein hoch qualifizierter Moderator ist erforderlich, um mit den Vorurteilen und Konflikten in der Gruppe umzugehen. Die Vorschläge aus den Brainstorming-Sitzungen müssen ordnungsgemäß dokumentiert werden, damit der Aktionsplan erstellt werden kann und jeder ihn sehen kann. Zuletzt wird ein Dokument vorbereitet, das eine Liste der Anforderungen und, falls möglich, deren Priorität enthält.

Rewiew/Analyse der Dokumente

In dieser Phase des Prozesses der Anforderungserhebung prüfen die Business Analysten die vorhandene Dokumentation, die ihnen zur Verfügung steht, mit dem Ziel, den Änderungs- oder Verbesserungsbedarf zu ermitteln. Die Dokumentenanalyse wird durchgeführt, bevor ausführlichere Bedarfserhebungssitzungen oder Interviews mit Stakeholdern vereinbart werden. Diese Analyse ist relevant, um die Umsetzung der aktuellen Lösungen zu validieren, und hilft auch dabei, die Bedürfnisse des Unternehmens zu verstehen. Sie ist auch von Vorteil, wenn der Plan darin besteht, ein bestehendes System zu aktualisieren, oder bei Migrationsprojekten.

Fokusgruppen

Fokusgruppen sind eine Möglichkeit, Produkt- oder Serviceinformationen von einer Gruppe zu sammeln. Die Fokusgruppe besteht aus Fachexperten, die Feedback geben, um Prozesse, Ideen oder Lösungen zu verfeinern, die sich aus früheren Ermittlungsaktivitäten wie Brainstorming und Dokumentenanalyse ergeben haben. Ziel der Gruppe ist es, das Thema zu diskutieren und Informationen bereitzustellen. Ein Moderator leitet diese Sitzung. Er sollte mit Business-Analysten zusammenarbeiten, um die Ergebnisse zu analysieren und den Stakeholdern entsprechende Ergebnisse zu liefern. Wenn sich ein Produkt in der Entwicklung befindet und die Diskussion sich auf dieses Produkt beziehen soll, wird das Ergebnis eine Aktualisierung der bestehenden Anforderungen oder eine Liste neuer Anforderungen sein. Wenn ein Produkt marktreif ist, wird sich die Diskussion auf die Herausgabe des Produkts beziehen [30].

Interface-Analyse

Die Interface-Analyse bezieht sich auf die Erforschung von Interfaces, die externen und internen Systemen oder Komponenten dabei helfen, miteinander und mit den Endbenutzern zu interagieren. Die Interface-Analyse hilft ihnen dabei, Regeln für die Verwaltung, mögliche Herausforderungen und fehlende oder übermäßige Funktionen zu identifizieren, die mit den Stakeholdern diskutiert werden müssen.

Interviews

Dies ist die am häufigsten verwendete Technik zur Erhebung von Anforderungen. Interviews sind ein hervorragendes Mittel, um wesentliche Daten zu extrahieren, indem den Beteiligten Fragen gestellt werden. Interviewtechniken sollten eingesetzt werden, um starke Beziehungen zwischen Business-Analysten und Beteiligten aufzubauen. Wenn der Interviewer eine Reihe von vordefinierten Fragen hat, handelt es sich um ein strukturiertes Interview. Wenn er kein bestimmtes Format oder keine spezifischen Fragen hat, handelt es sich um ein unstrukturiertes Interview [5].

Beobachtung

Beobachtung ist eine hervorragende Ermittlungstechnik, die es ermöglicht, Anforderungen auf der Grundlage von Bemerkungen zu verstehen, die mit den Prozessabläufen und Arbeitsumgebungen der beteiligten Parteien zusammenhängen. Der Beobachtungsplan stellt sicher, dass sich alle Beteiligten über den Zweck der Beobachtungssitzung im Klaren sind, dass sie sich über die erwarteten Ergebnisse einig sind und dass die Sitzung ihre Erwartungen erfüllt. Die Beobachtung kann aktiv oder passiv sein. Aktives Beobachten besteht darin, Fragen zu stellen und zu versuchen, die Arbeit anderer Personen zu erledigen. Beim passiven Beobachten verfolgen die Beobachter still die Arbeitsabläufe.

Prototyping

Das Prototyping, eine der wichtigsten Phasen des Prozesses der Anforderungsermittlung, ermöglicht es Eigentümern und Endnutzern, realistische Modelle von Anwendungen vor ihrer endgültigen Entwicklung zu betrachten. Prototyping wird eingesetzt, um fehlende oder nicht spezifizierte Anforderungen zu identifizieren. Bei dieser Technik werden dem Kunden häufig Demonstrationen vorgeführt, indem Prototypen erstellt werden, damit sich der Kunde ein Bild davon machen kann, wie das Produkt aussehen wird.

Befragungstechnik

Wenn mehrere Fachexperten und mehrere betroffene Parteien an einem Projekt beteiligt sind, führen Business Analysts eine Umfrage durch, um die Anforderungen zu erheben. Alle Beteiligten erhalten einen Fragebogen, den sie ausfüllen müssen. Anschließend werden die Antworten analysiert, um die Anforderungen zu verfeinern. Die Fragen müssen auf Risiken mit hoher Priorität basieren und sollten direkt und unmissverständlich formuliert sein. Sobald die Umfrage beendet ist, müssen die Teilnehmer darüber informiert und an die Teilnahme erinnert werden. Umfragen sind kostengünstiger als andere Techniken zur Bedarfsermittlung, einfach zu administrieren und können sowohl qualitative als auch quantitative Ergebnisse liefern.

2.4 Textuelle Beschreibung von Anforderungen

Wenn die Anforderungen auf Tools gestellt werden, die für ihre Validierung bestimmt sind, wie zum Beispiel Visionsvideos, ist es möglich, die Anforderungen in textueller Form zu beschreiben. Diese Anforderungen stammen also aus den Visionsvideos und es würde genügen, zum Beispiel das Verhalten der im Visionsvideos dargestellten Anwendung zu beschreiben. Für diese Fälle können bestimmte Methoden verwendet werden. Das Behavior-Driven-Development-Werkzeug zählt dazu [28, 70]. Behavior-Driven-Development, abgekürzt BDD [70, 28], ist eine Methode innerhalb agiler Softwareentwicklungsmodelle, die sich auf die Entwicklung von Software basierend auf ihrem Verhalten konzentriert [32]. Bei der Aufgaben-Anforderungsanalyse werden die Softwareziele und -ergebnisse als spezielle Dokumente gespeichert. Die Anforderungen an eine Software werden mittels Funktionalitäten beschrieben. In der Regel erfolgt diese Beschreibung in einem bestimmten Format. Eines der gängigen Formate beim Behavior Driven Development ist die bekannte Beschreibungssprache Gherkin [18, 78]. Gherkin ist eine einfach strukturierte Klartextsprache. Sie ist so konzipiert, dass sie auch von Nicht-Programmierern leicht erlernt werden kann. Gherkin hilft bei der Beschreibung des Geschäftsverhaltens, ohne in Details der Implementierung zu gehen, und verwendet einfache Sprache zur Beschreibung von Features. Gherkin hat die folgenden Vor- und Nachteile [27, 29, 67]:

2.4.1 Vorteile von Gherkin

- Die Gherkin-Sprache ist eine einfache Sprache und konzentriert sich auf die Anforderungen des Projekts.
- Die Anforderungen können später auch als Tests geschrieben werden.

- Gherkin ist auch für Nicht-Programmierer einfach genug zu verstehen. Kein Fachwissen wird benötigt, um den kleinen Gherkin-Befehlssatz zu verstehen.
- Es macht die beschriebenen Features einfacher zu begreifen

2.4.2 Nachteile von Gherkin

- Es erfordert ein hohes Maß an Engagement und Zusammenarbeit zwischen den Beteiligten. Dies bedeutet, dass die Teammitglieder (Entwickler, Tester, Anforderungsingenieur und Kunde) zu jeder Zeit gemeinsam an dem Projekt arbeiten müssen, was nicht oft möglich ist.
- Gherkin ist nicht für alle Projekte die beste Wahl. Kurze Projekte können länger dauern, wenn diese Methode verwendet wird.

2.4.3 Syntax von Gherkin

Wie YAML [69] oder Python [47] ist Gherkin [17] eine zeilenorientierte Sprache, die Einrückungen zur Strukturierung verwendet. Ein Gherkin-Dokument hat die Erweiterung *.feature* und die Beschreibung mit Gherkin wird normalerweise in sogenannten *Feature Files* abgelegt. Zeilenenden schließen Anweisungen (Schritte genannt) ab, und für die Einrückung können entweder Leerzeichen oder Tabulatoren verwendet werden (aus Gründen der Portabilität werden Leerzeichen verwendet). Schließlich beginnen die meisten Zeilen in Gherkin mit speziellen Schlüsselworten. Der Parser unterteilt die Eingabe in Features, Szenarien und Schritte. Zum besseren Verständnis muss nur das folgende Beispiel [17] betrachtet werden:

```

1 Feature: A terse but descriptive text of what is desired.
2   In order to achieve a named work value
3   As an express actor in the system
4   I want a beneficial outcome that supports the goal.
5
6 Scenario: A definable Business Case
7   Given a specific prerequisite
8     And a further prerequisite
9   When an action of an actor
10    And some other action
11    And yet another action
12   Then a testable result is achieved
13    And some other thing that we can verify also happens

```

Die Tabelle 2.1 beschreibt eine Liste der häufigsten Schlüsselwörter in Gherkin-Syntax.

PRIMÄRE	SEKUNDÄRE
Feature	Doc Strings
Scenario	Data Tables
Scenario Outline Examples	Tags
Step definition	Comments

Tabelle 2.1: Wichtige Schlüsselwörter [78]

Feature

Jede Feature-Datei beginnt mit dem Feature-Stoppwort und dem Namen der Funktionalität. Er ist einzigartig und bietet eine allgemeine Beschreibung eines Software-Features, um verwandte Szenarien zu gruppieren. Hinter dem Begriff «Feature: Featurename» gibt es eine user story [71]. Eine User Story [71] ist eine kleine, in sich geschlossene Einheit der Entwicklungsarbeit, die ein bestimmtes Ziel innerhalb eines Produkts erreichen soll. Der folgende Abschnitt startet die Features (Zeile 1), gibt ihnen einen Titel und startet auch die User Stories (Zeile 2 bis 4), die die Features beschreiben.

```

1 Feature: A terse but descriptive text of what is desired.
2   In order to achieve a named work value
3   As an express actor in the system
4   I want a beneficial outcome that supports the goal.
```

Scenario

Ein einzelnes Gherkin-Szenario ist ein Fluss von Ereignissen durch das beschriebene Feature. Es können mehrere Szenarien in einer Feature-Datei enthalten sein. Jedes Szenario beginnt mit **Scenario:** gefolgt vom Szenariennamen. Der folgende Abschnitt startet das Szenario und enthält eine Beschreibung des Szenarios. Ab der zweiten Zeile sind es die Schritte des Szenarios, von denen jeder einem definierten regulären Begriff entspricht.

```

1 Scenario: A definable Business Case
2   Given a specific prerequisite
3     And a further prerequisite
4   When an action of an actor
5     And some other action
6     And yet another action
7   Then a testable result is achieved
8     And some other thing that we can verify also happens
```

jedes Szenario besteht aus drei Teilen: **Voraussetzung, Aktion und Ergebnis**. Diese drei Teile werden durch die Gherkin-Schlüsselworte Given, When und Then markiert.

- Die Voraussetzung **Given** (Zeile 2) beschreibt den Zustand, in dem sich die Software zu Beginn des Szenarios befindet.

- Die Aktion **When** (Zeile 4) beschreibt eine Aktion oder ein Ereignis. Dabei kann es sich um eine Person handeln, die mit dem System interagiert, oder um ein Ereignis, das von einem anderen System ausgelöst wird.
- Das Ergebnis **Then** (Zeile 7) beschreibt, wie das Programm auf die Aktion reagiert.

Die Schlüsselwörter **But** und **And** können verwendet werden, um das vorherige Schlüsselwort entsprechend zu ergänzen und zu wiederholen.

Scenario outline

Das Kopieren und Einfügen von Szenarien zur Verwendung unterschiedlicher Werte wird schnell mühsam und repetitiv. Hier ist ein Beispiel für zwei gleiche Szenarien mit unterschiedlichen Werten:

```

1 Scenario: Eat 4 out of 11
2   Given there are 11 cucumbers
3   When I eat 4 cucumbers
4   Then I should have 7 cucumbers
5
6 Scenario: Eat 6 out of 22
7   Given there are 22 cucumbers
8   When I eat 6 cucumbers
9   Then I should have 16 cucumbers

```

Das Schlüsselwort Scenario Outline kann verwendet werden, um dasselbe Szenario mehrmals mit unterschiedlichen Wertekombinationen auszuführen. Diese beiden ähnlichen Szenarien können zusammenfassend in einem **Scenario Outline** dargestellt werden

Scenario Outlines ermöglichen es, diese Beispiele durch die Verwendung einer Vorlage mit < >-begrenzten Parametern prägnanter auszudrücken:

```

1 Scenario Outline: Eating
2   Given there are <begin> cucumbers
3   When I eat <food> cucumbers
4   Then I should have <lef> cucumbers
5
6   Examples:
7     | begin | food | lef |
8     | 11   | 4   | 7   |
9     | 22   | 6   | 16  |

```

Eine *Scenario Outline* muss ein oder mehrere *Examples* section(s) enthalten. Die Schritte werden als Vorlage interpretiert, die nie direkt ausgeführt wird. Stattdessen wird die *Scenario Outline* einmal für jede Zeile im darunter liegenden Abschnitt *Examples* ausgeführt (die erste Kopfzeile nicht mitgezählt). Die Schritte können <>-begrenzte Parameter verwenden, die auf Kopfzeilen in der Beispieltabelle verweisen. Gherkin ersetzt diese

Parameter durch Werte aus der Tabelle, bevor es versucht, den Schritt mit einer Schrittdefinition abzugleichen.

Kommentare sind nur am Anfang einer neuen Zeile an beliebiger Stelle in der Featuredatei erlaubt. Sie beginnen mit Null oder mehreren Leerzeichen, gefolgt von einem Rautenzeichen «# »und etwas Text. Das nachfolgende Beispiel veranschaulicht die Verwendung der Gherkin Sprache zur Beschreibung des Verhaltens eines Features. Blockkommentare werden derzeit von Gherkin nicht unterstützt. Jedes Schlüsselwort wird in 37 Sprachen übersetzt, aber die Standardsprache ist Englisch. Es ist möglich, die Sprache mit Hilfe der Überschrift in der ersten Zeile der Feature-Datei zu ändern zum Beispiel für Französisch.

```
1 #language : fr
```

2.5 Interpretation von Anforderungen

Die Interpretation von Anforderungen im Requirement Engineering bezieht sich auf den Prozess der Analyse und des Verständnisses von Kundenanforderungen und Geschäftsprozessen, um sicherzustellen, dass diese klar und umfassend verstanden werden. Dies umfasst die Identifizierung von Bedürfnissen und Erwartungen der Benutzer, die Beschreibung von Anforderungen in einer präzisen Sprache sowie die Festlegung von Grenzen und Einschränkungen für die Umsetzung. Ziel ist es, sicherzustellen, dass die Anforderungen vollständig, korrekt und konsistent sind, um eine erfolgreiche Softwareentwicklung zu gewährleisten und mögliche Missverständnisse oder Unklarheiten im Vorfeld zu beseitigen. Die Interpretation von Anforderungen ist ein entscheidender Schritt im gesamten Prozess des Requirement Engineerings und bildet die Grundlage für die erfolgreiche Umsetzung von Softwareprojekten [74, 58].

Es gibt verschiedene Techniken, die im Requirement Engineering angewendet werden können, um die Interpretation von Anforderungen zu verbessern. Einige der häufig verwendeten Techniken werden im Folgenden diskutiert [2].

2.5.1 Nachfragen

Es handelt sich dabei um eine sehr wichtige Technik zur Interpretation von Anforderungen. Bei dieser Technik geht es darum, bei Unklarheiten oder Fragen zu den Anforderungen direkt beim Kunden oder den Stakeholdern nachzufragen, um Missverständnisse zu vermeiden. Diese Technik erfordert eine offene und klare Kommunikation zwischen dem Requirement Engineer oder dem Entwickler und dem Kunden oder Stakeholder. Dabei sollten klare und präzise Fragen gestellt werden, um eine genaue Antwort zu erhalten. Offene Fragen können ebenfalls gestellt werden, um mögliche

Lücken oder Verständnisprobleme aufzudecken. Die Technik des Nachfragens kann ebenfalls dazu beitragen, eine bessere Beziehung zwischen dem Requirement Engineer oder dem Entwickler und dem Kunden oder Stakeholder aufzubauen, da eine offene und transparente Kommunikation gepflegt wird. Dies kann zu einem besseren Verständnis der Anforderungen führen und letztendlich dazu beitragen, dass das System oder die Anwendung den Bedürfnissen des Kunden oder Stakeholders entspricht [63].

2.5.2 Einholen von Expertenwissen

Diese Technik kann besonders nützlich sein, wenn der Requirement Engineer oder der Entwickler spezielles Fachwissen benötigt, um die Anforderungen zu interpretieren oder zu verstehen. Die Einholung von Expertenwissen kann durch verschiedene Methoden erfolgen, zum Beispiel durch Interviews mit Fachexperten, Workshops oder Diskussionen mit erfahrenen Entwicklern oder durch die Zusammenarbeit mit einem Berater oder einem Team von Experten auf dem betreffenden Gebiet. Darüber hinaus kann die Zusammenarbeit mit Experten dazu beitragen, das Verständnis des Requirements Engineers oder des Entwicklers für die Anforderungen zu vertiefen und auch dazu beitragen, mögliche Lösungen oder alternative Umsetzungsansätze zu identifizieren. Das Einholen von Expertenwissen sollte jedoch sorgfältig geplant und durchgeführt werden. Es ist wichtig, dass die Experten, die konsultiert werden, tatsächlich über das benötigte Fachwissen verfügen und dass sie in der Lage sind, ihre Erkenntnisse und Empfehlungen in einer klaren und verständlichen Art und Weise zu kommunizieren. Darüber hinaus sollte darauf geachtet werden, dass die Ergebnisse der Zusammenarbeit mit Experten in die gesamte Anforderungsanalyse integriert werden, um ein vollständiges und konsistentes Verständnis der Anforderungen zu gewährleisten [7].

2.5.3 Anforderungsreviews

Bei der Review von Entwicklungsdokumenten liest Kunde Spezifikation, gibt Feedback und korrigiert. Die Grundidee ist, dass ein Team aus Kunden des Systems, denjenigen, die mit dem Kunden interagieren, um die Anforderungen zu sammeln, und den Systementwicklern beginnt, die Anforderungen im Dokument zu lesen und sie im Detail auf Fehler, Inkonsistenzen, Konflikte und Unklarheiten zu kontrollieren. Anschließend verhandeln mit dem Kunden darüber, wie die gefundenen Probleme und Fehler behoben werden können. Eine Checkliste bestehend aus diversen Normen wird vorbereitet und die Prüfenden markieren die Checkboxen, damit es zu einer formalen Prüfung kommt. Anschließend wird eine abschließende Genehmigung erteilt. Als wesentliche Review-Techniken werden die Stellungnahme [15], die Inspektion [4] und der Walkthrough [45] bezeichnet.

2.5.4 Klassifikation

Die Technik der Klassifikation ist eine Methode zur Interpretation von Anforderungen, die darauf abzielt, die Anforderungen in verschiedene Kategorien oder Gruppen zu unterteilen. Diese Kategorisierung kann dabei helfen, die Anforderungen besser zu verstehen und zu organisieren, um sicherzustellen, dass alle wichtigen Aspekte abgedeckt sind. Ein Beispiel für eine mögliche Kategorisierung könnte sein, Anforderungen nach ihrer Art zu gruppieren, wie z.B. funktionale Anforderungen, nicht-funktionale Anforderungen, technische Anforderungen und so weiter. Eine andere Möglichkeit ist die Kategorisierung nach der Priorität, wobei die Anforderungen in *muss erfüllt werden*, *sollte erfüllt werden*, *konnte erfüllt werden* und *würde schön sein, wenn erfüllt* unterteilt werden können. Ein wichtiger Aspekt der Technik der Klassifikation ist, dass sie es ermöglicht, die Anforderungen in kleinere, besser handhabbare Teile zu zerlegen. Dadurch können die Anforderungen detaillierter und genauer definiert werden, was dazu beitragen kann, Missverständnisse oder Unklarheiten zu vermeiden [55, 43].

Es ist wichtig zu beachten, dass die Techniken zur Interpretation von Anforderungen je nach Projekt und Situation variieren können. Diese Techniken werden in Kombination mit den Techniken zur Erhebung von Anforderungen eingesetzt, um eine vollständige Interpretation der Anforderungen zu gewährleisten und sicherzustellen, dass das Endprodukt den Bedürfnissen und Erwartungen der Benutzer entspricht. Der Kontext des Projekts und die verfügbaren Ressourcen sollten bei der Auswahl der geeigneten Techniken berücksichtigt werden.

Kapitel 3

Verwandte Arbeiten

Dieses Kapitel stellt zunächst Forschungsarbeiten zu Visionsvideos in Requirement Engineering insbesondere in der Requirement Analyse vor. Anschließend wird erläutert, wie die verwandten Arbeiten abgegrenzt werden können.

3.1 Visionsvideos in Requirements Engineering

Viele Arbeiten haben sich bis heute mit dem Einsatz von Visionsvideos in Requirements Engineering(RE) befasst. Damit Visionsvideos in RE verwendet werden können, müssen sie erst produziert werden. Karras et al. [38] definierten einen interdisziplinären Leitfaden für die Produktion von Visionsvideos durch Softwareexperten. Sie [38] haben sich mit dem Mangel an Wissen und Fähigkeiten von Softwareprofis beschäftigt und zwei Richtlinien bereitgestellt, die als Checklisten zur Vermeidung häufiger Fehler während der Herstellung und Nutzung von Videos verwendet werden können.

Karras et al. [42] erwähnten in einer anderen Arbeit die Existenz einer inhaltlichen Verknüpfung von Anforderungen und Videos und versuchten, diese Verknüpfung zu verbessern. Karras [36] betrachtet außerdem ein Video als Medium im Bereich RE. Er [36] gibt Auskünfte über Stärken, Schwächen, Möglichkeiten und Gefahren einer Einstellung zu Videos als Medium im RE. In dieser Richtung schlugen Brill et al. [10] die Anwendung von Ad-hoc-Videos als konkrete Darstellung der ersten Anforderungen vor. Das Ziel bestand darin, schnell und einfach ein gegenseitiges Verständnis zwischen den Kunden und dem Anforderungsingenieur zu erreichen. Um dieses gegenseitige Verständnis zu fördern, haben Schneider et al. [64] Visionsvideos ausdrücklich eingeführt, um Feedback zu Produktvisionen und Softwareanforderungen einzuholen.

Im Rahmen dieser Arbeit werden Anforderungen durch Visionsvideos erhoben und validiert.

3.2 Videos in der Anforderungsanalyse

Anhand der Definition des Requirement Engineerings [48] laut des International Requirements Engineering Boards (IREB)[58] lassen sich die Anforderungsanalyse und das Anforderungsmanagement wie in Abschnitt 2.1 erwähnt identifizieren. Diese Arbeit fokussiert sich auf die Anforderungsanalyse, insbesondere auf die Anforderungserhebung und -validierung.

3.2.1 Anforderungserhebung

Fricker et al. [23] haben Workshop-Videos verwendet, um Anforderungen der Stakeholder an die Entwickler zu kommunizieren. Sie [23] haben diese Videos zur Erhebung von Anforderungen erstellt und genutzt. Die verschiedenen Methoden für die Erhebung von Anforderungen sind in Kapitel 2 dargestellt.

Einige von diesen Methoden wurden bereits diskutiert. Creighton et al. [16] haben Software Cinema für die Beschreibung von Szenarien durch Analyse von Videos entwickelt, indem Use Cases und UML Diagramme generiert werden können, die für die weitere Dokumentation verwendet werden können. Kiesling et al. [44] entwickelten das ReqVidA-Tool, um Requirements Engineers beim Sammeln von Anforderungen in einem Workshop zu unterstützen. Sie [44] versuchten eine Balance zwischen Mehraufwand und Nutzen zu finden, Videos mit möglichst geringem Aufwand bei größtmöglichem Nutzen der RE-Phase zu erstellen. Nagel et al. [53] haben Konzepte für synchrone und asynchrone Betrachtungen vorgestellt und die beiden Methoden in einem Experiment getestet. Die Ergebnisse haben gezeigt, dass die asynchrone Betrachtung die Nutzung von Visionsvideos für eine Vielzahl von Interessengruppen erleichtert hat. Der Ansatz ermöglicht es einer großen Anzahl von Stakeholdern, Ideen zu generieren und Bedenken zu dargestellten Visionen zu äußern.

Karras et al. [37] schlugen eine Kombination von Textprotokollen und Videos mit einem Software-Tool vor, um schriftliche Notizen mit dem entsprechenden Teil des Videos zu verbinden. Dies ermöglicht eine einfachere Erhebung von qualitativ hochwertigen Anforderungen. Horschka et al. [31] beschrieben den Umgang mit funktionalen Anforderungen und die Akzeptanzkriterien etwas formaler mit Hilfe der Gherkin-Syntax [78]. Sie [31] definierten die Gherkin-Sprache als

eine für Menschen lesbare, domänenspezifische Sprache, die speziell für die Beschreibung von erwarteten Verhaltensweisen von Produkten entwickelt wurde.

Sie [31] gaben dazu einen kurzen Überblick über die Funktionsweise und Struktur der Gherkin-Syntax, die ebenfalls zur Beschreibung der Anforderungen in dieser Arbeit verwendet wird.

3.2.2 Anforderungsvalidierung

Es gibt es eine Vielzahl von Forschungsarbeiten zur Validierung von Anforderungen. Jedoch existieren keine Veröffentlichungen zu Methoden, die auch textuelle Beschreibungen von Anforderungen mit Visionsvideos verbinden und ausführen. Robertson et al. [61] stellten einen industrieprobten Prozess zur Erfassung und Überprüfung von Anforderungen mit Blick auf die heutigen agilen Entwicklungsumgebungen dar. In dieser vollständigen aktualisierten Version des Bestsellers, zeigten Robertson et al. [61], wie sie herausfinden konnten, was der Kunde wollte und brauchte. Die Anforderungen werden entsprechend dem Agilitätsgrad des Projekts bearbeitet. Rupp et al. [62] stellten einzelne Prüftechniken genauer vor, um die Qualität und die Validierung von Anforderungen zu bewerten.

Brill et al. [10] vertreten den Standpunkt, dass Visionsvideos im Gegensatz zu traditionellen RE-Methoden dabei helfen konnten, schnell genug konkrete Ergebnisse für die Validierung zu liefern. Sie [10] haben eine Studie über die Verwendung von Videos im Vergleich zu Use Cases durchgeführt. Dabei zeigten Sie [10], dass Ad-hoc-Videos mit Use Cases zu vergleichen oder besser als diese sein können, um Fehlinterpretationen bezüglich der Anforderungen zu verhindern. So könnte es sein, dass die Kunden die Anforderungen besser verstehen, wenn die Videos eine hohe Qualität aufweisen. Karras et al. [40] entwickelten ein Qualitätsmodell für Visionsvideos. Dabei erarbeiteten sie [40] zehn Qualitätsmerkmale von Videos und fünf Qualitätsmerkmale von Visionen, die zusammen ein Qualitätsmodell für Visionsvideos bilden, das alle 15 Qualitätsmerkmale umfasst. Das Qualitätsmodell ermöglichte eine Unterstützung bei der Bewertung des Visionsvideos. Die Anforderungen können also durch mehrere Methoden validiert werden.

Pham et al. [57] entwickelten ein interaktives Storyboard, um die Interaktion zwischen Stakeholdern und Anforderungsingenieuren zu optimieren. Es ermöglichte eine Kombination von Fotos, Screenshots, Videos und Audioclips. Auf diese Weise wird das Verständnis der Anforderungen bei den Treffen der Stakeholder verbessert, damit die Anforderungen besser validiert werden können. Karras et al. [41] erstellten interaktive Prototypen und zeichneten die Nutzung dieser auf. Die aufgezeichneten Videos waren zusätzlich eine Unterstützung für textuelle Szenarien und zeigten Interaktionssequenzen der bestehenden Mockups. In diesem Zusammenhang haben Karras et al. [41] anhand einer Auswertung gezeigt, dass so erstellte Videos ein schnelleres Begreifen ermöglichen als statische Mockups.

3.3 Abgrenzungen

Die Forschungen von Fricker et al. [23], Kiesling et al. [44], Creighton et al. [16] beschränken sich insbesondere auf die Analyse von Anforderungen.

Bei der Elicitation werden die Anforderungen mit Hilfe von Workshops erhoben, bevor sie in Form von Videos oder Visionsvideos dargestellt werden. Im Gegensatz dazu existieren die Visionsvideos in dieser Arbeit bereits und sollen zur Elicitation der Anforderungen verwendet werden. Die Anwendung der Arbeit von Nagel et al. [53] wäre schwierig, da es viel Zeit und Geld kosten würde, die Teilnehmer zusammenzubringen, die sich das Video ansehen sollten. Vor allem, wenn am Ende der Arbeit noch eine Benutzerstudie durchgeführt werden musste, um die Validierungsmethode für die erhobenen Anforderungen zu evaluieren. Karras et al. [37] verwendet Textbeschreibungen durch Szenarien, um Visionsvideos zu produzieren. In dieser Arbeit werden die Visionsvideos verwendet, um eine Textbeschreibung mit Hilfe einer natürlichen Sprache zu erstellen. Die von Hurshka et al. [31] vorgestellte Syntax wurde noch nicht auf Visionsvideos angewendet. Aus diesem Grund wird in der vorliegenden Arbeit die Syntax zur textuellen Beschreibung von Anforderungen durch Visionsvideos verwendet.

Bezüglich der Validierung der Anforderungen ist anzumerken, dass Robertson et al. [61] dies allgemein erklärten. Es wird sich in dieser Arbeit besonders auf Visionsvideos konzentriert. Wie in Abschnitt 3.2.2 erwähnt wurde, arbeitet die Forschung kaum an einer Methode, die Visionsvideos mit deren textueller Beschreibung verwendet, was in dieser Arbeit gemacht wird. Pham et al. [57] und Karras et al. [41] hoben das gemeinsame Verständnis zwischen Anforderungsingenieur und Kunden hervor, bewerteten die Validierung der Anforderungen nicht ausführlich. In dieser Arbeit wird das Requirement Engineering in einem agilen Kontext verwendet. Testbare Anforderungen werden durch Visionsvideos textuell beschrieben und validiert. Die Validierungsmethode muss dann die Visionsvideos und die Beschreibungen verbinden und anschließend mit einer Benutzerstudie evaluiert werden.

Kapitel 4

Konzept und Umsetzung

Dieses Kapitel beschäftigt sich mit dem Konzept der Anforderungserhebung und dessen Umsetzung. Es stellt das allererste Ziel dieser Arbeit unter den in Abschnitt 1.1 genannten Zielen dar. Zunächst wird die in dieser Arbeit verwendete Art der textuellen Beschreibung und Notation von Anforderungen vorgestellt. Anschließend wird der Prozess beschrieben, wie präzise Anforderungen von einem einfachen Visionsvideo über eine textuelle Beschreibung formuliert werden können. Abschließend wird in diesem Kapitel erläutert, wie zwischen vagen und testbaren Anforderungen unterschieden werden kann.

Die in dieser Arbeit bereitgestellten Visionsvideos wurden von Studierenden und wissenschaftlichen Mitarbeitenden des Fachgebiets Software-Engineering am Institut für Praktische Informatik an der Leibniz Universität Hannover erstellt. Es handelt sich um eine Sammlung von fünf Visionsvideos, die jeweils ein Projekt in verschiedenen Bereichen behandeln. Das Visionsvideo „**iKühlschrank**“ beispielsweise beschäftigt sich mit einem intelligenten Kühlschrank in Verbindung mit einer mobilen App, die die Verwaltung von Lebensmitteln im Kühlschrank erleichtert. Ein weiteres Beispiel ist das Visionsvideo „**MobileMeetingMaster**“, das eine mobile App entwickelt, die Transkripte von aufgezeichneten Gesprächen/Meetings erstellt. Die App soll den Mitarbeitenden ermöglichen, Protokolle und Notizen auch mobil zu erstellen, um Rückenschmerzen vorzubeugen, die durch langes Sitzen am Computer entstehen können. Die Erstellung des Konzeptes und seine Umsetzung werden anhand dieser Beispiele vorgestellt. Eine detaillierte Beschreibung der anderen Videos ist im Anhang A.2 zu finden.

4.1 Detaillierte Beschreibung der Prozesse zur Erhebung von Anforderungen

In diesem Abschnitt wird dargestellt, wie die oben genannten Visionsvideos beschrieben werden können. Hierzu wird das Behavior-Driven-Development-

Werkzeug zur textuellen Beschreibung von Anforderungen mit dem Format Gherkin [18, 78] verwendet.

Die Abbildung 4.1 präsentiert den Prozess des Ableitens von Anforderungen durch den Anforderungsingenieur.

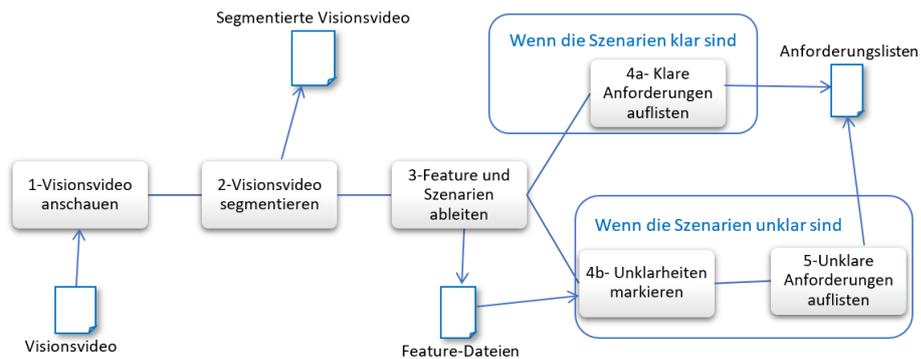


Abbildung 4.1: Prozess der Erhebung von Anforderungen

4.1.1 Schritt 1: Visionsvideo anschauen

Der erste Schritt besteht darin, die Visionsvideos anzusehen, um die Anforderungen durch das Visionsvideo zu erheben. Ein Visionsvideo ist eine Methode, die vom Anforderungsingenieur erstellt wurde, um die Validierung von Anforderungen durch den Kunden zu erleichtern. Das Video kann vom Anforderungsingenieur so oft angesehen werden, bis er in der Lage ist, den Weg eines beliebigen Benutzers des im Visionsvideo dargestellten Tools nachzuvollziehen, ähnlich wie bei der Erstellung von User Journey Maps [22]. Die in diesem Prozess betrachteten Dokumente sind die Visionsvideos.

4.1.2 Schritt 2: Visionsvideo segmentieren

Sobald der Anforderungsingenieur die Schritte aufgelistet hat, die ein potenzieller Nutzer durchlaufen könnte, kann er diese Schritte bearbeiten. Der Anforderungsingenieur kann das Visionsvideo nach diesen Schritten segmentieren, was eine bessere Visualisierung ermöglicht. Er kann sich ausschließlich auf das betreffende Segment konzentrieren und die Anforderungen pro Segment extrahieren. Hier ist zum Beispiel die nummerierte Liste aller Segmente des **iKühlschrank**-Tools:

1. Synchronisation
2. Inventar
3. Einkaufsliste

4.1. DETAILLIERTE BESCHREIBUNG DER PROZESSE ZUR ERHEBUNG VON ANFORDERUNGEN

4. Ablaufdatum

5. Rezeptideen

Nach diesem Schritt entstehen segmentierte Visionsvideos, die bei der Durchführung des nächsten Schritts helfen.

4.1.3 Schritt 3: Feature und Szenarien ableiten

Ab diesem Schritt greift die Gherkin-Syntax, die in Abschnitt 2.4.3 erläutert wurde. Diese Syntax wird auf die verschiedenen verfügbaren Visionsvideos angewendet. In dieser Arbeit wird die deutsche Sprache verwendet. Tabelle 4.1 zeigt, wie die in dieser Arbeit benutzen Schlüsselwörter auf Deutsch übersetzt sind.

ENGLISCH	DEUTSCH
Feature	Feature
Scenario	Szenario
Scenario Outline Examples	Szenarien Beispiele
Given	Gegeben sei
When	Wenn
Then	Dann
And	Und

Tabelle 4.1: Liste der verwendeten Schlüsselwörter [17]

Da die Funktionen, die ein potenzieller Nutzer verwenden sollte, von den verschiedenen Features der Software beschrieben werden, entsprechen die Segmente der Visionsvideos diesen Features. Um eine Anforderung mithilfe der Gherkin-Syntax in einem Visionsvideo besser zu beschreiben, ist es wichtig, die Features zu organisieren. Alle Features können in einer einzigen Datei namens Feature-Datei gesammelt werden, die das Hauptfeature des Projekts darstellt und alle detaillierten Features des Projekts in einer Datei zusammenfasst. Dieses Vorgehen hat den Vorteil, dass es einen Zusammenhang zwischen den verschiedenen Szenarien des Features herstellen kann. Ein Nachteil dieser Vorgehensweise ist jedoch, dass der Kunde aufgrund dieser einen Feature-Datei fälschlicherweise davon ausgehen könnte, dass das Projekt nur ein einziges Feature habe. Dies trifft jedoch in manchen Visionsvideos zu.

In dieser Arbeit wird eine andere Variante angewendet, bei der jedes Feature einer eigenen Feature-Datei entspricht. In diesem Fall wird für jedes Visionsvideo eine oder mehrere Feature-Dateien erstellt, die den Namen des Features tragen. Die Anzahl der Feature-Dateien entspricht der Anzahl der Funktionen des Visionsvideos. Der negative Aspekt besteht in der hohen Anzahl an Schritten pro Feature aufgrund der zahlreichen Feature-Dateien.

Positiv ist jedoch zu erwähnen, dass diese Methode eine detailliertere Beschreibung aller Features und User Stories [71] ermöglicht. Der größte Vorteil dieser Methode besteht darin, dass je detaillierter die Features sind, desto häufiger Probleme entdeckt werden können. Die Tatsache, dass es mehrere Feature-Dateien gibt, erleichtert es, Unklarheiten im Video zu identifizieren, die dann von den Kunden interpretiert werden müssen.

Nachdem die verschiedenen Features formuliert wurden, werden die Szenarien mithilfe der in Abschnitt 2.4.3 detaillierten Gherkin-Syntax abgeleitet. Die Szenarien beschreiben die verschiedenen Unterschritte (die auszuführenden Aktionen und die erwarteten Ergebnisse) zur Ausführung der Funktion wie bei den Abnahmetests [11]. Das folgende Beispiel zeigt die Darstellung einer Feature-Datei für das Feature **Einkaufslisten** des Visionsvideos **iKühlschrank** mit allen Schlüsselwörtern und definierten Schritten.

```

1 #Language: de
2 Feature: Einkaufsliste
3 Als Nutzer
4 moechte ich mit einer Liste einkaufen gehen
5 So dass ich gar keine Produkte mehr vergessen kann
6
7 Szenarien: Automatische Erstellung von Einkaufslisten
8   Gegeben sei, die Lebensmitteln sind automatisch erfasst
9     Und die Dauer bis Ablauf der Lebensmittel ist ermittelt
10    Und optische Sensoren und KI Tools sind eingesetzt
11    Wenn <Detail> beruecksichtigt ist
12    Dann erstellt der iKuehlschrank automatisch eine
13      Einkaufsliste
14      Und kann ich mit dieser Liste einkaufen gehen
15 Beispiele:
16   | Detail |
17   | Fuellstand des iKuehlschranks |
18   | Konsumverhalten |
19   | Das Einkaufen |
20 Szenario: Aktualisierung von Einkaufslisten
21 Gegeben sei, die Einkaufsliste ist automatisch erstellt
22 Wenn neue Produkte in den Kuehlschrank gelegt werden
23 Dann aktualisiert sich die Einkaufsliste automatisch
24 Und ich kann mit dieser Liste einkaufen gehen

```

4.1.4 Schritt 4a: Klare Anforderungen auflisten

Nach der vollständigen Beschreibung der Visionsvideos mit der Gherkin Syntax müssen die Anforderungen aus den Gherkin Texten extrahiert werden. Wenn keine Unklarheiten auftreten und alle beschriebenen Szenarien klar und für den Kunden verständlich sind, besteht der nächste Schritt darin, alle klaren Anforderungen aufzulisten. Wie bereits in Kapitel 2 beschrieben, gibt es funktionale [9] und nicht-funktionale Anforderungen [60]. Da der

4.1. DETAILLIERTE BESCHREIBUNG DER PROZESSE ZUR ERHEBUNG VON ANFORDERUNGEN

Fokus der Kunden darauf liegt, die Software für eine bestimmte Funktion nutzen zu können, werden in dieser Arbeit vor allem die Funktionen und das Verhalten der Software berücksichtigt. Das heißt, dass ausschließlich funktionale Anforderungen erhoben werden.

Anforderungsformulierung

Das Prinzip ist sehr anwenderfreundlich und basiert ausschließlich auf Szenarien. Wie bereits im Abschnitt 2.4.3 beschrieben, stellen die Szenarien die verschiedenen Schritte dar, die zur Ausführung der Funktionen durchlaufen. Gutiérrez et al. [26] vertreten die Ansicht in einem Artikel, dass Gherkin-Szenarien daher nützlich sind, um die zu implementierende funktionale Anforderung besser zu verstehen. Sie [26] führen in derselben Publikation alle Elemente der abstrakten Syntax von Gherkin auf und geben an, welches UML-Element verwendet wird, um sie in einem UML Use Case-Diagramm in Tabelle 4.2 darzustellen.

Gherkin Schlüsselwörter	Elemente in Use Case-Diagramm
Feature	nicht erforderlich
Scenario	Use Case
Scenario outline	Use Case
Step (Given, When, Then...)	Textnote

Tabelle 4.2: Gherkin-Schlüsselwörter und ihre Entsprechung in UML [26]

Sendall et al. [66] fügen in einem Tutorial ebenfalls hinzu, dass Use Cases eine einfache Möglichkeit bieten, Anforderungen zu erfassen. Dies bedeutet, dass es bei der Formulierung von Anforderungen möglich wäre, Use Cases zu verwenden. Aus Abbildung 4.2 lässt sich erkennen, dass Scenario und Scenario Outline in einer Gherkin-Syntax den Use Cases in einem UML-Use-Case-Diagramm entsprechen. Aus diesem Grund sollten Scenario und Scenario Outline auch die Formulierung von Anforderungen ermöglichen. Tabelle 4.3 zeigt zum Beispiel alle Anforderungen für die Funktion **Ablaufdatum** des Visionsvideos iKühlschrank. Im Anhang C werden alle Anforderungen aufgelistet.

SZENARIO	ANFORDERUNGEN
Bestimmung der Haltbarkeit	R8: Der Nutzer muss die Haltbarkeit bestimmen
Ermittlung der Dauer bis Ablauf	R9: Der Nutzer muss die Dauer bis zum Ablaufdatum ermitteln
Benachrichtigung beim Ablauf eines Produktes	R10: Das System muss den Nutzern beim Ablauf eines Produktes benachrichtigen

Tabelle 4.3: Anforderungen für Synchronisierung von iKühlschrank

4.1.5 Schritt 4b: Unklarheiten markieren

In einigen Visionsvideos kann es passieren, dass es Unklarheiten aller Art bei den Szenarien gibt. Unklarheiten in den Szenarien kommen aus fehlenden Daten in den Visionsvideos und/oder Interpretationen des Anforderungsingenieurs. In dieser Arbeit werden die Unklarheiten in die Feature-Datei eingefügt und als Kommentar direkt neben dem Schlüsselwort, in dem die Unklarheit auftritt, markiert. Die Syntax dieses Kommentars ist die normale Syntax eines Kommentars, mit dem einzigen Unterschied, dass der Text in Kommentar die Art der Unklarheit entspricht.

- Bei den fehlenden Angaben handelt es sich um Informationen, die in den Visionsvideos nicht vorhanden sind, die aber zum Verständnis beitragen sollen. Zum Beispiel wird im Fall von fehlenden Daten bei dem Szenario *Account erstellen* des Features *Synchronisation* in dem Visionsvideo **iKühlschrank**

```

1 Szenarien: Account erstellen #Fehlende Angaben
2 Gegeben sei, <Tools> ist funktionsfaehig #Fehlende
  Angaben
3 wenn ich auf den "create account" Link klicke #Fehlende
  Angaben
4 Dann kann ich meine <Daten> eingeben #Fehlende Angaben
5 Wenn ich auf den "create" Button klicke #Fehlende
  Angaben
6 Dann bestaetige ich meinen neuen Account #Fehlende
  Angaben
7
8 Beispiele: #Fehlende Angaben
9 | Tools | Daten |
10 | Internetverbindung | Name |
11 | iKuehlschrank | PIN |
12 | Smartphone | Repeat PIN |
13
```

nicht erwähnt, dass man sich registrieren muss, bevor man Zugang zu den Funktionen der App erhält. In diesem Fall ist diese fehlende Angabe besonders wichtig, da der Kunde zuvor ein Konto erstellt haben muss, um sich anzumelden.

- Im Falle der eigenen Interpretationen geht es um Informationen, die zwar in den Visionsvideos vorhanden sind, aber vom Anforderungsingenieur selbst in Abwesenheit der Kunden angepasst und interpretiert wurden, um das Verständnis zu erleichtern. Zum Beispiel wird im Fall von eigenen Interpretationen bei der Szenario *Auswahl eines Rezepts* des Features *Rezeptideen* in dem Visionsvideo **iKühlschrank**

```

1 Szenario: Auswahl eines Rezepts
2 Gegeben sei, es gibt passende Rezepte
3 Wenn ein Rezept mir gefaellt #Eigene Interpretation
```

4 Dann kann ich das Rezept auswählen.
5

ist die eigene Interpretation von großer Bedeutung, da die Tatsache, dass das Rezept dem Kunden gefällt, keine fehlende Angabe sein kann. Ein Kunde kann das Rezept genauso gut nicht mögen, es aber trotzdem für jemand anderen auswählen.

4.1.6 Schritt 5: Unklare Anforderungen auflisten

Aus diesen Unklarheiten in Szenarien werden dann die unklaren Anforderungen mithilfe des in Abschnitt 4.1.4 beschriebenen Verfahrens aufgelistet. Der Unterschied zu den klaren Anforderungen besteht darin, dass unklare Anforderungen, die sowohl auf fehlende Daten als auch auf eigene Interpretationen basieren, mit einem kleinen Stern * gekennzeichnet werden, wie das Beispiel in Tabelle 4.4 aller Anforderungen für die Funktion **Synchronisierung** des Visionsvideos iKühlschrank zeigt.

SZENARIO	ANFORDERUNGEN
Account erstellen	R1: Der Nutzer muss einen Account erstellen *
Anmeldung in iKühlschrank	R2: Der Nutzer muss sich in der App und im iKühlschrank anmelden können
Synchronisierung des Geräts	R3: Das Smartphone muss mit dem iKühlschrank synchronisiert werden

Tabelle 4.4: Anforderungen für Synchronisierung von iKühlschrank

In Anhang B sind alle Feature-Datei mit den Szenarien und Unklarheiten aufgelistet.

4.2 Anforderungsgrad

Heutzutage werden Funktionen und Aktivitäten in Bezug auf die gesamte Aktion unterteilt. Damit gliedert sich die sogenannte Granularität. In Rahmen dieser Arbeit bezeichnet die Granularität im Allgemeinen die Kleinteiligkeit einer Anforderung. Die Anforderungen nach Granularität [34, 62, 24] werden sich dann zwischen fein-, mittel- und grobgranularen Anforderungen unterscheiden. Die Abbildung 4.2 beschreibt diese genau. Ausgehend von einer Vision oder einem Ziel werden zunächst mehrere grobgranulare Anforderungen erstellt. Aus diesen grobgranularen Anforderungen werden danach Anforderungen mittlerer Granularität erstellt. Für jede dieser Anforderungen mittlerer Granularität werden zum Schluss feingranulare Anforderungen erstellt.

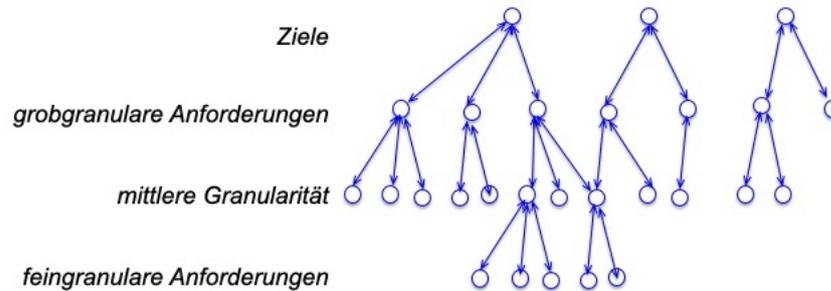


Abbildung 4.2: Anforderungen nach Granularität [34]

Bei den grobgranularen Anforderungen [34, 62, 24] werden ausschließlich die wichtigen Anforderungen, die der Kunde stellt, ohne jedoch tiefer zu gehen, erhoben. Dies hat den Vorteil, dass das Thema überschaubar ist und bei der Durchführung der Studie direkt zum Kern des Themas vorgedrungen werden kann [25]. Das Problem dabei ist, dass die grobgranularen Anforderungen oft unpräzise sind. Sie beschreiben die Anforderungen nicht konkret und es kann auch vorkommen, dass der Kunde bei der Validierung nicht gut zurechtkommt. Ein Beispiel von aggregierter, grobgranularer Sicht auf neue Anforderungen eines Produkts ist die Epics [71].

Bei einer Verfeinerung werden zum Beispiel zu einer groben Anforderung mehrere detailliertere Anforderungen erstellt. Diese nennt man mittlere Granularität [34, 62, 24]. Mit mittlerer Granularität ist gemeint, dass mehrere Anforderungen durch den Requirements Engineer einzeln ermittelt werden müssen. Ein positiver Aspekt ist die Bestimmung einer ersten größeren, unbestimmten Menge von Anforderungen, die ausgearbeitet werden können. Ein negativer Aspekt kann sein, dass mittlere Granularität sich noch verfeinern lässt. Beispiele davon sind natürlichsprachliche oder auch modellbasierte Anforderungen oder Features [12].

Zum Schluss kommen die feingranularen Anforderungen die von mittleren Granularitäten abgeleitet wurden. Systematisch konstruierte, feingranulare Einzelanforderungen [24] enthalten wenig Weglassungen und Mehrdeutigkeiten. Sie vereinfachen die Verwaltung von Anforderungen (Änderbarkeit, Rückverfolgbarkeit). Leider ist die Spezifikation von Zusammenhängen zwischen feingranularen Anforderungen schwierig zu erkennen. Ein Beispiel einer feingranularen, auf einer Sprachschablone basierenden Beschreibung eines Bedürfnisses eines bestimmten Stakeholders ist eine User story [71].

Nachdem die Vor- und Nachteile der drei Anforderungsgrade (grob, mittel und fein) bewertet wurden, wird in dieser Arbeit die Entscheidung für feingranulare Anforderungen getroffen. Epics können in Kapazitäten zerlegt werden, die wiederum in Features zerlegt werden können, die wiederum in

User Stories zerlegt werden können. Im Kontext dieser Arbeit geht es darum, die Anforderungen durch die Visionsvideos mit Hilfe der Gherkin-Syntax zu ermitteln. Gherkin besteht aus Features, die zu mittlerer Granularität gehören. Features bestehen dann aus User Stories, die dem Kunden einen erheblichen Wert bieten. Daher gibt es User Stories in Features, um die Bedürfnisse des Kunden im Feature auszudrücken. Zusätzlich werden die Features in Form von Szenarien beschrieben. Zwar können Szenarien jede Art von Anforderung beschreiben, aber wie in dem vorherigen Unterkapitel beschrieben, kann ein Szenario durch einen Use Case in einem UML Use Case Diagramm modelliert werden [26]. Da Use Cases genauso wie User stories zu den feingranularen Anforderungen gehören [62], gehören auch die Szenarien zu den feingranularen Anforderungen. Aus diesem Grund werden in dieser Arbeit die Anforderungen **feingranular** interpretiert. Grund dafür ist, dass anhand dieser Szenarien die Anforderungen erhoben werden.

4.3 Unterscheidung von Anforderungen

In diesem Abschnitt wird das Prinzip der Unterscheidung genau beschrieben. Nachdem die Anforderungen erhoben wurde, werden die klare Anforderungen als testbar betrachtet und die unklare Anforderungen als vage Anforderung vorgestellt.

4.3.1 Testbare Anforderungen

Es gibt verschiedene Methoden, um zwischen vagen und testbaren Anforderungen zu unterscheiden. Eine davon sind die sogenannte Fit-Kriterien [61]. Die Fit-Kriterien, auf Englisch Fit Criteria zeigen, wie die Messung einer Anforderung diese eindeutig, verständlich und vor allem testbar macht. Das Fit-Kriterium quantifiziert das Verhalten, die Leistung oder eine andere Qualität der Anforderung. Bei funktionalen Anforderungen legt das Fit-Kriterium fest, woher bekannt ist, dass ein Produkt eine Aktion erfolgreich ausgeführt hat. Dies ist eine allgemeine Regel für funktionale Anforderungen: Das Fit Kriterium garantiert, dass eine Funktion erfolgreich ausgeführt wurde. Dies führt wahrscheinlich zu einer Kreation von Testfällen [1]. Ein Testfall [1] ist eine Reihe von Aktionen, die an einem System durchgeführt werden, um festzustellen, ob es die Anforderungen der Software erfüllt und korrekt funktioniert. Die Given-When-Then-Syntax [78] bietet die Möglichkeit, Testfälle für funktionale Anforderungen zu schreiben, da sie sowohl die Voraussetzungen, die zu befolgenden Aktionen als auch das erwartete Ergebnis des Szenarios beziehungsweise der Anforderung präsentiert. Dadurch können diese Anforderungen testbar gemacht werden.

4.3.2 Vage Anforderungen

In dieser Arbeit können lediglich die Anforderungen aus dem Visionsvideo oder die von den Kunden interpretierten Anforderungen getestet werden. Diese testbaren Anforderungen müssen von den in den Visionsvideos enthaltenen Softwarefunktionen und den Interessen der Kunden abgeleitet werden. Einige Anforderungen können daher aufgrund von Unklarheiten in den Szenarien nicht getestet werden. Unklarheiten beziehen sich auf fehlende Angaben in den Visionsvideos und/oder Szenarien, die vom Anforderungsingenieur in Abwesenheit des Kunden interpretiert wurden. Obwohl diese Anforderungen für das Verständnis des Visionsvideos relevant sind, werden sie nicht vom Kunden gestellt. Aus diesem Grund können sie nicht getestet werden und werden als vage Anforderungen betrachtet. Der Unterschied zu den testbaren Anforderungen bei der Formulierung besteht darin, dass vage Anforderungen genauso wie unklare Anforderungen mit einem kleinen Stern * gekennzeichnet werden. Das Beispiel in Tabelle 4.4 zeigt Anforderungen, die aus ungeklärten Szenarien stammen. Hierbei handelt es sich um vage Anforderungen.

Im Anhang C sind alle vagen und testbaren Anforderungen aufgelistet

4.4 Interpretation

In diesem Unterkapitel geht es darum, dass potenzielle Kunden, die die in den Videos gezeigten Tools verwenden möchten, die Anforderungen an das Tool interpretieren und mit den in den Gherkin-Texten beschriebenen Anforderungen vergleichen. Die textuell beschriebenen Anforderungen sollen den Kunden ein besseres Verständnis der Anforderungen ermöglichen. Um eine bessere Interpretation zu ermöglichen, wird eine Benutzerstudie erstellt, um das Verständnis der Anforderungen durch die Kunden zu vergleichen, und zwar zunächst ausschließlich anhand der Visionsvideos und dann zusätzlich anhand der Gherkin-Texte. Im folgenden Kapitel 5 werden die Vorgehensweisen bei der Benutzerstudie erläutert.

Kapitel 5

Vorgehen bei der Benutzerstudie

In diesem Kapitel geht es um den Entwurf und die Durchführung einer Benutzerstudie. Diese Benutzerstudie entspricht, wie in Abschnitt 4.4 dieser Arbeit erwähnt wurde, einer Interpretation von Anforderungen. Sie ermöglicht es, die Anforderungen in den Visionsvideos ohne Gherkin-Texte mit denen in den Visionsvideos mit Gherkin-Texten zu vergleichen.

Die Planungsschritte, die von Wohlin et al. [77] entwickelt wurden, werden für das Experiment angewendet. Abbildung 5.1 illustriert den Verlauf der Experimentplanung mit den entsprechenden Phasen.

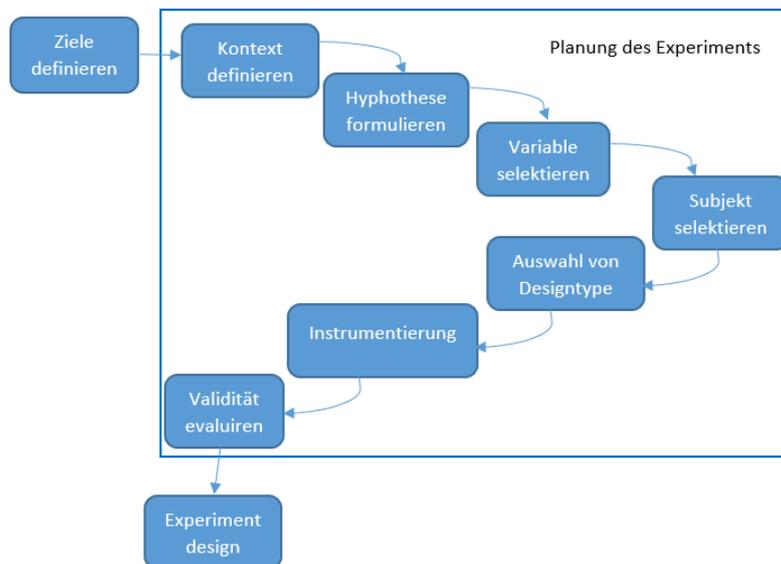


Abbildung 5.1: Überblick von Planungsphasen nach Wohlin et al. [77]

Laut Wohlin et al. [77] besteht der Vorteil eines Experiments in der Kontrolle verschiedener Einflussfaktoren. Es ermöglicht somit den Schluss auf allgemeingültige Ergebnisse und die Durchführung statistischer Analysen mithilfe von Verfahren zur Prüfung von Hypothesen. So muss zunächst das Ziel des Experiments definiert werden. Diese Zieldefinition ermöglicht dann die Planung des Experiments und seiner verschiedenen Phasen. Das Resultat dieser Experimentplanung ist das endgültige Experimentdesign. Sofern nicht anders angegeben, wird in diesem Kapitel der Leitfaden von Wohlin et al. [77] verwendet.

5.1 Vorbereitung mit GQM

Um das Experiment zu strukturieren, wird die Experimentvorbereitung in diesem Abschnitt detaillierter dokumentiert. Der systematische Ansatz des Goal-Question-Metric (GQM) Paradigmas von Solingen et al. [73] wird dabei helfen, Fragen, Metriken und Indikatoren zur Messung der Ziele zu erstellen und entsprechende Ergebnisse in Anbetracht dieser Ziele zu gewinnen. In diesem Abschnitt wird das Experimentdesign damit beginnen, das Ziel der Forschung zu definieren und es in Fragen und Metriken zu verfeinern. Der Zweck dieser Fragen und Metriken wird darin bestehen, eine Reduzierung der Arbeitsbelastung durch die Datenerhebung zu ermöglichen und Fehlinterpretationen zu vermeiden.

5.1.1 Ziele:

Um sicherzustellen, dass das Ziel dieses Experiments klar definiert ist, wird das von Wohlin et al. [77] bereitgestellte Zielmodell angewendet. Der Zweck der Verwendung dieses Modells ist die Definition der Hauptaspekte des Experiments, um danach mit der Planung und Durchführung des Experiments zu beginnen.

Ziel 1 der Benutzerstudie

Analysiere den Einsatz von Visionsvideos und dazugehörigen Gherkin-Texten

mit dem Ziel, Missverständnisse in der Anforderungen von Kundenansicht zu klären,

in Bezug auf Effektivität der Gherkin-Texte,

aus Sicht von potenziellen Kunden,

im Rahmen von einer Studie mit zwei Gruppen.

In dieser Studie sind Missverständnisse die Anforderungen, die nicht vollständig beziehungsweise falsch in Gherkin Texten beschrieben sind. Einige Szenarien wurden in Gherkin selber statt im Video interpretiert. Die

Testpersonen werden während der Studie gebeten, die Szenarien zu vervollständigen. Die Missverständnisse betrachten ausschließlich die Gruppe, die Visionsvideos und Gherkin Texten bekommen.

Ziel 2 der Benutzerstudie

Analysiere den Einsatz von Visionsvideos und dazugehörigen Gherkin-Texten

mit dem Ziel, Unterschiede bei der Anzahl von Anforderungen von Kundenansicht zu finden,

in Bezug auf Effektivität der Gherkin-Texte,

aus Sicht von potenziellen Kunden,

im Rahmen von einer Studie mit zwei Gruppen.

In dieser Studie sind Unterschiede die Differenz zwischen der Anzahl von Anforderungen aus Visionsvideos mit Hilfe von Gherkin-Texte und Visionsvideos ohne Hilfe von Gherkin-Texten. Die Grundidee ist das Vergleichen zwischen Visionsvideos mit Gherkin-Texten und Visionsvideos ohne Gherkin-Texte. Gherkin Texte sollen Kunden helfen, Anforderungen zu formulieren. Wenn Gherkin-Texte Kunden geholfen haben, könnten Kunden eine unterschiedliche Anzahl von Anforderungen haben.

5.1.2 Forschungsfragen

Nachdem das Ziel der Studie definiert ist, wurden Forschungsfragen formuliert und diese durch Metriken und Hypothesen verfeinert. Ausgehend vom ersten Ziel ergibt sich die folgende Forschungsfrage:

- **Forschungsfrage 1:** Wie können Anforderungsingenieur mit Kunden gefundenen Missverständnisse in der Anforderungen aus Visionsvideos mit Hilfe von Gherkin-Texten klären?

Ausgehend vom zweiten Ziel ergibt sich die folgende Forschungsfrage:

- **Forschungsfrage 2:** Wie unterscheidet sich die Anzahl der Anforderungen bei der Betrachtung von Visionsvideos mit Hilfe von Gherkin-Texten im Vergleich zu Visionsvideos ohne Gherkin-Texte?

5.1.3 Metriken

Tabelle 5.1 zeigt einen Überblick über alle Metriken, die für jede Forschungsfrage berücksichtigt wurden.

Forschungsfragen	Metriken
1	$M_{1,1}$ - Anzahl von falsch beschriebenen Anforderungen mit Hilfe von Gherkin-Texten $M_{1,2}$ - Anzahl von unvollständig beschriebenen Anforderungen mit Hilfe von Gherkin-Texten
2	$M_{2,1}$ - Anzahl von gefundenen Anforderungen ohne Hilfe von Gherkin-Texten $M_{2,2}$ - Anzahl von gefundenen Anforderungen mit Hilfe von Gherkin-Texten

Tabelle 5.1: Metriken

5.2 Kontext

Nach der Definition der Ziele, die mit der Studie erreicht werden sollen, besteht der erste Schritt der Experimentplanung darin, den Kontext für die Studie zu definieren. In dieser Arbeit wird die Studie jedoch in einer **Offline** Umgebung durchgeführt. In einer Offline-Umgebung wird die Umgebung speziell zum Lernen und Nachahmen eines realen Projekts geschaffen. Einer der Vorteile einer Offline Umgebung ist die Möglichkeit, verschiedene Aspekte (Themen, Struktur, Durchführungsort und vor allem die Durchlaufdauer) des Projekts zu kontrollieren. Des Weiteren können die Visionsvideos, die in der Studie verwendet werden, ohne Rücksicht auf den Kontext eines realen Softwareprojekts vom Experimentierenden kontrolliert werden.

Obwohl Fachleute aufgrund ihrer Erfahrung gegenüber Studierenden bevorzugt sind, wurden in dieser Studie aufgrund der Notwendigkeit einer großen Anzahl von Probanden Studierende eingesetzt. Es wird festgestellt, dass aufgrund der Offline-Umgebung und der vorhandenen Visionsvideos in dieser Studie kein reales Problem untersucht wird, sondern ein sogenanntes *Toy Problem* [77], das typischerweise kleiner als reale Probleme ist und häufig speziell für Experimente erstellt wird. Die vorliegende Studie entspricht dieser Beschreibung, da die verfügbaren Visionsvideos von Studierenden der Leibniz Universität Hannover erstellt wurden und ein Problem von relativ geringer Größe untersucht wird. Die Ergebnisse der Studie werden in einen bestimmten Kontext übertragen, da nur Probanden ausgewählt wurden, die Interesse an den in den Visionsvideos vorgestellten Tools haben.

5.3 Hypothesen

Eine Hypothese wird formell aufgestellt und die im Laufe des Experiments gesammelten Daten werden verwendet, um die Hypothese nach Möglichkeit zu verwerfen. Für jede Metrik, die quantitative Daten vergleicht, werden

im Vorfeld Hypothesen aufgestellt. Sie dienten als Referenzpunkt, mit dem die Ergebnisse verglichen werden konnten. Zur Ableitung der Hypothesen dieser Arbeit werden hier die Ziele 1 und 2 aus dem Abschnitt 5.1.1 zugrunde gelegt. Die Umsetzung von Ziel 1 kann unmittelbar in die folgende Hypothese in Tabelle 5.2 integriert werden.

Ziele	Ziel 1
Nullhypothese	- $H_{0,1,1}$: Es gibt keine Anforderungen, die in den gezeigten Gherkin-Texten richtig beschrieben werden müssen. - $H_{0,2,1}$: Es gibt keine Anforderungen, die in den gezeigten Gherkin-Texten vervollständigt werden müssen.
Alternativhypothese	- $H_{1,1,1}$: Es gibt Anforderungen, die in den gezeigten Gherkin-Texten richtig beschrieben werden müssen. - $H_{1,2,1}$: Es gibt Anforderungen, die in den gezeigten Gherkin-Texten vervollständigt werden müssen.

Tabelle 5.2: Hypothese zum Ziel 1

Die Umsetzung von Ziel 2 kann unmittelbar in die folgende Hypothese in Tabelle 5.3 integriert werden.

Ziele	Ziel 2
Nullhypothese	- $H_{0,2,2}$: Es gibt keinen Unterschied zwischen dem Betrachten von Visionsvideos mit Hilfe von Gherkin Texten und dem Betrachten ausschließlich von Visionsvideos in Bezug auf die Variable Anzahl von Anforderungen.
Alternativhypothese	- $H_{1,2,2}$: Es gibt einen Unterschied zwischen dem Betrachten von Visionsvideos mit Hilfe von Gherkin Texten und dem Betrachten ausschließlich von Visionsvideos in Bezug auf die Variable Anzahl von Anforderungen.

Tabelle 5.3: Hypothese zum Ziel 2

5.4 Variable

Die Wirksamkeit der durchgeführten Verfahren wird anhand der **abhängigen Variablen** gemessen. Dabei ist die abhängige Variable in den meisten Fällen nicht direkt messbar und erfordert stattdessen die Anwendung eines indirekten Maßes. Die vorliegende Arbeit identifiziert die in Tabelle 5.1 beschriebenen Metriken als abhängige Variablen.

Als **unabhängigen Variablen** werden jene Variablen bezeichnet, die in der Studie manipuliert und kontrolliert werden können. In dieser Studie werden die folgenden Variablen als unabhängige Variablen identifiziert:

- Die Visionsvideos (Komplexität, Länge, Zeit für die Betrachtung, Qualität)
- Inhaltliche Fragen (Umfrage und Gherkin Schulung)
- Erforderliche Betreuung für jeden Probanden
- Das Tools Vivi-Player
- Die Gherkin Texte (Komplexität, Länge, Zeit für die Betrachtung, Qualität)

5.5 Probandenauswahl

Für die Durchführung dieser Studie ist es sehr wichtig, dass die Probanden gut ausgewählt werden. Diese Auswahl ermöglicht es unter anderem, die Ergebnisse der Studie zu verallgemeinern. Damit das Ergebnis so verallgemeinerbar wie möglich ist, müssen die Probanden die Population repräsentieren, auf die die Ergebnisse angewendet werden sollen. Die Gesamtheit der Probanden entspricht also einer Stichprobe der Bevölkerung. Die Studie wurde von potenziellen Nutzern des in den Visionsvideos gezeigten Werkzeugs durchgeführt, die über Kenntnisse der Gherkin-Sprache verfügen. Da für diese Arbeit jedoch keine Gherkin-Experten zur Verfügung standen und nicht alle Nutzer des Tools Gherkin-Experten sein müssten, wurden Studierende für die Studie ausgewählt. Diese Studierende werden in die Gherkin-Sprache und die Art und Weise ihrer Verwendung eingewiesen.

Gruppeneinteilung der Probanden

Zu Beginn dieser Arbeit wurde eine Liste potenzieller Probanden erstellt, aus der die Probanden für die Studie ausgewählt wurden. Wie im Kontext der Studie erwähnt, waren die potenziellen Probanden ausschließlich Studierende. Es handelt sich dabei um Studierende aller Geschlechter, Studiengänge und akademischen Grade. Es wurden keine Vorkenntnisse über den RE-Prozess vorausgesetzt. Die Probanden wurden entsprechend

der Zeit, die sie für die Teilnahme an der Studie zur Verfügung hatten, eingeteilt. Im Anschluss daran wurden diese potenziellen Probanden in einer Umfrage gebeten, das Visionsvideo auszuwählen, in dem das Tool vorgestellt wird, das sie potenziell verwenden möchten. Eine detailliertere Beschreibung der Umfrage wird im nächsten Abschnitt dargestellt. Nach der Auswahl der Videos wurden die Probanden mit einem Briefing konfrontiert, an das sich ein Test zur Kenntnis der Gherkin-Sprache anschloss. Dieser Test lieferte eine Vorstellung von der Vorgehensweise bei der Formulierung der Gherkin-Syntax, die zur textuellen Beschreibung der Anforderungen in den Visionsvideos verwendet wird. Am Ende des Tests wurden 20 Personen ausgewählt. Diese 20 Personen wurden in zwei Gruppen zu je zehn Personen aufgeteilt: Eine Gruppe bekam Visionsvideos und Gherkin-Texte, die anderen ausschließlich Visionsvideos.

Visionsvideosauswahl durch Probanden

Die Auswahl der Visionsvideos, die in dieser Studie verwendet wurde, ist für die Probanden von großer Bedeutung. Da die Studie für potenzielle Nutzer des in den Visionsvideos dargestellten Tools konzipiert ist, sollten die Probanden selbst entscheiden können, welches Tool sie nutzen können und vor allem wollen. Die Visionsvideos wurden von den Probanden nach ihren Interessen ausgewählt. In Kapitel 4 wurden fünf Visionsvideos für die Anforderungserhebung zur Verfügung gestellt. Mithilfe einer kleinen Umfrage sollten die Probanden eine kurze Beschreibung aller fünf Visionsvideos lesen und entscheiden, an welches Tool sie das größte Interesse haben könnten. Jeder Proband konnte an dieser Umfrage je nach seiner Verfügbarkeit teilnehmen. Die Umfrage wurde mithilfe der Plattform *umfrageonline* [72] durchgeführt, bei der die Probanden aufgefordert wurden, ihre Präferenzen bezüglich bestimmter Produkte selbst einzutragen. Am Ende der Umfrage wurden zwei Videos, die für die Probanden am interessantesten waren, ausgewählt und für diese Studie verwendet. Um an dieser Umfrage teilzunehmen, musste lediglich auf den folgenden Link hier geklickt werden. Für diese Arbeit wurde eine kostenlose (für 30 Tage verfügbare) Version des Tools verwendet, weshalb der Link nur für 30 Tage verfügbar war. Der detaillierte Inhalt der Umfrage ist im Anhang A.3 zu finden.

5.6 Experiment Design

Um die Ergebnisse der Studie sinnvoll auswerten zu können, musste sie detailliert geplant und konzipiert werden. Welche statistischen Analysen angewandt werden können, hängt von dem gewählten Design ab. Daher sind Planung und Interpretation stark miteinander verbunden [77]. Die generellen Planungsprinzipien sind Randomisierung, Blockierung und Ausgleich [77].

Die Studie wurde in Form eines kontrollierten Experiments mit einem Experimentdesign zwischen zwei Gruppen von genau 10 Personen pro Gruppe durchgeführt (**Ausgleich**). Die Kontrollgruppe sah sich nur die Visionsvideos an und beantwortete die Fragen. Die Studiengruppe sah sich die Videos an und las die dazugehörigen Gherkin-Texte. Die Probanden wurden nach dem Zufallsprinzip den Gruppen zugeordnet (**Randomisierung**). Zu Beginn der Studie wurde ein kurzes Tutorial für die Probanden erstellt, um sicherzustellen, dass sie die Tools richtig verwenden und die Gherkin-Syntax verstehen konnten. Zusätzlich wurden den Probanden alle Schritte der Studie als PDF-Datei zur Verfügung gestellt, um zu vermeiden, dass sie aufgrund mangelnder Kenntnisse Schwierigkeiten beim Durchführen der Studie haben (**Blockierung**).

5.7 Instrumentierung

Das allgemeine Ziel der Instrumentierung besteht darin, Mittel für die Durchführung des Experiments und seine Beobachtung zur Verfügung zu stellen, ohne die Kontrolle der Studie zu beeinträchtigen [77]. Es gibt drei Sorten von Instrumenten, nämlich Objekte, Richtlinien und Messinstrumente [77].

Objekte

Bei der Planung dieser Studie ist es wichtig, geeignete Objekte auszuwählen. Zu den benötigten Objekten für dieser Studie gehören:

- **Die Visionsvideos:** Daraus werden die Anforderungen abgeleitet.
- **Die Gherkin-Texte:** Dort werden die Anforderungen textuell beschrieben.
- **Online Umfrage:** Um Visionsvideos nach der Präferenz der Probanden auszuwählen.
- **Inhaltlichen Fragen:** Die Fragen dienen dazu, den Probanden Grundkenntnisse der Gherkin Syntax zu vermitteln.
- **Das unterstützende Tool (VIVI Player):** Dort werden die Videos, die Gherkin-Texte und die Fragen hochgeladen.
- **Einladungslink:** Es sollte ein Link erstellt werden, der den Probanden zur Teilnahme am Experiment einlädt und zum entsprechenden Tool führt.
- **E-Mail von Probanden:** Die Probanden werden die Einladungslink per E-mail bekommen. Dafür benötige jeder eine E-mail Adresse.

Richtlinien

Es werden Richtlinien benötigt, um die Teilnehmer der Studie anzuleiten. Im Rahmen dieser Studie werden zwei verschiedene Verfahren verglichen. Ein Dokument, das den Ablauf der Studie beschreibt, ist notwendig, um den Probanden zu erklären, was sie bei jedem Schritt des Experiments tun müssen, indem sie eine detaillierte Beschreibung der Aufgabe und der Art und Weise erhalten, wie sie den jeweiligen Schritt durchführen sollen. Diese Richtlinien sind im Anhang D.1 zu finden.

Messinstrumente:

In dieser Studie werden die Messungen mithilfe von Datenerhebungen durchgeführt. Die Daten werden mithilfe von manuellen Bögen gesammelt. Aus diesem Grund ist das Messinstrument im Rahmen dieser Studie **die Fragebögen** und im Anhang D.5 zu finden.

5.8 Methode und Ablauf

Die Studie wird zu Beginn mit mehr als 30 Teilnehmern durchgeführt. Die Anzahl der Teilnehmer kann sich im Laufe des Prozesses entweder erhöhen oder verringern. Für den Anfang benötigt der Teilnehmer nur eine E-Mail-Adresse, einen Laptop (Tablet oder Telefon) und eine gute Internetverbindung. Sobald diese Voraussetzungen erfüllt sind, erhält er eine erste E-Mail mit dem Link für die Umfrage zur Auswahl der Videos. Sobald die Auswahl getroffen wurde und die Ergebnisse verfügbar sind, werden die beiden Videos mit dem größten Interesse für die weitere Studie festgelegt. Wenn das Interesse gleich groß ist, d.h. wenn es nicht möglich ist, die beiden Videos zu sortieren, werden nach und nach weitere Teilnehmer hinzugefügt, bis die beiden gewünschten Videos vorliegen. Es können auch Teilnehmer hinzugefügt werden, wenn weniger als 30 Personen an einer möglichen Studie über die Tools interessiert sind, für die sie sich interessieren. Es wird nur die Wahl der Personen berücksichtigt, die an der Studie interessiert sind. Diese Probanden werden gebeten, ihre E-Mail-Adresse für die weitere Untersuchung zu hinterlassen.

Anschließend wird ein Big Blue Button-Link zu einem virtuellen Raum gesendet, in dem die Studie durchgeführt wird. Der Link wird mit einer kurzen Beschreibung der zu erledigenden Aufgaben in Anhang D zu finden, an alle Teilnehmer, die ihre E-Mail-Adresse hinterlassen haben, gesendet. In diesem Raum wird diesen Einladungslink hier geschickt. Der Link dient dazu, am Quiz zum Verständnis der Gherkin-Syntax, genauer gesagt des Schlüsselworts Szenario und Szenario-Outline, teilzunehmen. Der Teilnehmer liest zuerst eine kurze Zusammenfassung der Syntax und beantwortet dann mit Hilfe dieser Zusammenfassung die gestellten Fragen.

Die Tests werden mit jedem Probanden durchgeführt, bis es zehn Probanden gibt, die den Test bestehen. Diese zehn Probanden erhalten während der Studie Visionsvideos und Gherkin-Texte. Aus den restlichen Probanden (diejenigen, die den Test nicht bestanden haben und diejenigen, die keinen Test gemacht haben) werden zehn Probanden zufällig für die weitere Studie ausgewählt. Diese zehn Probanden werden wiederum nur Visionsvideos erhalten.

Betrachtung von Visionsvideos ohne Gherkin-Texte(Gruppe 1)

Direkt nach dem Test erhält der Proband, der die Videos ohne Gherkin-Texte sehen wird, den Einladungslink und das Passwort zur Studie, wenn der Test nicht erfolgreich war, oder für diejenigen, die keinen Test durchgeführt haben. Sobald die Probanden sich eingeloggt haben, kann das Meeting fortgesetzt werden. Die Probanden sehen das komplette Video zum ersten Mal und haben fünf bis zehn Minuten Zeit, um die Fragen zu lesen. Danach wird das Video vom Moderator segmentiert. Die Probanden sehen das Video ein zweites Mal, diesmal bereits segmentiert, und haben 10 bis 15 Minuten Zeit, um Fragen zu beantworten.

Betrachtung von Visionsvideos mit Gherkin-Texten (Gruppe 2)

Falls der Test von der Gruppe durchgeführt wird, die die Visionsvideos mit Gherkin-Texten erhält, wird nach erfolgreichem Bestehen des Tests ein Einladungslink und das Passwort zur Studie an die Probanden gesendet. Sobald sich die Probanden eingeloggt haben, kann das Meeting fortsetzen. Die Probanden sehen das komplette Video ein erstes Mal und haben fünf bis zehn Minuten Zeit, um die Gherkin-Texte zu lesen. Danach haben die Probanden weitere fünf bis zehn Minuten Zeit, um ein erstes Mal alle Fragen zu lesen. Anschließend wird das Video vom Moderator segmentiert. Die Teilnehmer sehen das Video ein zweites Mal und dieses Mal bereits segmentiert. Nach jedem Segment haben die Probanden Zeit, um die Szenarien zu lesen, die dem Segment entsprechen. Am Ende der Wiedergabe des Videos haben die Probanden circa 15 Minuten Zeit, um Fragen zu beantworten.

Die Ergebnisse werden exportiert und im nächsten Kapitel analysiert. Danach müssen die Probanden das Formular mit den Informationen über die Probanden ausfüllen und die Einverständniserklärung unterschreiben. Sie sind alle im Anhang D zu finden. Abbildung 5.2 zeigt die verschiedenen Etappen der Studie aus der Sicht des Probanden. Die roten Schritte sind die Schritte, die von der Gruppe mit durchgeführt wurden, und die grünen Schritte werden von der Gruppe 2 durchgeführt.



Abbildung 5.2: Ablauf der Studie

5.9 Threats to Validity

Nach Wohlin et al. [77] lassen sich die Risiken der Validität wie folgt unterscheiden

5.9.1 Konklusion Validität

Die Validität der Konklusion wurde sichergestellt, indem stets darauf geachtet wurde, dass die für die Studie erforderlichen Vorbedingungen gegeben waren. Eine mögliche Bedrohung für die Validität dieser Arbeit ist die Granularität der von den Kunden formulierten Anforderungen. Um die Anzahl der Anforderungen von zwei Gruppen vergleichen zu können, müssen die Anforderungen aus gleicher Granularität von zwei Gruppen gesammelt werden. Anforderungen aus der Gruppe ohne Gherkin-Texte sind freie Texte. Anforderungen aus der anderen Gruppe mit Gherkin-Texte sind ebenso freie

Texte und vervollständigte Gherkin-Schritte.

Um dieses Ergebnis zu erreichen, muss lediglich die Granularität dieser gesammelten Anforderungen festgelegt werden. In den vorangegangenen Kapiteln wurde bereits gezeigt, dass die Anforderungen, die von Szenarien in Gherkins-Texten gestellt werden, vom Typ feingranular sind. Die anderen Anforderungen stammen alle von den Kunden, nachdem sie sich ein Video angesehen haben, das in Szenarien unterteilt ist. Das bedeutet, dass für jedes Segment die Anforderungen von den Kunden gestellt werden. Da jedes Segment einem Szenario entspricht, haben diese Anforderungen die gleiche Granularität wie bei Szenarien in Gherkin-Texten. Die gesammelten Anforderungen sind also vom Typ **feingranular**.

5.9.2 Interne Validität

Die Gestaltung von Experimenten ist von der internen Validität abhängig, da Probanden dazu neigen, andere Probanden mit denselben Behandlungen zu imitieren, wenn sie die Ergebnisse der anderen Probanden kennen. Ein solches Verhalten kann zu einer signifikanten Verzerrung der Daten und damit des Ergebnisses führen. Daher ist es notwendig, dass der Experimentator Maßnahmen ergreift, um den Austausch von Informationen zwischen den Versuchspersonen zu verhindern. Die Daten, die von anderen Probanden erzeugt wurden, dürfen auch nicht weitergegeben werden, um die Integrität des Experiments zu gewährleisten.

5.9.3 Konstruktvalidität

Eine der mit Design Threats verbundenen Gefahren besteht darin, dass ein oder mehrere Konstrukte bei der Erstellung dieser Arbeit nicht genau genug definiert wurden. Diese Ungenauigkeit würde sich dann negativ auf die aus der Theorie dieser Arbeit abgeleiteten Behandlungen auswirken. Beispielsweise könnte der Gherkin Schulung von den Probanden falsch verstanden werden, was die Bewertung verfälschen könnte. Aus diesem Grund werden bei einer eventuellen Nachfrage Begriffe zur Gherkin-Syntax noch spezifiziert.

5.9.4 Externe Validität

Ein Risiko besteht darin, dass Studierende ohne Hintergrundwissen im Requirements Engineering die Kunden des Visionsvideos sind. Sie können zwar die potenziellen Kunden sein, aber es fehlt ihnen das echte Interesse an dem im Visionsvideo gezeigten Produkt. Im Rahmen dieser Arbeit wurde diese Bedrohung insofern gelöst, als eine Umfrage durchgeführt wurde, um tatsächlich herauszufinden, ob diese Studierende ein Interesse an den entsprechenden Produkten haben oder nicht.

Kapitel 6

Ergebnisse und Interpretation

Dieses Kapitel wird sich auf die Ergebnisse der oben durchgeführten Studie und eine Bewertung der erzielten Ergebnisse konzentrieren. Hier werden die Ergebnisse und ihre Interpretation in Bezug auf die verschiedenen Hypothesen, die im vorherigen Kapitel formuliert wurden, dargestellt.

6.1 Vorbereitung der Studie

Am Ende dieser Studie werden mehrere Ergebnisse erwartet. Um die Studie sinnvoll bewerten zu können, müssen alle erhobenen Parameter ausgewertet werden. Hierzu zählen die Auswahl der Visionsvideos, die demographischen Daten der Probanden und die Gherkin Schulung.

6.1.1 Visionsvideoseauswahl

Um die zwei Visionsvideos auszuwählen, die für diese Studie verwendet werden sollen, wurde eine Umfrage mit acht Fragen durchgeführt, die sich auf die Auswahl der Tools durch die Probanden nach ihren Wünsche bezog. Es handelte sich um eine Liste der folgenden fünf Tools: **iKühlschrank**, **Watertracer**, **MobileMeetingMaster**, **CVparser** und **Intelligent Coach**.

Genau 60 Probanden nahmen an der Umfrage teil und 25 davon hinterließen ihre E-Mail-Adresse für die weitere Durchführung der Studie. Aus dieser Umfrage geht hervor, dass die beiden am meisten genutzten Tools **IntelligentCoach** mit 42 Stimmen und **CVParse** mit 39 Kunden sind.

Um genauere und präzisere Ergebnisse zu erhalten, sollten lediglich die Ergebnisse der Teilnehmer berücksichtigt werden, die an der Studie teilnehmen wollten und ihre E-Mail-Adresse hinterlassen haben. Diese Teilnehmer werden von der Studie und folglich von den Visionsvideos betroffen sein, die ausgewählt werden.

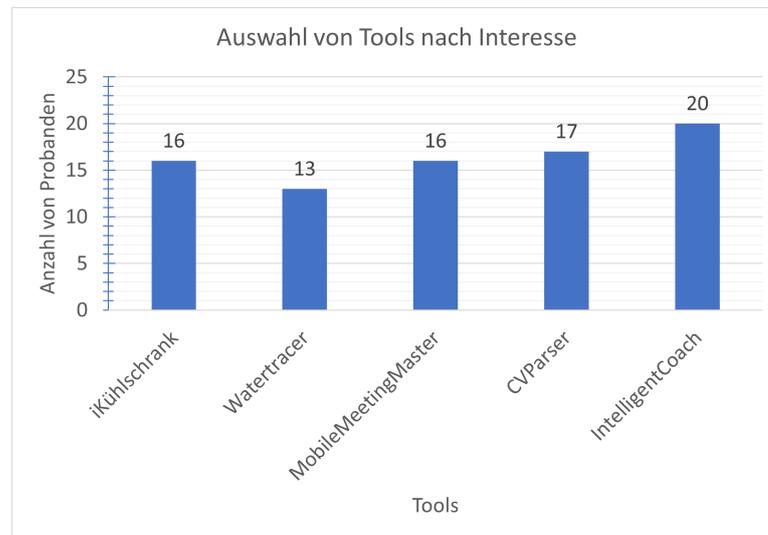


Abbildung 6.1: Ergebnisse der Umfrage.

Abbildung 6.1 zeigt dann die Ergebnisse dieser 25 Teilnehmer und ihre Stimmenverteilung. Es ist ebenfalls festzustellen, dass die beiden gleichen Tools, die im vorherigen Abschnitt am häufigsten gewählt wurden, auch in dieser Teilmenge von Teilnehmern bevorzugt wurden. Es gab 20 Probanden, für die IntelligentCoach von Interesse ist, und 17 Probanden, für die CVParse von Interesse ist. Watertracer, auf der anderen Seite, hat das geringste Interesse, da für dieses Tool ausschließlich 13 Personen stimmten.

Für die Fortsetzung dieser Studie werden daher ausschließlich die Visionsvideos berücksichtigt, die die Tools **IntelligentCoach** und **CVParse** präsentieren.

6.1.2 Demographische Daten

Von den 25 Probanden, die in der Phase der Auswahl der Visionsvideos registriert wurden, nahmen lediglich 20 Probanden an der Studie teil. Die restlichen 5 Probanden mussten aus Verfügbarkeitsgründen absagen. Diese 20 Teilnehmer waren ausschließlich Studierende im Alter von 20 bis 30 Jahren, die in Hannover wohnen. Tabelle 6.1 zeigt zusätzliche Informationen wie Studiengang, Fachgebiete, aktuelles Semester und Universität, an der sie studieren. LUH, HsH und MHH stehen jeweils für Leibniz Universität Hannover, Hochschule Hannover und Medizinische Hochschule Hannover.

TN	Geschlecht	Studiengang	Fachbereich	Semester	HS
1	Männlich	Master	Informatik	2	LUH
2	Männlich	Master	Informatik	3	LUH
3	Weiblich	Master	Maschinenbau	5	LUH
4	Männlich	Bachelor	Chemie	5	HsH
5	Weiblich	Bachelor	Medizin	5	MHH
6	Männlich	Master	Informatik	3	HsH
7	Männlich	Master	Elektrotechnik	5	LUH
8	Männlich	Master	Informatik	6	LUH
9	Männlich	Bachelor	Chemie	4	LUH
10	Weiblich	Master	Informatik	4	LUH
11	Männlich	Bachelor	Energietechnik	9	LUH
12	Männlich	Bachelor	Informatik	7	HsH
13	Männlich	Bachelor	Chemie	6	HsH
14	Weiblich	Master	Elektrotechnik	5	LUH
15	Männlich	Master	Informatik	3	LUH
16	Männlich	Master	Maschinenbau	3	HsH
17	Männlich	Master	Informatik	7	LUH
18	Weiblich	Bachelor	Informatik	6	LUH
19	Männlich	Bachelor	Maschinenbau	5	LUH
20	Weiblich	Master	Medizin	1	MHH

Tabelle 6.1: Demographische Daten

6.1.3 Gherkin-Schulung

Gemäß dem Ablaufplan der Studie werden die Teilnehmer in zwei Gruppen eingeteilt: die erste Gruppe ohne Gherkin-Texte und die zweite Gruppe mit Gherkin-Texten. Bei der zweiten Gruppe sollten die Probanden zumindest über Grundkenntnisse der Gherkin-Syntax verfügen. Allerdings hatten viele, wenn nicht sogar alle Probanden noch überhaupt nichts von der Gherkin-Syntax gehört. Die Gherkin-Schulung sollte daher den Probanden helfen, die Syntax zu verstehen. Um das Verständnis des Konzeptes zu bewerten, wird ein kleiner Test mit fünf multiple-choice Fragen zu je vier Punkten (siehe Anhang D.6) durchgeführt. Abbildung 6.2 zeigt die Statistiken der Ergebnisse dieses Tests.

Genau 15 Probanden nahmen an diesem Test teil. Von diesen 15 Probanden haben zehn den Test bestanden: Fünf Probanden mit einer Note von 12/20 und fünf Probanden mit 16/20. Diese zehn Probanden gehören der zweiten Gruppe an. Die anderen fünf Probanden, die den Test nicht bestanden haben (drei Probanden mit einer Note von 04/20 und zwei Probanden mit 08/20), sowie fünf weitere Probanden, die nicht am Test teilgenommen haben, wurden der ersten Gruppe zugeordnet.

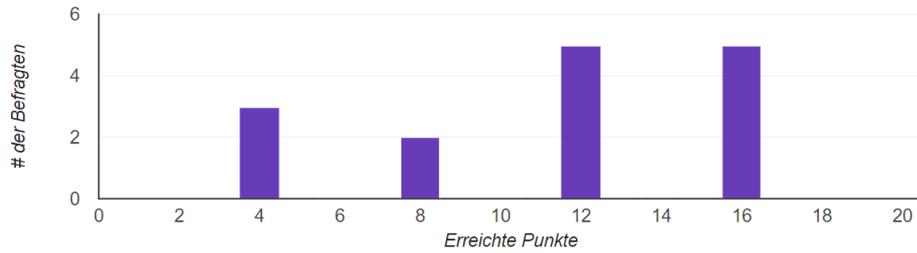
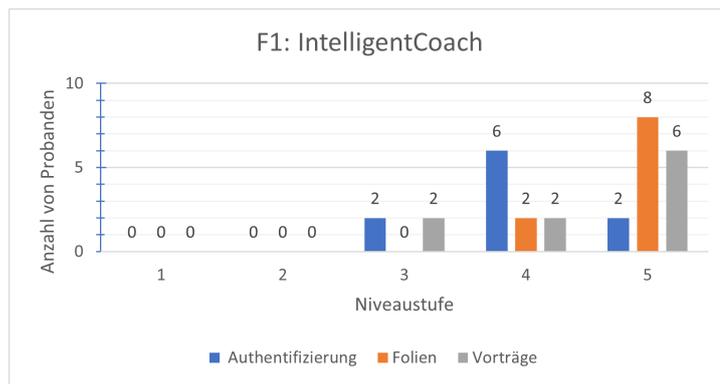


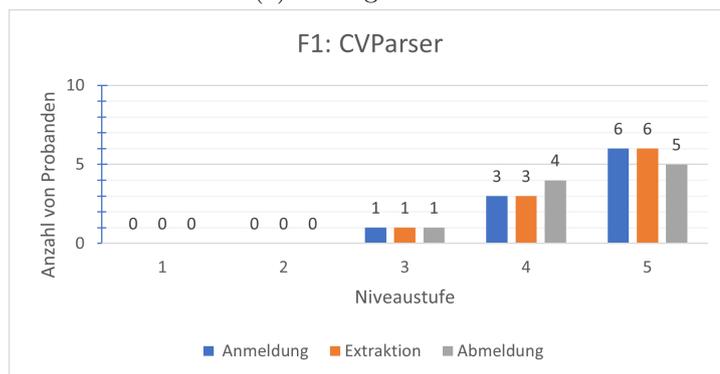
Abbildung 6.2: Verteilung der Gesamtpunktzahlen

F1-Sie haben den Test zur Gherkin Schulung bestanden. Im Vergleich zu dieser Schulung, wie gut haben Sie die in Gherkin textuellen beschriebenen Szenarien in dieser Studie verstanden? Geben Sie eine Note zwischen 1 und 5. (5 bedeutet sehr gut und 1 nicht gut)

Diese Frage wird mit einer 5-Likert-Skala bewertet.



(a) IntelligentCoach



(b) CVParser

Abbildung 6.3: Ergebnisse der erste Frage

Abbildung 6.3 zeigt die unterschiedlichen Stufen des Verständnisses in Abhängigkeit von den in der Studie behandelten Probanden und für jedes Segment (Anmeldung, Abmeldung und Extraktion) des Visionsvideo. Auf der X-Achse werden die verschiedenen Skalen und auf der Y-Achse die Anzahl der Probanden aufgelistet. Diese Frage wird in diesem Abschnitt erwähnt, da sie dazu dient, die Effektivität der Gherkin Schulung zu bestimmen.

Zusammenfassend lässt sich feststellen, dass die Probanden die in der Studie beschriebenen Szenarien gut oder sehr gut verstehen, sowohl bei dem ersten als auch dem zweiten Visionsvideo. Es gab keine Probanden, die die Gherkin-Szenarien nicht verstanden haben. Die Ergebnisse zeigen, dass die Gherkin-Schulung effektiv ist und von den Probanden verstanden wurde. Diese Effektivität der Gherkin-Szenarien wird zu einem besseren Verständnis der Anforderungen von Seiten der Probanden beitragen.

6.2 Überprüfung der Hypothesen

Bevor dieser Teil beginnt, ist es wichtig, sich die verschiedenen Hypothesen in Tabelle 6.2 unten in Erinnerung zu rufen.

Ziele 1	- $H_{0,1,1}$: Es gibt keine Anforderungen, die in den gezeigten Gherkin-Texten richtig beschrieben werden müssen. - $H_{0,2,1}$: Es gibt keine Anforderungen, die in den gezeigten Gherkin-Texten vervollständigt werden müssen.
Ziele 2	- $H_{0,2,2}$: Es gibt keinen Unterschied zwischen dem Betrachten von Visionsvideos mit Hilfe von Gherkin-Texten und dem Betrachten ausschließlich von Visionsvideos in Bezug auf die Variable Anzahl von Anforderungen

Tabelle 6.2: Hypothesen

Auf der Grundlage dieser Hypothesen werden die detaillierten Antworten der Fragebögen präsentiert. Auf diese Antworten folgt eine Interpretation der betreffenden Hypothesen. Da Alternativhypothesen das Gegenteil von Nullhypothesen sind, werden ausschließlich die Nullhypothesen betrachtet. Die Alternativhypothesen werden in der Interpretation behandelt.

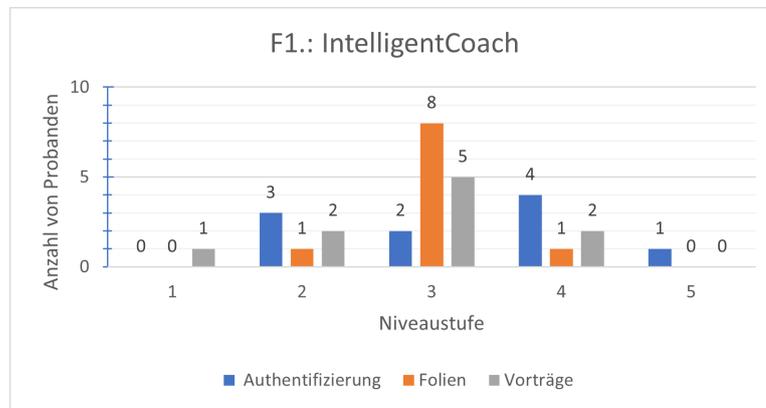
6.2.1 $H_{0,2,2}$: Es gibt keinen Unterschied zwischen dem Betrachten von Visionsvideos mit Hilfe von Gherkin-Texten und dem Betrachten ausschließlich von Visionsvideos in Bezug auf die Variable Anzahl von Anforderungen.

Diese Hypothese vergleicht die Ergebnisse der beiden Gruppen, die für die Studie erstellt wurden. Genauer gesagt sind die beiden Fragen der

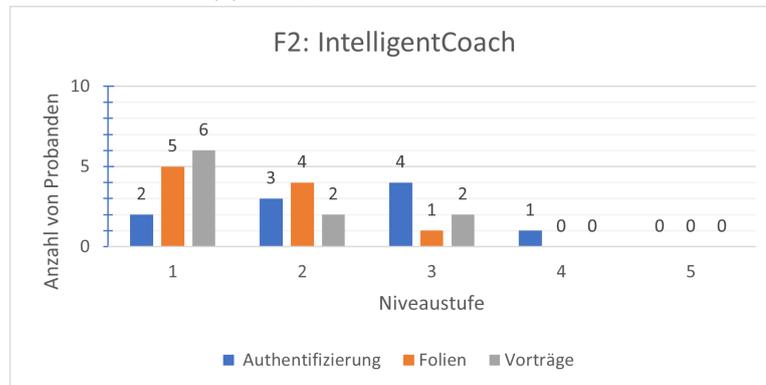
ersten Gruppe genau die gleichen wie die zweite und dritte Frage der zweiten Gruppe. Die Idee hier ist es, die Anzahl der Anforderungen in den Visionsvideos mit Hilfe von Gherkin-Texten und ohne Hilfe von Gherkin-Texten zu vergleichen. Die Antworten auf diese beiden Fragen werden in den folgenden Abschnitten ausführlich erläutert.

Frage 1: Wie schwer können Sie aus dem gerade angesehenen Video Anforderungen erkennen? Geben Sie eine Note zwischen 1 und 5. (5 bedeutet sehr schwer und 1 nicht schwer)

Diese Frage wird mit einem 5-Likert-Skala bewertet.

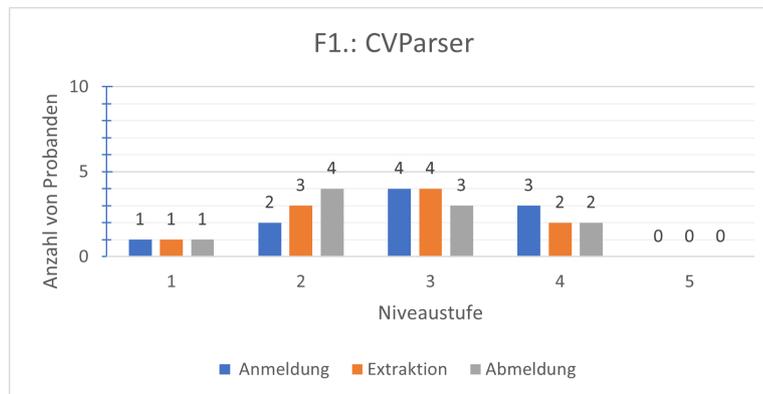


(a) Gruppe ohne Gherkin-Texte

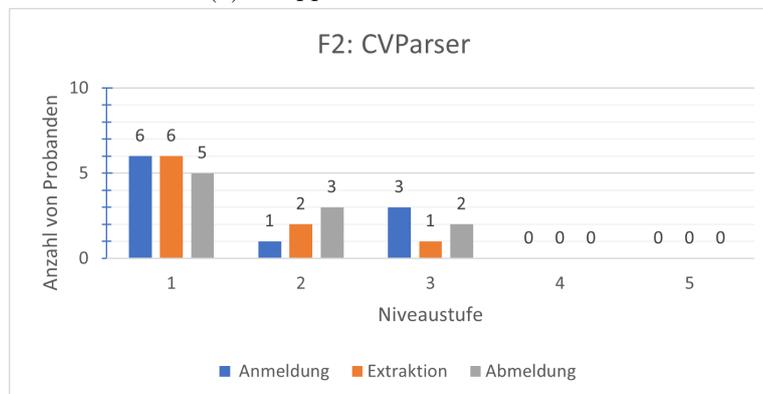


(b) Gruppe mit Gherkin-Texten

Abbildung 6.4: Ergebnisse der Frage 1 für IntelligentCoach



(a) Gruppe ohne Gherkin-Texte



(b) Gruppe mit Gherkin-Texten

Abbildung 6.5: Ergebnisse der Frage 1 für CVParse

Abbildung 6.4 und 6.5 zeigt die unterschiedlichen Stufen des Verständnisses in Abhängigkeit von den in der Studie behandelten Probanden und für jedes Segment des Visionsvideos für die Gruppe ohne Gherkin-Texten und für die Gruppe mit Gherkin-Texten. Auf der X-Achse werden die verschiedenen Skalen und auf der Y-Achse die Anzahl der Probanden aufgelistet.

Die Ergebnisse der Bewertungen für die Gruppe ohne Gherkin-Texten sind eher neutral mit ca. 4 von 10 Probanden für IntelligentCoach und 5 von 10 Probanden für CVParse in Skala 3. Das Erkennen von Anforderungen ist jedoch mit dem CVParse Visionsvideo etwas einfacher als mit dem IntelligentCoach Visionsvideo. Was die Gruppe mit Gherkin-Texten angeht, lässt sich feststellen, dass die Probanden die Anforderungen in den Visionsvideos einfach erkennen. Diese Anforderungen sind besonders in Visionsvideo CVParse deutlicher erkennbarer als in Visionsvideo IntelligentCoach. Es gibt keine Probanden, die es tatsächlich schwierig fanden, die Anforderungen zu erkennen.

Frage 2: Listen Sie diese Anforderungen auf, die Sie erkannt haben (Schreiben Sie vollständige aktive Sätze auf)

Im Prinzip werden die vom Anforderungsingenieur analysierten Anforderungen in den Visionsvideos dargestellt, um von den Kunden validiert zu werden. Daher wurden die Kunden gebeten, diese Anforderungen durch die Visionsvideos klar aufzulisten. Der Unterschied zwischen den beiden Gruppen besteht darin, dass bei der zweiten Gruppe jedes Segment in jedem Visionsvideo von den entsprechenden Szenarien und der Szenario-Outline (Gherkin-Texte) begleitet wird. Beispielsweise zeigt Tabelle 6.3 die bestätigte Anforderungen des Segmentes **Authentifizierung** des Visionsvideos IntelligentCoach.

Segment	Anforderungen
Authentifizierung	R1: Als Nutzer möchte ich mich über WEBSSO einloggen können.
	R2: Als Nutzer möchte ich mich authentifizieren können
	R3: Als Nutzer möchte ich mich registrieren können.
	R4: Als Nutzer möchte ich Daten erfassen und Bestätigungscode senden können, um einen Account zu erstellen.
	R5: Als Nutzer möchte ich Zugriff auf die Module haben können.
	R6: Als Nutzer sollte eine gute Internetverbindung notwendig sein.

Tabelle 6.3: Aufgelistete Anforderungen von IntelligentCoach

Die Anzahl der aufgelisteten Anforderungen ist für jeden Probanden in Tabelle 6.4 für die Gruppe ohne Gherkin-Texte und in Tabelle 6.5 für die Gruppe mit Gherkin-Texten dargestellt. Es sollte auch beachtet werden, dass es sich hierbei um die Summe der Ergebnisse für IntelligentCoach und der Ergebnisse für CVParse handelt. TN steht für Teilnehmer und ANZ steht für Anzahl von Anforderungen.

TN	1	2	3	4	5	6	7	8	9	10
ANZ	7	5	8	4	5	3	6	8	7	6

Tabelle 6.4: Gruppe ohne Gherkin-Texte

TN	11	12	13	14	15	16	17	18	19	20
ANZ	10	8	8	10	7	13	11	9	7	12

Tabelle 6.5: Gruppe mit Gherkin-Texten

Interpretation

In diesem kleinen Abschnitt werden die Antworten auf die beiden vorangegangenen Fragen interpretiert. Bei der ersten Frage ging es darum zu bestimmen, wie schwierig es für die Probanden ist, die Anforderungen aus dem betrachteten Visionsvideo zu erkennen. Die Antworten auf diese Frage zeigen, dass die Probanden die Anforderungen mithilfe der Gherkin-Texte deutlich einfacher verstehen. Sie können die Anforderungen daher leichter erkennen und sollten sie folglich besser auflisten. Diese Frage hebt den Unterschied beim Erkennen und Verstehen von Anforderungen hervor und ermöglicht es, mithilfe der zweiten Frage herauszufinden, ob ein besseres Verständnis zu einer größeren Anzahl von Anforderungen führt.

Zur besseren Interpretation der Ergebnisse der zweiten Frage wurden die verschiedenen Antworten mit Hilfe des Mann-Whitney-U-Tests [51, 49, 35] getestet. Der Mann-Whitney-U-Test, auch bekannt als Wilcoxon-U-Test, ist ein nicht-parametrischer Test, der zum Vergleich der Mediane zweier unabhängiger Gruppen verwendet wird. Bevor der Test durchgeführt werden kann, müssen jedoch die folgenden Voraussetzungen [21] erfüllt sein:

- **Unabhängigkeit der Messungen:** In dieser Studie enthält jede Gruppe unterschiedliche Teilnehmer. Kein Teilnehmer hat mehr als einmal an der Studie teilgenommen.
- **Die unabhängige Variable hat zwei Modalitäten und ist auf einer Nominalskala:** In dieser Studie sind die Gruppen unabhängig voneinander. Es gibt eine Gruppe mit Gherkin-Texten und eine Gruppe ohne Gherkin-Texte.
- **Die abhängige Variable ist zumindest ordinalskaliert:** Die Anzahl der aufgelisteten Anforderungen (abhängige Variable) ist ordinal skaliert und umfasst Werte von 3 bis 13.

Da die Daten bereits gesammelt wurden, ist es möglich, eine Statistiksoftware oder eine Verteilungstabelle zu verwenden, um den Mann-Whitney-U-Test zu berechnen und festzustellen, ob die beiden Gruppen statistisch unterschiedlich sind. Im Rahmen dieser Arbeit wurde die Statistiksoftware **R** [50] verwendet. R ist eine Open-Source-Software für Statistik und Datenwissenschaft. Sie wurde entwickelt, um statistische Analysen, Grafiken und Simulationen durchzuführen sowie Daten zu verwalten und zu manipulieren. Das Signifikanzniveau wird mit 0,05 angenommen.

```

> x <- c(7, 5, 8, 4, 5, 3, 6, 8, 7, 6)
> y <- c(10, 8, 8, 10, 7, 13, 11, 9, 7, 12)
> wilcox.test(x, y, alternative = "two.sided", mu = 0, paired = FALSE, exact = NULL, correct = TRUE)

      Wilcoxon rank sum test with continuity correction

data:  x and y
W = 8, p-value = 0.001553
alternative hypothesis: true location shift is not equal to 0

Warnmeldung:
In wilcox.test.default(x, y, alternative = "two.sided", mu = 0, :
  kann bei Bindungen keinen exakten p-Wert Berechnen

```

Abbildung 6.6: Mann-Whitney U Test

Abbildung 6.6 zeigt das Testergebnis für das Tool R. Hier ist ausschließlich der p-value von Interesse. Mit 0,001553 liegt er deutlich unter dem Signifikanzniveau von 0,05. Das bedeutet, dass die Nullhypothese, dass es keinen Unterschied zwischen den Gruppen gibt, abgelehnt bzw. nicht akzeptiert werden kann. Die Warnmeldung ist nachvollziehbar, da es tatsächlich Bindungen gibt. Aus diesem Grund kann der p-value lediglich geschätzt werden.

Die beiden Gruppen unterscheiden sich statistisch signifikant in Bezug auf die Anzahl der Anforderungen. Die durchschnittliche Anzahl der Anforderungen, die von der Gruppe mit den Gherkin-Texten formuliert wurde, war mit 9,5 statistisch höher als die der Gruppe ohne die Gherkin-Texte mit 5,9. Die Hypothese $H_{0,2,2}$, dass es keinen Unterschied zwischen der Anzahl der Anforderungen gibt, die beim Betrachten der Visionsvideos mit Hilfe von Gherkin-Texten gefunden wurden und der Anzahl der Anforderungen, die beim Betrachten der Visionsvideos ohne Gherkin-Texte gefunden wurden, kann nicht akzeptiert werden. Diese Hypothese kann durch die Berechnung der Metriken $M_{2,1}$ und $M_{2,2}$ bewiesen werden. Mit der Metrik $M_{2,1}$ wird die Anzahl der Anforderungen für die beiden kombinierten Visionsvideos berechnet, die von 10 Probanden mit Hilfe der Gherkin-Texte aufgelistet wurden. Die Anzahl von 95 formulierten Anforderungen wird festgestellt. Mit der Metrik $M_{2,2}$ wird dagegen die Anzahl der Anforderungen für die beiden kombinierten Visionsvideos berechnet, die von 10 Probanden ohne die Hilfe der Gherkin-Texte aufgelistet wurden. Eine Anzahl von 59 Anforderungen wird festgestellt.

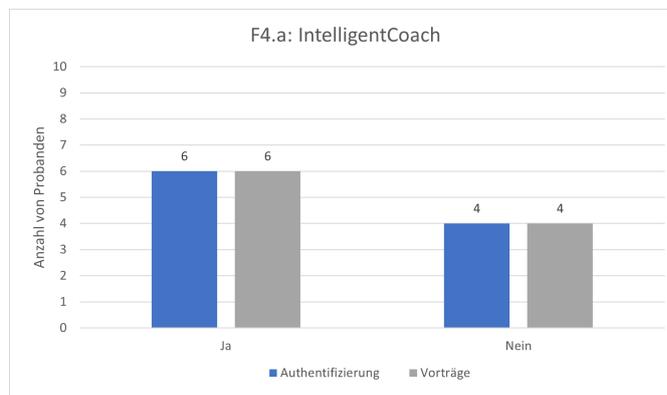
6.2.2 $H_{0,1,1}$: Es gibt keine Anforderungen, die in den gezeigten Gherkin-Texten richtig beschrieben werden müssen.

Diese Hypothese lässt erkennen, ob die Gherkin-Szenarien falsch formuliert wurden oder ob sie nicht unbedingt die von den Kunden gewünschten Anforderungen widerspiegeln. Um diese Hypothese zu überprüfen, werden die Fragen 4.a, 4.b und 4.c beantwortet, die in der Gruppe mit den Gherkin-

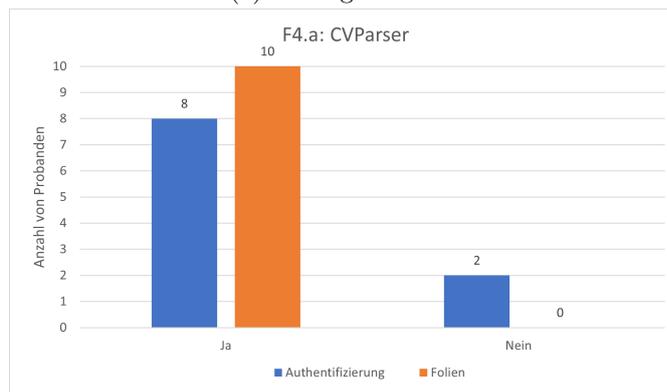
Texten gestellt wurden.

Frage 4.a: Einige Schritte in Gherkin wurden nicht aus den Visionsvideos interpretiert, sondern vom Anforderungsingenieur selbst hinzugefügt. Diese Schritte sind mit #FA markiert und beschreiben die möglichen fehlenden Angaben. Sind Sie mit diesen hinzugefügten Schritten einverstanden?

Die für diese Studie gewählten Visionsvideos enthalten ausschließlich Unklarheiten aufgrund fehlender Angaben. Diese Frage konnte lediglich mit **Ja** oder **Nein** beantwortet werden. Abbildung 6.7 zeigt die verschiedenen Antworten der Probanden für jedes Segment. Diese Antworten sind in Abbildung 6.7a für das Visionsvideo IntelligentCoach und in Abbildung 6.7b für das Visionsvideo CVParser aufgeteilt.



(a) IntelligentCoach



(b) CVParser

Abbildung 6.7: Ergebnisse der vierte Frage

Die Segmente *Folien* aus IntelligentCoach und *Abmeldung* aus CVParser werden nicht dargestellt, da sie keine Unklarheiten enthalten. Die Ergebnisse zeigen eine differenzierte Meinung zu IntelligentCoach mit einer durchschnittliche Anzahl von 6 Personen, die den vom Anforderungsingenieur erkannten Unklarheiten zustimmten, und 4 Personen, die nicht zustimmten. Bei CVParser stimmten zwischen 8 und 10 Personen den Unklarheiten zu und beispielsweise beim Segment Anmeldung waren es lediglich 2 Personen.

Frage 4.b: Wenn Nein, Erläutern Sie an welche Stelle Sie die Schritte anders interpretiert hätten

Diese Frage konzentriert sich ausschließlich auf die Probanden, die mit den Unklarheiten nicht einverstanden waren. Um diese Fragen zu beantworten, werden in Tabelle 6.6 lediglich alle der betreffenden Szenarien aufgelistet.

Visionsvideo	Segment	Szenario
IntelligentCoach	Authentifizierung Vorträge	Alle Szenarien Tutorial lernen
CVParser	Anmeldung	Anmeldung in CVParser Account erstellen

Tabelle 6.6: Szenarien, die vom Kunden nicht zugestimmt wurden

Frage 4.c: Wie würden Sie dann die Schritte anders interpretieren?

Um diese Fragen zu beantworten, werden für jedes Segment einige Bemerkungen zusammengefasst. Beim Segment Authentifizierung des Visionsvideos IntelligentCoach zum Beispiel fand eine Probandin die lange Authentifizierungsprozedur nicht notwendig. Im Fall des Segments Vorträge sprach eine Probandin an, dass eine Bewertungsplattform nicht der richtige Ort sei, um sich Tutorials anzusehen. Ihrer Meinung nach gibt es dafür andere Plattformen und Webseiten, wie StudIP in Bezug auf die Leibniz Universität Hannover.

Betrachtet wird hierzu das Segment *Anmeldung* aus dem CVParser-Visionsvideo und die beiden Personen, die mit den Unklarheiten nicht einverstanden waren: Sie können die Homepage zwar einsehen, aber nicht bearbeiten. Das ist deutlich abweichend von dem, was im Szenario beschrieben wird.

Interpretation

Zusammenfassend lässt sich feststellen, dass die Gherkin-Texte im Allgemeinen angemessen formuliert sind. Allerdings ist diese Formulierung zwar für die meisten Probanden richtig, aber nicht für alle Probanden.

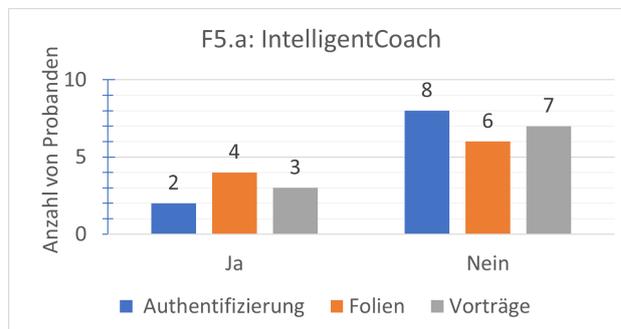
Einige Probanden fanden die Anforderungen falsch beschrieben. Diese Anforderungen müssen dann richtig beschrieben werden. Die Hypothese $H_{0,1,1}$, dass es keine Anforderungen gibt, die in den gezeigten Gherkin-Texten beschrieben werden müssen, wird daher abgelehnt. Die Alternativhypothese kann deshalb akzeptiert werden. Diese Hypothese kann durch die Berechnung der Metrik $M_{1,1}$ bewiesen werden. In dieser Metrik wird die Anzahl der Anforderungen gezählt, die in Gherkin Texten falsch beschrieben wurden berechnet. Das Ergebnis zeigt eine Anzahl von neun Anforderungen.

6.2.3 $H_{0,2,1}$: Es gibt keine Anforderungen, die in den gezeigten Gherkin-Texten vervollständigt werden müssen.

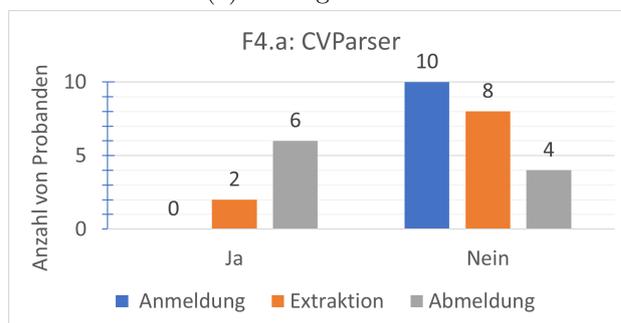
Diese Hypothese unterscheidet sich von der vorherigen Hypothese und konzentriert sich auf die Vollständigkeit der Gherkin-Texte. Die Antworten auf die Fragen 5.a und 5.b werden dabei helfen, diese Hypothese genauer zu überprüfen.

Frage 5.a: Haben Sie andere Anforderung als gegebene Gherkin-Texte?

Diese Frage konnte ebenfalls mit Ja oder Nein beantwortet werden.



(a) IntelligentCoach



(b) CVParser

Abbildung 6.8: Ergebnisse der fünften Frage

Abbildung 6.8 zeigt die verschiedenen Antworten der Probanden für jedes Segment. Diese Antworten sind in Abbildung 6.8a für das Visionsvideo IntelligentCoach und in Abbildung 6.8b für das Visionsvideo CVParser aufgeteilt. Auf der X-Achse werden die verschiedenen Antwortmöglichkeiten und auf der Y-Achse die Anzahl der Probanden aufgelistet.

Frage 5.b: Wenn ja, listen Sie diese auf. (Schreiben Sie vollständige Aktive Sätze auf)

Diese Frage betrifft ausschließlich Probanden, die bei der vorherigen Frage mit Ja geantwortet haben. Die folgende Tabelle 6.7 enthält eine Liste der zusätzlichen Anforderungen des Kunden.

VISIONSVIDEO	ANFORDERUNGEN
IntelligentCoach	R1: Als Nutzer möchte ich eine Rückmeldung bei falschen Anmeldedaten erhalten können.
	R2: Als Nutzer, wenn ich mich anmelden kann, muss ich mich auch abmelden können.
	R3: Das System muss bestimmte Dateiformate unterstützen z.B. PowerPoint, PDF oder Keynote.
	R4: Als externer Nutzer, der nicht an der Uni ist und über keinen WebSSO Account verfügt, möchte ich bewerten können.
CVParser	R1: Als Nutzer sollte ich auch einfach den Link der LinkedIn Seite kopieren können.
	R2: Der Lebenslauf sollte vom System gespeichert werden können, sodass dieser bei erneuter Anmeldung wieder abgerufen werden kann.
	R3: Als Nutzer möchte ich mein Passwort zurücksetzen können.
	R4: Wenn ich mich anmelde, möchte ich als Nutzer die E-Mail bestätigen, sodass niemand anderes in meinen Namen registriert. Wenn ich mich anmelde, möchte ich als Nutzer die E-Mail bestätigen, sodass ich einen Tippfehler in der Registrierung bemerke.
	R5: Als Nutzer kann ich auch durch Kopieren meines XING-Profiles den Lebenslauf generieren.

Tabelle 6.7: Zusatzanforderungen von Kunden

Interpretation

Insgesamt lässt sich zusammenfassen, dass die Gherkin-Texte fast alle möglichen Anforderungen an Visionsvideos abgedeckt haben. Wie die

obigen Abbildungen zeigen, hatten die meisten Probanden keine weiteren Erklärungsmöglichkeiten hinzuzufügen. Trotzdem gab es eine kleine Anzahl von Probanden, die Anforderungen in den gezeigten Gherkin-Texten nicht vollständig beschrieben fanden. Diese Anforderungen müssen vervollständigt werden. Die Nullhypothese $H_{0,2,1}$, dass es keine Anforderungen gibt, die in den gezeigten Gherkin-Texten vervollständigen werden müssen kann verworfen werden. Dies führt dazu, dass die Alternativhypothese $H_{1,2,1}$, die in diesem Abschnitt interpretiert wird, tatsächlich akzeptiert wurde. Diese Hypothese kann durch die Berechnung der Metrik $M_{1,2}$ bewiesen werden. In dieser Metrik wird die Anzahl der Anforderungen gezählt, die in Gherkin Texte unvollständig beschrieben wurden berechnet. Das Ergebnis zeigt eine Anzahl von zehn Anforderungen.

Kapitel 7

Diskussionen

In diesem Kapitel wird zunächst über die detaillierten Antworten auf die verschiedenen Forschungsfragen und die formulierten Metriken diskutiert. Anschließend werden die verschiedenen Grenzen vorgestellt, die bei der Durchführung dieser Arbeit aufgetreten sind. Abschließend wird eine allgemeine Diskussion über einige Punkte geführt.

7.1 Antwort auf die Forschungsfragen

Bevor dieser Teil beginnt, ist es wichtig, sich die verschiedenen Forschungsfragen unten in Erinnerung zu rufen:

- **Forschungsfrage 1:** Wie können Anforderungsingenieur mit Kunden gefundenen Missverständnisse in der Anforderungen aus Visionsvideos mit Hilfe von Gherkin-Texten klären?
- **Forschungsfrage 2:** Wie unterscheidet sich die Anzahl der Anforderungen bei der Betrachtung von Visionsvideos mit Hilfe von Gherkin-Texten im Vergleich zu Visionsvideos ohne Gherkin-Texte?

Die Antwortvorschläge auf diese Fragen werden in den folgenden zwei Absätzen vorgestellt.

Forschungsfrage 1:

Wenn die Hypothesen $H_{0,1,1}$ und $H_{0,2,1}$ unter der Berechnung der Metriken $M_{1,1}$ und $M_{1,2}$ des vorherigen Kapitels berücksichtigt werden, können die Anforderungen missverstanden werden. Das heißt, die Anforderungen sind in den Gherkin-Texten nicht vollständig oder falsch beschrieben. Das Missverständnis entsteht aufgrund von Unklarheiten in den Anforderungen. Dies führte zu der Tatsache, dass die Anforderungen, die ursprünglich vom Anforderungsingenieur abgeleitet und beschrieben wurden, nicht mit den Anforderungen der Kunden übereinstimmten. Die Studie zeigte daher,

dass es eine kleine, aber hochrelevante Anzahl von Anforderungen gab, die von den Probanden interpretiert werden sollten. Die Probanden waren mit den Änderungen, die der Anforderungsingenieur aufgrund dieser Unklarheit an den Gherkin-Texten vornahm, nicht einverstanden. Sie konnten diese Anforderungen mit Hilfe der Fragen 4.a, 4.b, 4.c 5.a und 5.b der Studie klären. Darüber hinaus hatten sie noch weitere Anforderungen aufgelistet, die nicht in den Gherkins-Texten beschrieben waren.

Forschungsfrage 2:

Hierbei geht es um die Frage, wie sich die Anzahl der Anforderungen in den einzelnen Gruppen der Studie unterscheidet. Dabei wird die Hypothese $H_{0,2,2}$ unter der Berechnung der Metriken $M_{2,1}$ und $M_{2,2}$ des vorherigen Kapitels berücksichtigt. Diese Forschungsfrage ist so zu verstehen, dass der Beitrag der Gherkin-Texte entweder eine größere, eine kleinere oder die gleiche Anzahl von Anforderungen ergibt. Der Mann–Whitney U test hat bestätigt, dass die Anzahl der Anforderungen, die beim Betrachten von Visionsvideos mit Hilfe von Gherkin-Texten gefunden wurden, sich von der Anzahl der Anforderungen unterscheidet, die beim Betrachten von Visionsvideos ohne Gherkin-Texte gefunden wurden. Im Durchschnitt hat darüber hinaus die Gruppe mit den Gherkin-Texten statistisch gesehen mehr Anforderungen (9.5) als die Gruppe ohne Gherkin-Texte (5.9).

Um eine mögliche Antwort zu geben, könnte argumentiert werden, dass sich die Anzahl der Anforderungen in jeder Gruppe nicht allein auf der Ebene des Mann–Whitney U test, sondern auch auf der Ebene des Durchschnitts der Anzahl der Anforderungen unterscheidet.

7.2 Formulierung von Anforderungen

7.2.1 Einfluss vom Betrachten von Gherkin-Texten

Um über Einfluss auf das Betrachten von Gherkin-Texten auf die Formulierung von Anforderungen besser diskutieren zu können, wird zuerst eine kurze Zusammenfassung des Begriffs BDD [70] gegeben. Die verhaltensorientierte Programmierung (engl. behaviour-driven development oder BDD) ist eine agile Programmiermethode, die die Zusammenarbeit zwischen Entwicklern, Qualitätsingenieuren und nicht-technischen oder kommerziellen Beteiligten an einem Softwareprojekt (Kunde, Anforderungsingenieur) mit dem Format Gherkin fördert. Gherkin [78] ermöglicht eine textuelle Beschreibung von Anforderungen in Form von Szenarien und ist eine für alle Beteiligten leicht nachvollziehbare Beschreibungssprache.

In dieser Studie wird versucht, eine einfache Zusammenarbeit zwischen dem Anforderungsingenieur und dem Kunden durch Visionsvideos zu ermöglichen. Aus diesem Grund wird bei der Beschreibung der Anforderungen

in den Visionsvideos die Gherkin-Syntax verwendet, um das Verständnis für den Kunden zu erleichtern. Den Ergebnissen der Studie zufolge unterstützten die Gherkin-Texte die Kunden dabei, die Funktionen und Anforderungen der Visionsvideos zu verstehen und sie schließlich formulieren zu können. Die Schritte (Given, When, Dann) helfen zu verstehen, wie die Anforderung auf das Programm angewendet wird. In diesem Fall wird ein Kunde präzise erfahren, was er mit der Software unternehmen kann, um die von ihm formulierten Anforderungen zu erfüllen.

7.2.2 Einfluss von User story

Die User Story [71] hat eine entscheidende Rolle bei der Formulierung der Anforderungen gespielt. Während der gesamten Studie wurde von den Probanden gefordert, die Anforderungen in Form von User Stories zu formulieren. Erstens, weil sie es ermöglicht, den Inhalt einer zu entwickelnden Funktionalität mit ausreichender Genauigkeit zu definieren [13]. Sie ist schnell zu erstellen und erleichtert die Einschätzung des Aufwands, der für ihre Umsetzung erforderlich ist.

Zweitens ermöglichen die User Storys eine leichte Anpassung an agile Frameworks [76]. Framework wie Jira Software [46] ermöglicht diese Anpassung.

Drittens ermöglicht die Formulierung von Anforderungen in User Stories eine Klassifizierung der Anforderungen hinsichtlich der Wahl der Anforderungsgranularität. In Kapitel 4 dieser Arbeit wurde die Auswahl der Granularität auf feingranulare Anforderungen beschränkt. Alle Anforderungen, die vom Anforderungsingenieur aus den Szenarien extrahiert wurden, waren vom Typ feingranular. Da die User Story auch aus feingranularen Anforderungen besteht, kann sie auch nach der Formulierung und Klärung der Anforderungen die gleiche Granularität bewahren.

Die Ergebnisse dieser Studie würden sicherlich stark beeinflusst, wenn die Probanden die Anforderungen in einfachen Sätzen formulieren würden. Zunächst zur Granularität: Es ist möglich, dass am Ende Anforderungen mit unterschiedlicher Granularität vorliegen. In diesem Fall könnte die Anzahl der Anforderungen nicht mehr gezählt und/oder klassifiziert werden. Zweitens in Bezug auf die Anforderungsbeschreibung: Wenn Anforderungen nicht detailliert beschrieben sind, muss der Anforderungsingenieur die Interpretationsphase erneut durchlaufen, um klare und prägnante Anforderungen zu erhalten.

7.2.3 Einfluss der Anzahl der Anforderung

Bei der Auflistung der Forderungen kam es vor, dass eine, zwei oder mehrere Probanden die gleiche(n) Anforderung(en) formulierten. Angesichts der Tatsache, dass die Betrachtung der Visionsvideos asynchron erfolgt

und eine Probandin nicht wissen darf, welche Anforderung eine andere formuliert hat, war es vorstellbar, dass Anforderungen ein bis zehn Mal auftauchten. In diesem Fall wurden die identischen Anforderungen zu einer einzigen Anforderung zusammengefasst, um zu vermeiden, dass die gleiche Anforderung bei der Entwicklung der betreffenden Software mehrmals wiederholt werden muss. Im Rahmen dieser Studie wurde hingegen die genaue Anzahl der Anforderungen für jedes Proband unabhängig von den Anforderungen berücksichtigt, die von anderen Probanden formuliert wurden.

Anhang E.1 enthält eine Liste aller Anforderungen, die von den Kunden für jedes Segment des entsprechenden Visionsvideos aufgelistet wurden. Anhang E.1.1 bezieht sich auf Visionsvideo IntelligentCoach und Anhang E.1.2 auf Visionsvideo CVParser. Anhang E.2 enthält eine Liste aller Anforderungen, die von den Kunden für jedes Segment des Visionsvideos zusammen mit den entsprechenden Gherkin-Texten aufgelistet wurden. Anhang E.2.2 steht für Visionsvideo IntelligentCoach und Anhang E.2.1 für Visionsvideo CVParser. Aus diesem Grund ergibt sich zum Beispiel eine Gesamtzahl von 92 formulierten Anforderungen, aber lediglich 33 Anforderungen in Anhang E.1.1.

7.3 Begrenzung der Arbeit

7.3.1 Requirement Engineering

Die wichtigste Einschränkung dieser Arbeit betrifft das Referenzmodell. Grundsätzlich besteht die Idee darin, dass der Anforderungsingenieur die Kundenanforderungen ermittelt, bei Bedarf interpretiert, priorisiert, dokumentiert und schließlich durch Kundvalidierung, beispielsweise mit Hilfe von Visionsvideos, abschließt. In dieser Arbeit stehen jedoch ausschließlich Visionsvideos zur Verfügung, ohne Kontakt zu den Erstellern der Videos oder den Kunden, die die ursprünglichen Anforderungen gestellt haben. Während der Studie wurden andere Kunden zur Validierung verwendet, die nichts mit den ursprünglichen Kunden zu tun hatten. Einige dieser anderen Kunden äußerten Zweifel, ob sie die gleichen Anforderungen gestellt hätten und sahen das Tool aus einer anderen Perspektive. Daher sollten die Kunden in der Studie nicht ihre eigenen Anforderungen validieren, sondern die Anforderungen der ursprünglichen Kunden, die die Grundlage für die Erstellung der Visionsvideos bildeten.

7.3.2 Visionsvideo

Die in dieser Arbeit verwendeten Visionsvideos entsprechen nicht allen Richtlinien, die für die Produktion von Visionsvideos festgelegt wurden. Zu diesem Zweck definierten Karras et al. [38] einen interdisziplinären Leitfaden

für die Produktion von Visionsvideos durch Softwareexperten. Sie [38] haben sich mit dem Mangel an Wissen und Fähigkeiten von Softwareprofis beschäftigt und zwei Richtlinien bereitgestellt, die als Checklisten zur Vermeidung häufiger Fehler während der Herstellung und Nutzung von Videos verwendet werden können. Die verwendeten Visionsvideos haben keinen angemessenen Übergang, die Software ist nicht deutlich zu erkennen und die Texte sind ungenau. Die Visionsvideos sind bei einigen extrem kurz und bei anderen besonders lang. Es ist sehr wahrscheinlich, dass die Kunden die in den Visionvideos gezeigten Funktionen nicht verstehen, weil sie die Guidelines nicht erfüllen.

Es ist ebenfalls schwierig, spezifische Details und Eigenschaften der Anforderungen aus den Visionsvideos abzuleiten, da sie in der Regel ausschließlich einen Überblick über die Funktionalität des zu entwickelnden Systems geben. Die Informationen über die Qualität, Robustheit oder Leistung des Systems werden oft implizit vermittelt und können leicht übersehen werden. Daher ist es wichtig, bei der Verwendung von Visionsvideos als Validierungstool zu berücksichtigen, dass diese Visionsvideos für die Abdeckung aller notwendigen Anforderungen möglicherweise nicht ausreichen und weitere Informationen für die Erstellung einer vollständigen Anforderungsspezifikation benötigt werden.

7.3.3 Textuelle Beschreibung von Anforderungen

Es sollte jedoch beachtet werden, dass die Verwendung von Szenarien zur Beschreibung von Anforderungen im Rahmen von BDD [70] eine gängige Praxis in der Industrie ist. Die Verwendung von Szenarien als Anforderungsbeschreibungen ermöglicht es, Anforderungen in einer leicht verständlichen und verifizierbaren Weise zu beschreiben und somit Missverständnisse und Fehler zu minimieren. Es ist auch erwähnenswert, dass die Verwendung von Szenarien nicht auf BDD beschränkt ist und in anderen Ansätzen zur Anforderungsspezifikation wie zum Beispiel Use Case-basierten Ansätzen eingesetzt wird.

7.3.4 Benutzerstudie

Als Limitationen der Studie sind die Schwierigkeiten zu erwähnen, die bei der Verwendung des Tools Vivi Player auftraten. Diese beinhalteten Probleme mit der Browserkompatibilität, dem begrenzten Platz für die Videowiedergabe sowie der Notwendigkeit, das Tool an die Erfordernisse der Studie anzupassen. Eine weitere Herausforderung war die Dauer der Studie. Aufgrund der Qualität des Visionvideos mussten die Probanden es zweimal ansehen, was zu einer längeren Dauer der Studie führte. Trotzdem hatten einige Probanden Schwierigkeiten, die Anforderungen zu verstehen und kopierten stattdessen die Szenarien und formulierten sie als User Story.

Auch die Rekrutierung von Probanden war eine Herausforderung, da einige von ihnen keine IT-Kenntnisse hatten und daher nicht mit dem Begriff der Anforderung vertraut waren.

Kapitel 8

Zusammenfassung und Ausblick

In diesem Kapitel werden die wichtigsten Ergebnisse und weiterer bzw. zukünftiger Forschungsbedarf dieser Arbeit vorgestellt

8.1 Zusammenfassung

In dieser Arbeit wurde darauf hingewiesen, dass die von Anforderungsingenieuren erstellten Visionsvideos implizit die Anforderungen darstellen, die von Kunden nicht verstanden werden und von Testern nicht als Testfälle genutzt werden können. Das Ziel der vorliegenden Arbeit war daher, Visionsvideos in einem agilen Kontext als Mittel zur Anforderungserhebung zu verwenden und im Falle von Unklarheiten zu interpretieren. Hierzu wurden die Visionsvideos mit Hilfe einer einfachen Sprache, Gherkin, textuell beschrieben, und aus diesen Beschreibungen wurden die Anforderungen abgeleitet und in vage und testbare Anforderungen unterteilt.

Eine Studie mit potenziellen Nutzern des Produkts, wobei die Anforderungen in den Visionsvideos visualisiert wurden, ergab, dass die zusätzliche Betrachtung der Gherkin-Texte zu einer höheren Anzahl von formulierten Anforderungen führte als die Betrachtung der Visionsvideos allein. Darüber hinaus konnten auch missverstandene Anforderungen interpretiert werden. Dieser Ansatz ermöglicht eine verbesserte Kommunikation zwischen dem Anforderungsingenieur, dem Kunden und dem Tester und erleichtert die Beschreibung des Verhaltens von Softwarefunktionen für alle Stakeholder.

8.2 Ausblick

Die grundlegende Idee dieser Arbeit bestand darin, Anforderungen zu erheben und durch Visionvideos zu validieren. Da jedoch Unklarheiten bei der Erhebung der Anforderungen auftraten, wurden sie interpretiert. Das

Problem der Validierung bleibt ungelöst. Eine Erweiterung dieser Arbeit könnte die Berücksichtigung nicht-funktionaler Anforderungen und des Anforderungsmanagements im Referenzmodell des Requirements Engineering beinhalten. Diese Arbeit konzentrierte sich auf die Anforderungsanalyse und könnte erweitert werden, um das Anforderungsmanagement zu umfassen.

Außerdem wurde diese Arbeit mit amateurhaften Visionvideos durchgeführt. Diese Visionsvideos entsprechen nicht allen Richtlinien, die für die Produktion von Visionsvideos von Karras et al. [38] festgelegt wurden. Es wäre interessant zu untersuchen, ob die gleichen Ergebnisse auch mit professionellen Visionvideos zu erwarten sind. Die Ergebnisse dieser Arbeit könnten auch auf andere Problembereiche wie die Programmierung angewendet werden. Programmierer können Texte als solide Grundlage für ihre Testverfahren verwenden, sei es in BDDs oder in der testgetriebenen Entwicklung (TDD)[6]. Eine zukünftige Aufgabe für Programmierer könnte darin bestehen, die im Visionvideo visualisierte Software zu entwickeln und sie mit Hilfe der Gherkin-Texte zu testen.

Anhang A

A.1 Inhalt der CD

1. Die Masterarbeit in digitaler Form (als PDF-Datei)
2. Die Masterarbeit in Latex Form (als Zip-Datei)
3. Der Wöchentliche Plan (als Excel-Datei)
4. Ein Ordner mit alle Feature-Dateien pro Visionsvideos
5. Unterschiedliche Studienmaterialien:
 - Aufgabenstellung zur Studie (als PDF-Datei)
 - Die Gherkin-Schulung mit Test (als PDF-Datei)
 - Die Ergebnisse der Schulung(als CSV-Datei)
 - Ein Ordner mit mehreren ODT-Dateien mit den Antworten der Probanden

A.2 Kurze Beschreibung der Visionsvideos

A.2.1 „CVparser“

Das Visionsvideo „**CVparser**“ im Projekt soll ein Tool entwickelt werden, das Lebensläufe von Mitarbeitern eines Unternehmens in eine einheitliche und verkürzte Struktur bringt. Dabei sollen die Lebensläufe von der Seite `linkedin.com` benutzt werden.

A.2.2 „Intelligent Coach“

Im Visionsvideo „**Intelligent Coach**“ wird eine Webanwendung entwickelt, die als Online-Coach für Studenten der Leibniz Universität Hannover (LUH) agieren soll. Der Online-Coach soll unterschiedliche Möglichkeiten anbieten, Fähigkeiten, die zwar im Studium gebraucht, aber nicht ausführlich gelehrt werden, zu erlernen und zu üben. Spezifisch wird die Webanwendung für die

Verbesserung der Präsentationsfähigkeiten und das Erhalten von Feedback verwendet.

A.2.3 „iKuhlschrank“

Im Visionsvideo „**iKuhlschrank**“ geht es um einen intelligenten Kühlschrank in Verbindung mit einer mobile App für die Verwaltung von Lebensmitteln in der Kühlschrank.

A.2.4 „MobileMeetingMaster“

Im Visionsvideo „**MobileMeetingMaster**“ wird eine Mobileanwendung entwickelt, die als Transkripte von aufgenommenen Gespräche von Mitarbeiter im Meeting verwendet soll, damit vorprogrammierte Rückenschmerzen in einer Welt voller sitzender Tätigkeiten vermeiden werden können.

A.2.5 „Watertracer“

Im Visionsvideo „**Watertracer**“ soll eine mobile Anwendung entwickelt werden, die zusammen mit der Tracerglas bei der Kontrolle der Wasserkonsum eines Mitarbeiters in Unternehmen hilft, so dass Jeden Tag ausreichende Wasser getrunken wird.

A.3 Umfrage

In dieser kleinen Umfrage geht es darum, kurze Beschreibungen von fünf Tools zu lesen und anhand dieser zu entscheiden, ob Sie an dem jeweiligen Tool Interesse haben oder nicht.

Wollen Sie "iKühlschrank" verwenden?

„iKühlschrank“ beschreibt einen intelligenten Kühlschrank in Verbindung mit einer mobilen App für die Verwaltung von Lebensmitteln im Kühlschrank.

Ja

Nein

Wollen Sie "Watertracer" verwenden?

„Watertracer“ ist eine mobile App, die zusammen mit einem „Tracerglas“ bei der Kontrolle des Wasserkonsums der Mitarbeiter in Unternehmen hilft, so dass jeden Tag ausreichend Wasser getrunken wird.

Ja

Nein

Wollen Sie „MobileMeetingMaster“ verwenden?

„MobileMeetingMaster“ ist eine App, die Transkripte von aufgenommenen Gesprächen beim Laufen von Mitarbeitern im mobilen Meeting verwenden soll, damit Rückenschmerzen bei sitzenden Tätigkeiten vermieden werden können.

Ja

Nein

Wollen Sie „CVparser“ verwenden?

„CVparser“ ist ein Tool, das Lebensläufe von Mitarbeitern eines Unternehmens in eine einheitliche und verkürzte Struktur bringt. Dabei sollen die Lebensläufe von der Seite linkedin.com benutzt werden

Ja

Nein

Wollen Sie „Intelligent Coach“ verwenden?

„Intelligent Coach“ ist eine Webanwendung, die als Online-Coach für Studierende unterschiedliche Möglichkeiten für die Verbesserung der Präsentationsfähigkeiten und das Erhalten von Feedback anbieten soll.

Ja

Nein

Möchten Sie an einer weiteren Studie zum Thema Visionsvideos zu diesen Produkten teilnehmen? *

Für meine Masterarbeit werde ich eine weitere Studie durchführen, in der die oben genannten Produkte in kurzen Visionsvideos vorgestellt werden. Der Zeitaufwand wird ca. 30 Min. betragen. Ihre Teilnahme leistet einen wichtigen Beitrag, um eine ausreichend große Stichprobenmenge zu generieren. Vielen Dank!

Ja

Nein

Wenn Sie Interesse daran haben, an dieser Studie ebenfalls teilzunehmen, geben Sie bitte Ihre E-mail-Adresse an.

Selbstverständlich wird Ihre E-mail-Adresse nur im Rahmen dieser Studie verwendet und anschließend gelöscht.

@ |

Einverständniserklärung *

Hiermit erteile ich meine Einwilligung, dass meine Daten und die angegebenen Informationen ausschließlich im Rahmen dieser Studie verwendet werden dürfen.

Ja

Nein

Anhang B

Feature-Dateien mit Szenarien pro Visionsvideos

B.1 iKülschrank

1. Synchronisation

```
1 #Language: de
2 Feature: Synchronisation
3 Als Nutzer
4 moechte ich mich anmelden koennen
5 So dass ich die Geraete synchronisieren kann
6
7 Szenarien: Account erstellen #Fehlende Angaben
8 Gegeben sei, <Tools> ist funktionsfaehig #Fehlende Angaben
9 wenn ich auf den "create account" Link klicke #Fehlende
  Angaben
10 Dann kann ich meine <Daten> eingeben #Fehlende Angaben
11 Wenn ich auf den "create" Button klicke #Fehlende Angaben
12 Dann bestaetige ich meinen neuen Account #Fehlende Angaben
13
14 Beispiele: #Fehlende Angaben
15 | Tools | Daten |
16 | Internetverbindung | Name |
17 | iKuehlschrank | PIN |
18 | Smartphone | Repeat PIN |
19
20 Szenarien: Anmeldung in ikuehlschrank
21 Gegeben sei, ich habe einen Account erstellt
22 Wenn ich mich anzumelden versuche
23 Und ich meine <Anmeldung> eintrage #Eigene Interpretation
24 Dann bin ich angemeldet
25
26 Beispiele: #Eigene Interpretation
27 | Anmeldung |
28 | Name |
29 | PIN |
```

76 ANHANG B. FEATURE-DATEIEN MIT SZENARIEN PRO VISIONSVIDEOS

```
30
31 Szenarien: Synchronisierung der Geraete
32 Gegeben sei, ich bin angemeldet
33 Wenn ich das <Verbindung> fuer meine <Geraete> aktiviere
    #Fehlende Angaben
34 Und ich die <Geraete> kopple #Fehlende Angaben
35 Dann werden die Geraete synchronisiert
36 Und kann ich die viele Funktionen auch unterwegs nutzen
37
38 Beispiele:
39 | Verbindung | Geraete |
40 | Bleuthout | Smartphone |
41 | NFC | IKuehlschrank |
42 | Wifi | |
```

2. Inventur

```
1 #Language: de
2 Feature: Inventur
3 Als Nutzer
4 moechte ich meinen Lebensmitteln erfassen
5 So dass ich daraus ein Inventar machen kann
6
7 Szenarien: Kamera und Sensor aktivieren #eigene
    Interpretationen
8 Gegeben sei, ich bin angemeldet #eigene Interpretationen
9 Wenn ich das <Geraet> aktiviere #eigene Interpretationen
10 Dann ist mein <Geraet> aktiviert #eigene Interpretationen
11
12 Beispiele: #eigene Interpretationen
13 | Geraete |
14 | Kamera |
15 | Sensor |
16
17 Szenarien: Lebensmitteln automatisch erfassen
18 Gegeben sei, Kamera und Sensoren sind bereit aktiviert
19 Wenn ich mein Geraet mit meinem Smartphone synchronisiere
20 Dann erfassen Sensor und Kamera automatisch die <Angaben> den
    Lebensmitteln
21
22 Beispiele:
23 | Angaben |
24 | Name |
25 | Menge |
26 | Mindesthaltbarkeit |
```

3. Einkaufslisten

```
1 #Language: de
2 Feature: Einkaufskisten
3 Als Nutzer
```

```

4 moechte ich mit einer Liste einkaufen gehen
5 So dass ich keine Produkte mehr vergessen kann
6
7 Szenarien: Automatische Erstellung von Einkaufslisten
8 Gegeben sei, Die Lebensmitteln sind automatisch erfasst
9   Und die Dauer bis Ablauf der Lebensmittel ist ermittelt
10  Und Optische Sensoren und KI Tools sind eingesetzt
11 Wenn <Detail> beruecksichtigt ist
12 Dann erstellt der iKuehlschrank automatisch eine
   Einkaufsliste
13 Und kann ich mit dieser Liste einkaufen gehen
14
15 Beispiele:
16   | Detail |
17   | Fullstand der iKuehlschrank |
18   | Konsumverhalten |
19   | Das Einkaufen |
20
21
22 Szenario: Aktualisierung von Einkaufslisten #Eigene
   Interpretationen
23 Gegeben sei, Die Einkaufsliste ist automatisch erstellt #
   Eigene Interpretationen
24 Wenn neue Produkte in den Kuehlschrank gelegt werden #
   Eigene Interpretationen
25 Dann aktualisiert sich die Einkaufsliste automatisch #Eigene
   Interpretationen
26 Und kann ich mit dieser Liste einkaufen gehen #Eigene
   Interpretationen

```

4. Ablaufdatum

```

1 #Language: de
2 Feature: Ablaufdatum
3 Als Nutzer
4 moechte ich ein Produkt vor seinem Ablaufdatum benutzen
5 So dass ich das Essen nicht mehr wegwerfen kann
6
7 Szenario: Bestimmung der Haltbarkeit
8 Gegeben sei, Das Produkt ist schon gekauft
9   Wenn der Ablaufdatum vorbei ist
10  Dann ist mein Produkt abgelaufen
11  Wenn der Ablaufdatum nicht vorbei ist
12  Dann ist ein Produkt nicht abgelaufen
13
14 Szenario: Ermittlung der Dauer bis Ablauf
15 Gegeben sei, die Haltbarkeit der Zutaten von den Rezepten kann
   bestimmt werden
16 Wenn die Produkte verpackt sind
17 Dann wird die Mindesthaltbarkeit erkannt
18 Wenn die Produkte unverpackt sind
19 Dann bestimmt der iKuehlschrank die voraussichtliche
   Haltbarkeit

```

78 ANHANG B. FEATURE-DATEIEN MIT SZENARIEN PRO VISIONSVIDEOS

```
20
21 Szenario: Benachrichtigung beim Ablauf eines Produkts
22 Gegeben sei, die Haltbarkeit ist bestimmt
23 Und die Dauer bis Ablauf der Lebensmitteln ist ermittelt
24 Wenn ein Produkt bald abgelaufen ist
25 Dann bekomme ich eine Benachrichtigung
```

5. Rezeptideen

```
1 #Language: de
2 Feature: Rezeptideen
3 Als Nutzer
4 moechte ich Rezeptideen basierend auf den vorhandenen
  Lebensmitteln suchen
5 So dass ich gute Rezepte kochen kann
6
7 Szenario: Passende Rezepte in Plattform vom iKoch
8 Gegeben sei, es gibt eine Plattform von iKoch
9 Wenn es passende Rezepte gibt
10 Dann kann ich die Rezepte blaetterten
11
12 Szenario: Auswahl eines Rezepts
13 Gegeben sei, es gibt passende Rezepte
14 Wenn ein Rezept mir gefaellt # Eigene Interpretation
15 Dann kann ich das Rezept auswaehlen.
16
17 Szenario: Fehlende Rezepte
18 Gegeben sei, ich kann ein Rezept auswaehlen
19 Wenn es einige Zutaten im Rezept fehlt
20 Dann werden die fehlenden Zutaten in einer Einkaufsliste
  automatisch hinzugefuegt
```

B.2 Watertracer

1. Verbindung

```
1 #Language: de
2 Feature: Verbindung
3 Als Nutzer
4 moechte ich das Smartphone mit dem Tracer Glas verbinden
5 So dass ich die Geraete zusammen benutzen kann
6
7 Szenarien: Account erstellen #Fehlende Angaben
8 Gegeben sei, <Tools> ist funktionsfaehig #Fehlende Angaben
9 Und die App ist installiert
10 Wenn ich auf den "create account" Button klicke #Fehlende
  Angaben
11 Dann kann ich meine <Daten> eingeben #Fehlende Angaben
12 Wenn ich auf den "create" Button klicke #Fehlende Angaben
13 Dann bestaetige ich meinen neuen Account #Fehlende Angaben
14
```

```

15 Beispiele: #Fehlende Angaben
16
17 | Tools | Daten |
18 | Internetverbindung | Name |
19 | Smartphone | PIN |
20 | Watertracer | Repeat PIN |
21
22 Szenarien: Anmeldung in Watertracer #Fehlende Angaben
23 Gegeben sei, ich habe einen Account erstellt #Fehlende
    Angaben
24 Wenn ich mich anzumelden versuche #Fehlende Angaben
25 Und ich meine <Anmeldedaten> eintrage #Fehlende Angaben
26 Dann bin ich angemeldet #Fehlende Angaben
27
28 Beispiele: #Fehlende Angaben
29 | Anmeldedaten |
30 | Name |
31 | PIN |
32
33 Szenarien: Automatische Verbindung
34 Gegeben sei, ich bin auf die App angemeldet
35 Wenn ich das Tracerglas einschalte #Eigene Interpretation
36 Dann verbindet sich das Tracerglas mit der App per <Verbindung
    > automatisch #Fehlende Angabe
37
38 Beispiele:
39 | Verbindung |
40 | Bleuthout |
41 | NFC |
42 | Wifi |

```

2. Übertragung

```

1 #Language: de
2 Feature: Uebertragung
3 Als Nutzer
4 moechte ich Daten uebertragen die koennen
5 So dass ich die Geraete synchronisieren kann
6
7
8 Szenario: Uebertragung der Menge der getrunkenen Wasser
9 Gegeben sei, ich habe Wasser getrunken
10 Wenn das Glas mit dem Smartphone automatisch verbunden ist
11 Dann uebertraegt das Glas die Menge der getrunkenen Wasser an
    die App
12 Und wird die getrunkene Menge auf der App angezeigt

```

3. Erinnerung

```

1 #Language: de
2 Feature: Erinnerung

```

80 ANHANG B. FEATURE-DATEIEN MIT SZENARIEN PRO VISIONSVIDEOS

```
3 Als Nutzer
4 moechte ich von der App benachrichtigt werden koennen
5 So dass ich Wasser regelmaessig trinken kann
6
7
8 Szenario: Zustand der getrunkenen Menge      #Eigene
   Interpretation
9 Gegeben sei, das Glas uebertraegt die Menge der getrunkenen
   Wasser an die App      #Eigene Interpretation
10 Wenn ich auf die App schaue      #Eigene Interpretation
11 Dann wird die getrunkene Menge auf der App angezeigt
12
13 Szenario: Erinnerung an das Trinken
14 Gegeben sei, die Menge ist auf der App angezeigt
15 Wenn ich Wasser nicht regelmaessig trinke
16 Dann werde ich benachrichtigt
17 Und ist die Benachrichtigung eine Erinnerung daran
   regelmaessige Wasser zu trinken
```

B.3 CVParse

1. Anmeldung

```
1 #Language: de
2 Feature: CVParse-1
3 Als Nutzer
4 moechte ich mich anmelden
5 so dass ich mein Lebenslauf kopieren kann
6
7 Szenarien: Registrierung
8 Gegeben sei, <Tools> ist funktionsfaehig
9 Wenn ich auf den "create" Link klicke
10 Dann kann ich meine <Daten> eingeben
11 Wenn ich auf den "create" Buttons klicke
12 Dann bekomme ich eine E-Mail mit Code fuer die Bestaetigung
13 Wenn ich auf den "Bestaetigen" Buttons klicke #Fehlende
   Angaben
14 Dann bestaetige ich meinen neuen Account      #Fehlende Angaben
15
16 Beispiele:
17 | Tools          | Daten          |
18 | Internetverbindung | E-Mail        |
19 | Geraet         | Passwort       |
20 |                | Repeat Passwort |
21
22 Szenario: Anmeldung
23 Gegeben sei, ich bin registriert
24 Wenn ich mein E-Mail und mein Passwort eingebe
25 Und ich auf dem "Login" Button klicke
26 Dann bin ich angemeldet
27 Und kann ich die Startseite sehen und bearbeiten #Fehlende
   Angaben
```

```
28
29 Szenario: Profil bearbeiten
30 Gegeben sei, ich bin angemeldet
31 Wenn ich auf der Logo von CVParser-1 klicke
32 Dann kann ich mein CVParser-1 Profil bearbeiten
```

2. Extraktion

```
1 #Language: de
2 Feature: CVParser-1
3 Als Nutzer
4 moechte ich mein Lebenslauf Mithilfe von LinkedIn umschreiben
5 so dass ich mein LinkedIn Profil nicht kopieren und an einer
   Firma schicken kann
6
7 Szenario: Lebenslauf generieren
8 Gegeben sei, ich bin angemeldet
9 Wenn ich mein LinkedIn Profil kopiere
10 Und ich es im Blank einfuege
11 Dann bekomme ich die Infos fuer mein LinkedIn Account
   eingefuegt
12 Wenn ich auf den "Extract" Button klicke
13 Dann wird mein Profil extrahiert
14 Und als Lebenslauf dargestellt #Fehlende Angaben
15 Und in der Datenbank der Firma
16
17 Szenario: Lebenslauf editieren
18 Gegeben sei, ich habe mein Lebenslauf extrahiert
19 Wenn ich mein Lebenslauf noch aendern moechte
20 Dann klicke ich auf den "Edit" Tabs
21 Und ich kann den Extrahierten Lebenslauf editieren
22 Wenn ich mit der Modifikation fertig bin
23 Dann klicke ich auf den "Confirm" Button
```

3. Abmeldung

```
1 #Language: de
2 Feature: CVParser-1
3 Als Nutzer
4 moechte ich mich abmelden
5 so dass jemand fremdes kein Zugriff zu meinem Account haben
   kann
6
7 Szenario: Tutorial
8 Gegeben sei, ich moechte mich einarbeiten
9 Wenn ich auf den "About" Tabs klicke
10 Dann kann ich ein paar Tutorial sehen.
11
12 Szenario: Abmeldung
13 Gegeben sei, ich bin angemeldet
14 Wenn ich auf dem "Logout" Button klicke
15 Dann bin ich abgemeldet
```

B.4 IntelligentCoach

1. Hochladen

```

1 #Language: de
2 Feature: Hochladen
3 Als Studierende
4 moechte ich meine Dateien hochladen
5 So dass ich sie gut bearbeiten kann
6
7 Szenarien: Registrierung #Fehlende Angaben
8 Gegeben sei, <Tools> ist funktionsfaehig #Fehlende Angaben
9 Wenn ich auf den "create" Link klicke #Fehlende Angaben
10 Dann kann ich meine <Daten> eingeben #Fehlende Angaben
11 Wenn ich auf den "create" Buttons klicke #Fehlende Angaben
12 Dann bekomme ich eine E-Mail mit Code fuer die Bestaetigung
   #Fehlende Angaben
13 Wenn ich auf den "Bestaetigen" Buttons klicke #
   Fehlende Angaben
14 Dann bestaetige ich meinen neuen Account #Fehlende Angaben
15
16 Beispiele: #Fehlende Angaben
17 | Tools | Daten |
18 | Internetverbindung | E-Mail |
19 | Geraet | Passwort |
20 | | Repeat Passwort |
21
22 Szenarien: Anmeldung in IntelligentCoach2
23 Gegeben sei, <Tools> ist funktionsfaehig
24 Wenn ich mich anzumelden versuche
25 Und ich meine WebSSo Daten eintrage
26 Dann bin ich angemeldet
27 Und habe Zugriff auf die Module
28
29 Beispiele:
30 | Tools |
31 | Internetverbindung |
32 | Geraet |
33
34 Szenarien: <Datei> hochladen.
35 Gegeben sei, ich bin angemeldet
36 Und ich habe Zugriff auf die Module <Datei>
37 Wenn ich auf dem Modul <Datei> klicke
38 Dann kann ich das Menue des Moduls ansehen
39 Wenn ich auf dem Modul "Dateien durchsuchen" klicke
40 Dann suche ich die <Datei>, die ich hochladen moechte
41 Und lade diese hoch.
42
43 Beispiele:
44 | Datei |
45 | Folien |
46 | Vortraege |

```

2. Folien bewerten

```

1 #Language: de
2 Feature: Folien bewerten
3 Als Studierende
4 moechte ich meine Folien bearbeiten
5 So dass ich meinen Vortrag gut halten kann
6
7 Szenario: Folien bewerten
8 Gegeben sei, die Folien sind hochgeladen
9   Und ich bin auf dem Menue des Moduls "Folien"
10 Wenn ich auf die Auswahl "Bewerten" klicke
11 Dann wird Jede Folie bewertet
12   Und der komplette Foliensatz am Ende mit einer gesamten
13   Einschaetzung und Verbesserungsvorschlaege versehen
14 Wenn ich auf "Bewertung beenden" klicke
15 Dann wird Meine gesamte Liste der Foliensaetze inklusive der
16   letzten Bewertung aktualisiert

```

3. Vorträge bewerten

```

1 #Language: de
2 Feature: Vortraege bewerten
3 Als Studierende
4 moechte ich meinen Vortrag bearbeiten
5 So dass ich gut vortragen kann
6
7 Szenarien: Vortraege bewerten
8 Gegeben sei, die Folien sind hochgeladen
9   Und ich bin auf dem Menue des Moduls "Folien"
10 Wenn ich auf die Auswahl "Link" klicke
11 Dann kann ich einen Link zum Vortrag erstellen und kopieren
12 Wenn ich den Link in einer Webseite einfuege
13 Dann kann ich mir den Vortrag anschauen
14   Und ein Bewertungsbogen mit einen 5-Stern Skala und einen
15   Freitext Feld ausfuellen
16 Wenn die Bewertung abgeschlossen ist
17 Dann kann ich der Bewertung <machen>
18
19 Beispiele:
20   | machen      |
21   | Ausblenden |

```

4. Tutorial

```

1 #Language: de
2 Feature: Tutorial
3 Als Studierende
4 moechte ich mein Tutorial lesen oder anschauen
5 So dass ich gut vortragen kann

```

```

6
7 Szenario: Tutorial lernen          #Fehlende Angabe
8 Gegeben sei, ich bin angemeldet    #Fehlende Angabe
9   Und ich habe Zugriff auf die Module #Fehlende Angabe
10 Wenn ich auf dem Modul "Tutorial" klicke #Fehlende Angabe
11 Dann kann ich durch Tutorial lernen #Fehlende Angabe

```

B.5 MobileMeetingMaster

1. Gesprächsaufnahme

```

1 #Language: de
2 Feature: Gespraechaufnahme
3 Als Mitarbeiter
4 moechte ich waehrend eines Meetings aufgenommen werden
5 So dass ich das Meeting auch draussen machen kann
6
7 Szenarien: Account erstellen      #Fehlende Angaben
8 Gegeben sei, <Tools> ist funktionsfaehig #Fehlende Angaben
9 Wenn ich auf den "create account" Link klicke #Fehlende
  Angaben
10 Dann kann ich meine <Daten> eingeben #Fehlende Angaben
11 Wenn ich auf den "create" Buttons klicke #Fehlende Angaben
12 Dann bekomme ich eine E-Mail mit Code fuer die Bestaetigung
  #Fehlende Angaben
13 Wenn ich auf den "Bestaetigen" Buttons klicke #Fehlende
  Angaben
14 Dann bestaetige ich meinen neuen Account #Fehlende Angaben
15
16 Beispiele: #Fehlende Angaben
17 | Tools | Daten |
18 | Internetverbindung | E-Mail |
19 | Smartphone | Name |
20 | | Passwort |
21 | | Repeat Passwort |
22
23 Szenarien: Anmeldung in MobileMeetingMaster #Fehlende Angaben
24 Gegeben sei, ich habe einen Account erstellt #Fehlende
  Angaben
25 Wenn ich mich anzumelden versuche #Fehlende Angaben
26   Und ich meine <Anmeldedaten> eintrage #Fehlende Angaben
27 Dann bin ich angemeldet #Fehlende Angaben
28
29 Beispiele: #Fehlende Angaben
30 | Anmeldedaten |
31 | Name |
32 | Passwort |
33
34 Szenario: Meeting starten          #Eigene Interpretation
35   Gegeben sei, ich bin angemeldet #Eigene Interpretation
36 Wenn ich druecke auf "Starten" #Eigene Interpretation
37 Dann kann das Meeting starten #Eigene Interpretation

```

```

37
38 Szenario: Automatische Aufnahme des Gespraechs #Eigene
    Interpretation
39 Gegeben sei, das Meeting laeuft #Eigene Interpretation
40 Wenn ich mit den Kollegen rede #Eigene Interpretation
41 Dann erkennt die App die Sprache #Eigene Interpretation
42 Und nimmt das Gespraech automatisch auf #Eigene
    Interpretation

```

2. Transkription des Gesprächs

```

1 #Language: de
2 Feature: Transkription des Gespraechs
3 Als Mitarbeiter
4 moechte ich mein Gespraech, waehrend eines Meeting
    transkribieren koennen
5 So dass ich das transkribierte Meeting weiterleiten kann
6
7 Szenario: Transkription des Gespraeches
8 Gegeben sei, das Gespraech ist aufgenommen
9 Wenn die App das aufgenommene Gespraech eingegeben bekommt
10 Dann transkribiert die App das Gespraech
11
12 Szenario: Zustand des Gespraechs
13 Gegeben sei, das Gespraech ist in der App transkribiert
14 Wenn ich schaue in der App rein
15 Dann kann ich sehen, was meine Kollegen und ich dann
    besprochen haben
16
17 Szenario: Automatische Gliederung des Gespraechs #Eigene
    Interpretation
18 Gegeben sei, ich kann sehen, was meine Kollegen und ich dann
    besprochen haben #Eigene Interpretation
19 Wenn ich das Gespraech zu gliedern versuche #Eigene
    Interpretation
20 Dann laesst sich das Gespraech automatisch gliedern #Eigene
    Interpretation

```

3. Weitergabe des Meetings

```

1 #Language: de
2 Feature: Weitergabe des Meetings
3 Als Mitarbeiter
4 moechte ich das transkribierte Meeting an der Chefin schicken
5 So dass sie das auch lesen kann
6
7 Szenario: Anpassung des Meetings #Eigene Interpretation
8 Gegeben sei, das Meeting ist transkribiert #Eigene
    Interpretation
9 Wenn ich mit dem transkribierten Meeting nicht einverstanden
    bin #Eigene Interpretation

```

86 ANHANG B. FEATURE-DATEIEN MIT SZENARIEN PRO VISIONSVIDEOS

```
10 Dann kann ich das Meeting anpassen      #Eigene Interpretation
11
12 Szenario: Ergebnisse weiterschicken
13 Gegeben sei, das Gespraech ist transkribiert und angepasst
14 Und der Empfaenger hat ein Konto erstellt #Eigene
    Interpretation
15 Wenn ich der Name der Empfaenger auswaehle #Eigene
    Interpretation
16 Dann schicke ich die Transkriptionen weiter
```

Anhang C

Erhobenen Anforderungen

Die unklare bzw. vagen Anforderungen sind mit ein Stern * markiert.

IntelligentCoach

- Feature 1: Hochladen
 1. Der Nutzer muss sich registrieren können. *
 2. Der Nutzer muss sich anmelden können
 3. Der Nutzer muss die Folien hochladen können
 4. Der Nutzer muss die Vorträge hochladen können
- Feature 2: Folien Bewerten
 5. Der Nutzer muss die Folien bewerten können
- Feature 3: Vorträge bewerten
 6. Der Nutzer muss die Vorträge bewerten können
- Feature 4: Tutorium
 7. Der Nutzer muss durch das Tutorium etwas lernen können. *

iKühlschrank

- Feature 1: Synchronisation
 1. Der Nutzer muss einen Account erstellen. *
 2. Der Nutzer muss sich in der App und im iKühlschrank anmelden können.
 3. Das Smartphone muss mit dem iKühlschrank synchronisiert werden

- Feature 2: Inventur
 4. Der Nutzer muss Kamera und Sensor aktivieren können.*
 5. Das System muss die Lebensmittel automatisch erfassen.
- Feature 3: Einkaufslisten
 6. Das System muss automatisch Einkaufslisten erstellen können.
 7. Das System sollte die Einkaufslisten aktualisieren können. *
- Feature 4: Ablaufdatum
 8. Der Nutzer muss die Haltbarkeit bestimmen können.
 9. Der Nutzer muss die Dauer bis zum Ablauf ermitteln können.
 10. Das System muss den Nutzern beim Ablauf eines Produkts benachrichtigen können.
- Feature 5: Rezeptideen
 11. Der Nutzer sollte passende Rezepte auf der Plattform von iKoch finden können.
 12. Der Nutzer muss ein Rezept auswählen können.
 13. Der Nutzer muss fehlende Zutaten ergänzen können.

CVParser

- Feature 1: Synchronisation
 1. Der Nutzer muss sich registrieren können.
 2. Der Nutzer muss sich anmelden können.
 3. Der Nutzer kann sein Profil bearbeiten.
- Feature 2: Extraktion
 4. Das System muss einen Lebenslauf generieren können.
 5. Der Nutzer muss seinen Lebenslauf bearbeiten können.
- Feature 3: Abmeldung
 6. Der Nutzer kann sich das Tutorial ansehen.
 7. Der Nutzer muss sich abmelden können.

Watertracer

- Feature 1: Verbindung
 1. Ein Account muss vom Nutzer erstellt werden können.*
 2. Der Nutzer muss sich im Watertracer anmelden können.*
 3. Das Tracer Glas muss automatisch mit dem System verbunden werden.
- Feature 2: Übertragung
 4. Die Menge des getrunkenen Wassers muss vom System übertragen werden können.
- Feature 3: Erinnerung
 5. Der Zustand des getrunkenen Wassers sollte vom Nutzer einsehbar sein.*
 6. Der Nutzer sollte regelmäßig vom System an das Trinken von Wasser erinnert werden.

MobileMeetingMaster

- Feature 1: Gesprächsaufnahme
 1. Der Nutzer muss einen Account erstellen können. *
 2. Der Nutzer muss sich im MobileMeetingMaster anmelden können. *
 3. Der Nutzer kann das Meeting starten. *
 4. Das System kann das Gespräch automatisch aufnehmen. *
- Feature 2: Transkription des Gesprächs
 5. Das System muss das Gespräch transkribieren können.
 6. Das System muss den Fortschritt der Transkription anzeigen können. *
 7. Das System muss das Gespräch automatisch strukturieren können. *
- Feature 3: Weitergabe des Meetings
 8. Der Nutzer sollte das transkribierte Meeting bearbeiten können. *
 9. Der Nutzer sollte die Ergebnisse weitergeben können.

Anhang D

Begleitdokumente der Studie

D.1 Information zur Studie Visionsvideos

Vielen Dank, dass Sie an dieser Studie teilnehmen. Wir erheben Daten von mehreren Teilnehmenden.

Die in der Studie gesammelten Daten helfen uns, Anforderungen aus Visionsvideos mit und ohne Gherkin Texte zu interpretieren. Die Daten werden zu Analysezielen in der Regel gemittelt und der Trend aller Daten ist für unsere Arbeit von Interesse.

Ablauf und Dauer

Im Rahmen der Studie werden Ihnen 2 kurze Visionsvideos gezeigt. Anschließend werden wir Sie um Ihr Feedback zu den Videoabschnitten bitten. Anschließend werden Sie gebeten einen knappen Fragebogen, inklusive demographischen Fragen, auszufüllen. Die Studie wird insgesamt ca. **30 Minuten** dauern. Sie haben das Recht, die Teilnahme an dieser Studie jederzeit und ohne Angabe von Gründen abubrechen.

Erhobene Daten

Während der Studie werden mehrere Daten in Form von Antworten auf einem Fragebogen erhoben. Teile davon sind Demographische Daten, welche nicht einer Zuordnung, sondern einer Einordnung der Allgemeingültigkeit der Ergebnisse dienen.

Datenschutz und Datenspeicherung

Die von Ihnen zur Verfügung gestellten Daten werden ausschließlich anonym und ohne Rückschlüsse auf einzelne Personen ausgewertet. Vor der Verarbeitung der Daten erfolgt eine umfangreiche Anonymisierung. Die Daten werden in Rahmen einer Masterarbeit in anonymisierte Form verfasst.

D.2 Einverständniserklärung

Bitte lesen Sie sich nun sorgfältig die Einverständniserklärung und die dort beschriebenen Hinweise durch.

Ich habe die Richtlinien dieser Studie gelesen und verstanden. Ich nehme freiwillig und ohne Vergütung an dieser Studie teil. Ich habe das Recht, die Teilnahme jederzeit und ohne Angabe von Gründen abzubrechen. Diese Einwilligung ist freiwillig. Ich kann sie ohne Angabe von Gründen verweigern, ohne dass ich deswegen Nachteile zu befürchten hätte.

Datenerfassung, -speicherung und -verwendung

Ich wurde darüber informiert, dass während der Studie die Angaben in dem Fragebogen erfasst, elektronisch für einen begrenzten Zeitraum gespeichert und zur Auswertung der Studie herangezogen werden. Die erfassten Daten werden alleinig für die Masterarbeit genutzt, ausschließlich anonymisiert ausgewertet und nur für uns zugänglich sein. Die Daten werden in einer Masterarbeit in anonymisierter Form verfasst. Sie haben gemäß Datenschutz gegenüber dem Informationsträger das Recht auf Auskunft, Berichtigung sowie Löschung Ihrer personenbezogenen Daten. Sie können diese Einwilligungserklärung jederzeit schriftlich widerrufen. Nach erfolgtem Widerruf werden Ihre personenbezogenen Daten gelöscht und ab diesem Zeitpunkt für keine weiteren Masterarbeiten mehr verwendet.

Unterschrift

Ich habe die Richtlinien der Studie gelesen und verstanden. Ich bin mit den aufgeführten Punkten der Einverständniserklärung einverstanden. Ich nehme freiwillig und ohne Vergütung an dieser Studie teil, indem ich meine E-Mail-Adresse somit eingebe.

Vorname, Nachname:.....

Ort, Datum und Unterschrift:.....

D.3 Probandeninformation

TN	Geschlecht	Studiengang	Fachbereich	Semester	Hochschule
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					

Tabelle D.1: Probandeninformation

D.4 Aufgabenstellung

Aufgabenstellung:

Ausgangssituation:

Sie haben die Rolle eines Kunde in einem Softwareprojekt. Sie sehen sich ein Visionsvideo an, welche für das Projekt erstellt wurde.

Aufgabenstellung:

Gruppe 1:

Sehen Sie sich das Visionsvideo an und füllen Sie einen allgemeinen Fragebogen bezüglich des Visionsvideos aus.

Gruppe 2:

Sehen Sie sich das Visionsvideo an und lesen Sie die dazugehörigen Gherkin Texte.

Füllen Sie einen allgemeinen Fragebogen bezüglich des Visionsvideos aus.

Hinweise für die Bearbeitung der Aufgabe:

Sie werden insgesamt zwei Visionsvideos ansehen. Jedes Visionsvideos wird zweimal betrachten (einmal durch und einmal segmentiert).

Gruppe 1:

Sie sehen das komplette Video zum ersten Mal und haben 5 Minuten Zeit, um die Fragen zu beantworten. Danach wird das Video vom Moderator segmentiert. Dann sehen Sie das Video ein zweites Mal, diesmal bereits segmentiert, und haben 10 Minuten Zeit, um Fragen zu beantworten.

Gruppe 2:

Sie sehen das komplette Video ein erstes Mal und haben 5 Minuten Zeit, um die Gherkin-Texte zu lesen. Danach haben Sie weitere 5 Minuten Zeit, um ein erstes Mal alle Fragen zu beantworten. Danach wird das Video vom Moderator segmentiert. Sie sehen das Video ein zweites Mal und dieses Mal bereits segmentiert. Nach jedem Segment haben Sie 30 Sekunden Zeit, um das Szenario zu lesen, das dem Segment entspricht. Am Ende der Wiedergabe des Videos haben Sie weitere 10 Minuten Zeit, um Fragen zu beantworten.

Bei Fragen oder Unklarheiten bitte ich Sie vor Beginn des Experiments zu fragen, damit vor der eigentlichen Durchführung alle Unklarheiten beseitigt sind.

Abbildung D.1: Aufgabenstellung

D.5 Fragebogen

Fragen	Anwortmöglichkeiten
1. Wie schwer können Sie aus das gerade angesehene Video Anforderungen erkennen?	Note zwischen 1 und 5. (5 bedeutet sehr schwer und 1 nicht schwer)
2. Listen Sie diese Anforderungen auf, die Sie erkannt haben	vollständige Aktive Sätze

Tabelle D.2: Fragen zu Gruppe 1

Fragen	Anwortmöglichkeiten
1-Sie haben den Test zur Gherkin Schulung bestanden. Im Vergleich zu dieser Schulung, wie gut haben Sie die in Gherkin textuellen beschriebenen Szenarien in dieser Studie verstanden?	Note zwischen 1 und 5. (5 bedeutet sehr schwer und 1 nicht schwer)
2. Wie schwer können Sie aus dem gerade angesehenen Video, Anforderungen erkennen?	Note zwischen 1 und 5. (5 bedeutet sehr schwer und 1 nicht schwer)
3. Listen Sie diese Anforderungen auf, die Sie erkannt haben	vollständige Aktive Sätze
4.a. Einige Schritte in Gherkin wurden nicht aus den Visionsvideos interpretiert, sondern vom Anforderungsingenieur selbst hinzugefügt. Diese Schritte sind mit #FA markiert und beschreiben die möglichen fehlenden Angaben. Sind Sie mit diesen hinzugefügten Schritten einverstanden	Ja oder Nein
4.b. Wenn Nein, Erläutern Sie an welche Stelle Sie die Schritte anders interpretiert hätten	Offene Frage
4.c. Wie würden Sie dann die Schritte anders interpretieren?	Offene Frage
5.a. Haben Sie andere Anforderung als gegebene Gherkin-Texte?	Ja oder Nein
5.b. Wenn ja, listen Sie diese auf	vollständige Aktive Sätze

Tabelle D.3: Fragen zu Gruppe 2

D.6 Gherkin-Schulung

24.01.23, 04:31

Gherkin-Schulung

Gherkin-Schulung

Für die Durchführung dieser Studie ist es notwendig, dass Sie die Grundlagen von Gherkin kennen. Diese Grundlagen werden Ihnen im Folgenden erklärt.

Gherkin ist eine Beschreibungssprache, die von Softwareentwicklern genutzt wird, um zu beschreiben, wie sich eine Software in bestimmten Situationen verhält. Dazu bietet Gherkin eine Reihe von Schlüsselbegriffen, die diese Beschreibung strukturieren. Für jeden Aspekt des Softwareverhaltens wird ein sogenanntes Szenario formuliert. Die Menge aller Szenarios beschreibt am Ende das vollständige Verhalten der Software. Im Folgenden sehen Sie ein Beispiel für ein Szenario, an dem wir die Grundlagen von Gherkin erklären. Die Software, auf die sich dieses Szenario bezieht, ist eine Webseite zum Verkauf von Kinotickets.

Szenarien: Erfolgreicher Kauf eines Kinotickets

Gegeben sei, ich befinde mich auf der Warenkorb-Seite

Und ich habe ein Kinoticket im Wert von 15€ in meinem Warenkorb

Und ich habe 20€ Guthaben

Wenn ich "Warenkorb bezahlen" auswähle

Dann sehe ich die Meldung "Vielen Dank für Ihren Einkauf. Viel Spaß mit dem Film."

Und mein neues Guthaben beträgt 5€

Und ich erhalte eine E-Mail, in der das Ticket als PDF-Datei angehängt ist

Abgesehen von dem Titel, der durch das Schlüsselwort Szenario markiert wird, besteht ein Szenario aus drei Teilen: Voraussetzung, Aktion und Ergebnis. Diese drei Teile werden durch die Gherkin-Schlüsselworte Gegeben sei, Wenn und Dann markiert.

- Die Voraussetzung (Gegeben sei,) beschreibt den Zustand, in dem sich die Software zu Beginn des Szenarios befindet.
- Das Schlüsselwort (Wenn) beschreibt eine Aktion oder ein Ereignis. Dabei kann es sich um eine Person handeln, die mit dem System interagiert, oder um ein Ereignis, das von einem anderen System ausgelöst wird.
- Das Ergebnis (Dann) beschreibt, wie das Programm auf die Aktion reagiert.

Das Schlüsselwort Und kann verwendet werden, um das vorherige Schlüsselwort zu wiederholen.

Es ist möglich, das gleiche Szenario mehrmals mit verschiedenen Wertkombinationen auszuführen. Das Kopieren und Einfügen von Szenarien, um verschiedene Werte zu verwenden, wird schnell langweilig und repetitiv.

Szenario: 5 von 12 Gurken essen

Gegeben sei, es gibt 12 Gurken

Wenn ich 5 Gurken esse

Dann sollte ich 7 Gurken haben

Szenario: 5 von 20 Gurken essen

Gegeben sei, es gibt 20 Gurken

24.01.23, 04:31

Gherkin-Schulung

Wenn ich 5 Gurken esse
Dann sollte ich 15 Gurken haben

Wir können diese beiden ähnlichen Szenarien zu einem Szenario mit Beispielen zusammenfassen. Es ermöglicht uns, diese Szenarien durch die Anwendung einer Vorlage (<>) mit begrenzten Parametern genauer zu formulieren:

Szenarien: Essen
Gegeben sei, es gibt <start> Gurken
Wenn ich <Anzahl> Gurken esse
Dann sollte ich <Rest> Gurken haben

Beispiele:

start	Anzahl	Rest
12	5	7
20	5	15

Eine Szenariogliederung muss ein oder mehrere Beispiel(e) enthalten. Die Schritte werden als Vorlage interpretiert, die nie direkt ausgeführt wird. Stattdessen wird die Szenariogliederung einmal für jede Zeile im darunter liegenden Abschnitt Beispiele ausgeführt (die erste Kopfzeile nicht mitgezählt).

Die Schritte können <>-begrenzte Parameter verwenden, die auf Kopfzeilen in der Beispieltabelle verweisen. Gherkin ersetzt diese Parameter durch Werte aus der Tabelle, bevor es versucht, den Schritt mit einer Schrittdefinition abzugleichen.

* Erforderlich

Um zu bestätigen, dass Sie den Text aufmerksam gelesen haben, beantworten Sie bitte einige Fragen. Die Studie kann erst fortgeführt werden, wenn Sie die Fragen korrekt beantwortet haben.

1. Was ist das Ziel von Gherkin? *

4 Punkte

Wählen Sie alle zutreffenden Antworten aus.

- Gherkin entscheidet, ob eine Software funktioniert oder nicht
- Gherkin ist eine Programmiersprache für die Entwicklung einer Software
- Gherkin beschreibt die Art und Weise, wie sich eine Software in bestimmten Situationen verhält
- Keine Antwort ist richtig

24.01.23, 04:31

Gherkin-Schulung

2. Welche der folgenden Schlüsselworte sind (gemäß obigem Text) in einem Szenario immer enthalten? * 4 Punkte

Wählen Sie alle zutreffenden Antworten aus.

- Dann
- Gegeben sei
- Und
- Wenn

3. Welche der folgenden Aussagen ist/sind korrekt? * 4 Punkte

Wählen Sie alle zutreffenden Antworten aus.

- Ein Szenario beschreibt, wie sich eine Software in einer bestimmten Situation verhält.
- Eine Voraussetzung wird durch das Schlüsselwort Wenn eingeleitet.
- Für eine Software werden in der Regel mehrere Szenarios erstellt.
- Nach dem Schlüsselwort Dann wird die Reaktion des Programms beschrieben.

4. Wofür kann das Schlüsselwort Und verwendet werden? * 4 Punkte

Wählen Sie alle zutreffenden Antworten aus.

- um das vorherige Schlüsselwort zu wiederholen.
- um das vorherige Schlüsselwort zu ergänzen.
- um das vorherige Schlüsselwort zu widersprechen.
- um das nächste Schlüsselwort zu wiederholen.

5. Wie kann man gleiche Szenario mehrmals mit verschiedenen Wertkombinationen ausführen? * 4 Punkte

Wählen Sie alle zutreffenden Antworten aus.

- Man kann nicht
- Durch Szenarien mit Beispielen
- Ein Szenario löschen und das andere behalten
- Durch die Anwendung einer Vorlage mit <> begrenzten Parametern kann man ein gleiches Szenario mehrmals ausführen

Anhang E

Ergebnisse der Studie

E.1 Frage 2 für Gruppe ohne Gherkin-Texte

E.1.1 Visionsvideo IntelligentCoach

Segment	Anforderungen
Authentifizierung	R1: Als Nutzer möchte ich mich über WebSSO registrieren können.
	R2: Als Nutzer möchte ich mich registrieren können.
	R3: Als Nutzer möchte ich meine Daten erfassen und einen Bestätigungscode senden können, um einen Account zu erstellen.
	R4: Als Nutzer möchte ich Zugriff auf die Module haben können.
Folien bewerten	R5: Als Nutzer kann ich Folien hochladen und einen Überblick über meine Dateien haben.
	R6: Als Nutzer kann ich meine Dateien bewerten lassen und sie jederzeit aktualisieren.
	R7: Als Nutzer kann ich die Verbesserungsvorschläge einsehen.
Vorträge bewerten	R8: Als Nutzer möchte ich Vortragsvideos hochladen können.
	R9: Als Nutzer möchte ich einen zeitlich und in seiner Aufrufzahl begrenzten Link zu dem Video erstellen und teilen können.
	R10: Als Nutzer möchte ich einen Link zu dem Video erstellen und teilen können, der zeitlich und in seiner Aufrufzahl begrenzt ist.
	R11: Als Nutzer möchte ich die Dauer festlegen können.
	R12: Als Nutzer möchte ich einen Bewertungsbogen ausfüllen können.

Tabelle E.1: Aufgelistete Anforderungen von IntelligentCoach

E.1.2 Visionsvideo CVParser

Segment	Anforderungen
Anmeldung	R1: Als Nutzer möchte ich die E-Mail-Adresse durch das Öffnen eines per E-Mail gesendeten Links bestätigen können.
	R2: Als Nutzer möchte ich mit E-Mail einen Account erstellen können
	R3: Als Nutzer möchte ich mich mit dem Benutzernamen und dem Passwort einloggen können.
Extraktion	R4: Als Nutzer möchte ich mein LinkedIn-Profil in den Parser kopieren und einfügen können.
	R5: Als Nutzer möchte ich den Lebenslauf bearbeiten können.
	R6: Als Nutzer möchte ich mein LinkedIn-Profil in den Parser einfügen und die damit verbundenen Daten in der Datenbank der Firma speichern und verarbeiten können.
Abmeldung	R7: Als Nutzer möchte ich Tutorials unter dem Reiter Tutorials ansehen können.
	R8: Als Nutzer möchte ich über meinem Profil durch Anklicken vom Smile mein Passwort bearbeiten können.
	R9: Als Nutzer möchte ich mich über den Logout-Button abmelden können.

Tabelle E.2: Aufgelistete Anforderungen von CVParser

E.2 Frage 3 für Gruppe mit Gherkin-Texte

E.2.1 Visionsvideo CVParser

Segment	Anforderungen
Anmeldung	R1: Als Nutzer möchte ich die E-Mail-Adresse durch das Öffnen eines per E-Mail gesendeten Links bestätigen können.
	R2: Als Nutzer möchte ich mit meiner E-Mail einen Account erstellen können.
	R3: Als Nutzer möchte ich mich mit dem Benutzernamen und dem Passwort einloggen können.
	R4: Als Nutzer möchte ich die Webseite aufrufen können.
Extraktion	R5: Als Nutzer möchte ich mein LinkedIn-Profil in den Parser des Systems kopieren und einfügen können.
	R6: Als Nutzer möchte ich aus dem eingefügten Profil einen Lebenslauf generieren können.
	R7: Als Nutzer möchte ich den Lebenslauf bearbeiten können.
	R8: Als Nutzer möchte ich meinen Lebenslauf parsen können.
	R9: Als Nutzer möchte ich das Format des Lebenslauf nicht ändern können.
	R10: Das System sollte mein Profil in der Datenbank der Firma speichern und versenden.
Abmeldung	R11: Als Nutzer möchte ich Tutorials unter dem Reiter Tutorials ansehen können.
	R12: Als Nutzer möchte ich mich durch Tutorials einarbeiten können.
	R13: Als Nutzer möchte ich über mein Profil durch Anklicken vom Smile mein Passwort bearbeiten können.
	R14: Als Nutzer möchte ich mich über den Logout-Button abmelden können.

Tabelle E.3: Aufgelistete Anforderungen von CVParser

E.2.2 Visionsvideo IntelligentCoach

Segment	Anforderungen
Authentifizierung	R1: Als Nutzer möchte ich mich über WEBSSO einloggen können.
	R2: Als Nutzer möchte ich mich authentifizieren können.
	R3: Als Nutzer möchte ich mich registrieren können.
	R4: Als Nutzer möchte ich Daten erfassen und einen Bestätigungscode senden können, um ein Konto zu erstellen.
	R5: Als Nutzer möchte ich Zugriff auf die Module haben können.
	R6: Als Nutzer ist eine gute Internetverbindung notwendig.
Folien bewerten	R7: Als Nutzer möchte ich einen Foliensatz ansehen können.
	R8: Als Nutzer möchte ich einen Foliensatz hochladen können.
	R9: Als Nutzer möchte ich einen Foliensatz automatisch bewerten lassen können.
	R10: Als Nutzer möchte ich die Bewertung des Foliensatzes einsehen können.
	R11: Als Nutzer kann ich die Verbesserungsvorschläge sehen.
	R12: Das System sollte jede Folie Einzel bewerten können.
	R13: Das System sollte der gesamte Foliensatz durch Auswählen nach Beendigung der Bewertung aktualisieren.
Vorträge bewerten	R14: Als Nutzer möchte ich ein Video hochladen können.
	R15: Als Nutzer möchte ich einen Link zu dem Video erstellen und diesen teilen können. Dabei soll der Zugriff auf das Video zeitlich begrenzt und die Anzahl der Aufrufe begrenzt sein.
	R16: Als Nutzer möchte ich ein Video über einen Link aufrufen und bewerten können, indem ich eine 5-Sterne-Skala und Textfelder verwende.
	R17: Als Nutzer möchte ich die Möglichkeit haben, Vortragsvideos anzuschauen, nachdem ich das Video verlinkt habe.
	R18: Als Nutzer möchte ich die Anzahl der Klicks und die Dauer des Videos bestimmen können.
	R19: Als Nutzer kann ich eine Videobewertungsfunktion ein- und ausblenden.

Tabelle E.4: Aufgelistete Anforderungen von IntelligentCoach

Literaturverzeichnis

- [1] Iso/iec/ieee international standard - systems and software engineering – vocabulary. *ISO/IEC/IEEE 24765:2010(E)*, pages 1–418, 2010.
- [2] I. F. Alexander and R. Stevens. *Writing better requirements*. Pearson Education, 2002.
- [3] S. Ambler. *Agile modeling: effective practices for extreme programming and the unified process*. John Wiley & Sons, 2002.
- [4] H. Anas, M. Ilyas, Q. Tariq, and M. Hummayun. Requirements validation techniques: An empirical study. *International Journal of Computer Applications*, 148:5–10, 08 2016.
- [5] apriorit. Requirements elicitation: Reasons, stages, techniques, and challenges. <https://www.apriorit.com/white-papers/699-requirement-elicitation#q4>, December 2022. Accessed on: 31.12.2022.
- [6] K. Beck. *Test-driven development: by example*. Addison-Wesley Professional, 2003.
- [7] A. Bogner, B. Littig, and W. Menz. *Interviews mit Experten: eine praxisorientierte Einführung*. Springer-Verlag, 2014.
- [8] E. Börger, B. Hörger, D. Parnas, and D. Rombach. Requirements capture, documentation and validation (dagstuhl seminar 99241). Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.
- [9] M. Braun. Nicht-funktionale anforderungen. *Juristisches IT-Projektmanagement Lehrstuhl für Programmierung und Softwaretechnik Ludwig-Maximilians-Universität München*, 2016.
- [10] O. Brill, K. Schneider, and E. Knauss. Videos vs. use cases: Can videos capture more requirements under time pressure? In *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pages 30–44. Springer, 2010.
- [11] R. Cimperman. *Uat defined: A guide to practical user acceptance testing (digital short cut)*. Pearson Education, 2006.
- [12] A. Classen, P. Heymans, and P. Y. Schobbens. What’s in a feature: A requirements engineering perspective. pages 16–30, 03 2008.
- [13] M. Cohn. *User stories applied: For agile software development*. Addison-Wesley Professional, 2004.
- [14] P. Consulting. Das Validieren von Anforderungen. https://www.peterjohann-consulting.de/validieren-von-anforderungen/#4_haufige_fragen_und_antworten_zum_validieren_von_anforderungen. letzter Zugriff: September 2022.
- [15] P. Consulting. Reviews, prüfen von produktanforderungen. https://www.peterjohann-consulting.de/reviews/#2_die_drei_wesentlichen_review-techniken, 2005-2022. Accessed on: 27.12.2022.

- [16] O. Creighton, M. Ott, and B. Bruegge. Software cinema-video-based requirements engineering. In *14th IEEE International Requirements Engineering Conference (RE'06)*, pages 109–118, 2006.
- [17] Cucumber. Gherki reference. <https://cucumber.io/docs/gherkin/reference/>. Accessed on: 27.12.2022.
- [18] I. Dees, M. Wynne, and A. Hellesoy. *Cucumber Recipes: Automate Anything with BDD Tools and Techniques*. Pragmatic Bookshelf, 2013.
- [19] J. Dick, E. Hull, and K. Jackson. Requirements engineering. In *Requirements Engineering*. Springer, 2017.
- [20] C. Ebert. *Systematisches Requirements Engineering: Anforderungen ermitteln, dokumentieren, analysieren und verwalten*. dpunkt. verlag, 2019.
- [21] M. Eid, M. Gollwitzer, and M. Schmitt. *Statistik und forschungsmethoden*. 2017.
- [22] A. Endmann and D. Keßner. User journey mapping—a method in user experience design. *i-com*, 15(1):105–110, 2016.
- [23] S. A. Fricker, K. Schneider, F. Fotrousi, and C. Thuemmler. Workshop videos for requirements communication. *Requirements Engineering*, 21(4):521–552, 2016.
- [24] M. Glinz. Requirements engineering i kapitel 5, anforderungsspezifikation mit natürlicher sprache universität zürich institut für informatik. https://files.ifi.uzh.ch/rerg/amadeus/teaching/courses/re_I_hs09/Kapitel_05_Natuerl_Spr.pdf. letzter Zugriff: September 2022.
- [25] A. Gołąb. Banda. <https://issuu.com/chochlik/docs/banda>, 2012. Accessed on: 27.01.2023.
- [26] J. J. Gutiérrez, I. Ramos, M. Mejías, C. Arévalo, J. M. Sánchez-Begines, and D. Lizcano. Modelling gherkin scenarios using uml. 2017.
- [27] T. Hamilton. Gherkin language: Format, syntax & gherkin test in cucumber. <https://www.guru99.com/gherkin-test-cucumber.html>, 2022. Accessed on: 27.12.2022.
- [28] M. Härlin. Testing and gherkin in agile projects, 2016.
- [29] S. Hasan. Behavior driven development(bdd) with gherkin. https://testsigma.com/blog/behavior-driven-development-bdd-with-gherkin/#Benefits_of_using_BDD, 2022. Accessed on: 27.12.2022.
- [30] S. T. help. Top 10 most common requirements elicitation techniques. <https://www.softwaretestinghelp.com/requirements-elicitation-techniques/>, 2022. Accessed on: 27.12.2022.
- [31] P. Hruschka et al. Handbuch RE@Agile – Aus- und Weiterbildung zum IREB Certified Professional for Requirements Engineering Advanced Level RE@Agile . https://www.ireb.org/content/downloads/23-cpre-advanced-level-re-agile-handbook/handbook_cpre_al_re@agile_de_v2.0.pdf. letzter Zugriff: September 2022.
- [32] C. Hubbard. bdd vs tdd. <http://hubbardhive.com/wp-content/uploads/2018/a2ed945/bdd-vs-tdd-16ad35>, December 12, 2020. Accessed on: 27.12.2022.
- [33] IREB. International requirements engineering board. <https://www.ireb.org/de>. Accessed on: 27.01.2023.
- [34] Jaxenter. Req4Arc: Vom Umgang mit funktionalen Anforderungen. <https://entwickler.de/agile/req4arc-vom-umgang-mit-funktionalen-anforderungen>. letzter Zugriff: September 2022.
- [35] S. M. Karadimitriou, E. Marshall, and C. Knox. Mann-whitney u test. *Sheffield: Sheffield Hallam University*, 2018.

- [36] O. Karras. Software professionals' attitudes towards video as a medium in requirements engineering. In M. Kuhrmann, K. Schneider, D. Pfahl, S. Amasaki, M. Ciolkowski, R. Hebig, P. Tell, J. Klünder, and S. Küpper, editors, *Product-Focused Software Process Improvement*, pages 150–158, Cham, 2018. Springer International Publishing.
- [37] O. Karras, S. Kiesling, and K. Schneider. Supporting requirements elicitation by tool-supported video analysis. In *2016 IEEE 24th International Requirements Engineering Conference (RE)*, pages 146–155, 2016.
- [38] O. Karras and K. Schneider. An interdisciplinary guideline for the production of videos and vision videos by software professionals, 2020.
- [39] O. Karras, K. Schneider, and S. A. Fricker. Representing software project vision by means of video: A quality model for vision videos. *Journal of Systems and Software*, 162:110479, 2020.
- [40] O. Karras, K. Schneider, and S. A. Fricker. Representing software project vision by means of video: a quality model for vision videos. *journal of Systems and Software*, 162:110479, 2020.
- [41] O. Karras, C. Unger-Windeler, L. Glauer, and K. Schneider. Video as a by-product of digital prototyping: Capturing the dynamic aspect of interaction. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*, pages 118–124, 2017.
- [42] S. Karras, Klünder. Enrichment of requirements specifications with videos - enhancing the comprehensibility of textual requirements. In *Zenodo*, 2016.
- [43] T. Kasper. Änderungsmuster im requirements engineering identifikation und klassifikation. *Softwaretechnik-Trends*, 24(4), 2004.
- [44] S. Kiesling, O. Karras, and K. Schneider. Reqvida—requirements video analyzer. *GI Softwaretechnik-Trends*, 36(3), 2015.
- [45] C. Lewis, P. G. Polson, C. Wharton, and J. Rieman. Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. CHI '90, page 235–242, New York, NY, USA, 1990. Association for Computing Machinery.
- [46] P. Li. *Jira Software Essentials: Plan, track, and release great applications with Jira Software*. Packt Publishing Ltd, 2018.
- [47] M. Lutz. *Learning python: Powerful object-oriented programming*. O'Reilly Media, Inc., 2013.
- [48] L. A. Macaulay. *Requirements engineering*. Springer Science & Business Media, 2012.
- [49] T. W. MacFarland, J. M. Yates, T. W. MacFarland, and J. M. Yates. Mann–whitney u test. *Introduction to nonparametric statistics for the biological sciences using R*, pages 103–132, 2016.
- [50] N. Matloff. *The art of R programming: A tour of statistical software design*. No Starch Press, 2011.
- [51] P. E. McKnight and J. Najab. Mann-whitney u test. *The Corsini encyclopedia of psychology*, pages 1–1, 2010.
- [52] M. Mernik, J. Heering, and A. M. Sloane. When and how to develop domain-specific languages. *ACM Comput. Surv.*, 37(4):316–344, dec 2005.
- [53] L. Nagel, J. Shi, and M. Busch. Viewing vision videos online: Opportunities for distributed stakeholders. In *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*, pages 306–312, 2021.

- [54] D. Pandey, U. Suman, and A. K. Ramani. An effective requirement engineering process model for software development and requirements management. In *2010 International Conference on Advances in Recent Technologies in Communication and Computing*, pages 287–291. IEEE, 2010.
- [55] H. Partsch. *Requirements-Engineering systematisch: Modellbildung für softwaregestützte Systeme*. Springer-Verlag, 2010.
- [56] H. Peterjohann. Requirement engineering, 2017. Accessed on: 27.01.2023.
- [57] R. Pham, S. Meyer, I. Kitzmann, and K. Schneider. Interactive multimedia storyboard for facilitating stakeholder interaction: Supporting continuous improvement in it-ecosystems. In *2012 Eighth International Conference on the Quality of Information and Communications Technology*, pages 120–123, 2012.
- [58] K. Pohl and C. Rupp. *Basiswissen requirements engineering: Aus-und Weiterbildung nach IREB-Standard zum certified professional for requirements engineering foundation level*. dpunkt. verlag, 2021.
- [59] Raven51. Workshop. <https://raven51.de/wiki/workshop/>, 2022. Accessed on: 27.01.2023.
- [60] R.Heini. Funktionale, nicht funktionale anforderungen. http://www.anforderungsmanagement.ch/in_depth_vertiefung/funktionale_nicht_funktionale_anforderungen/index.html. Accessed on: 27.12.2022.
- [61] S. Robertson and J. Robertson. *Mastering the requirements process: Getting requirements right*. Addison-wesley, 2012.
- [62] C. Rupp. Requirements-engineering und-management; das handbuch für anforderungen in jeder situation. 7., aktualisierte und erweiterte auflage. *Hanser, München*, 2021.
- [63] C. Rupp, M. Simon, and F. Hocker. Requirements engineering und management. *HMD Praxis der Wirtschaftsinformatik*, 46(3):94–103, 2009.
- [64] K. Schneider and L. M. Bertolli. Video variants for crowdre: How to create linear videos, vision videos, and interactive videos. In *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, pages 186–192, 2019.
- [65] W. Schramm. Kapitel 3 – ANFORDERUNGSANALYSE UND SPEZIFIKATION Software Technik. <https://services.informatik.hs-mannheim.de/~schramm/see/files/Kapitel03.pdf>. letzter Zugriff: September 2022.
- [66] S. Sendall. Requirements elicitation with use cases. In *International Workshop on Scientific Engineering of Distributed Java Applications*, pages 203–203. Springer, 2002.
- [67] A. Shahid. What is gherkin language? steps to write accurate tests. <https://enou.co/blog/gherkin-language/>, july 2022. Accessed on: 27.12.2022.
- [68] Simplilearn. What is elicitation and top requirement elicitation techniques you should know in 2023. https://www.simplilearn.com/what-is-elicitation-article#requirements_elicitation_techniques, 2023. Accessed on: 27.12.2022.
- [69] V. Sinha, F. Doucet, C. Siska, R. Gupta, S. Liao, and A. Ghosh. Yaml: a tool for hardware design visualization and capture. In *Proceedings 13th International Symposium on System Synthesis*, pages 9–14. IEEE, 2000.
- [70] J. Smart. *BDD in Action: Behavior-driven development for the whole software lifecycle*. Simon and Schuster, 2014.
- [71] H. Tresp. Funktionale anforderungen spezifizieren. In *Agile objektorientierte Anforderungsanalyse*, pages 119–143. Springer, 2022.

- [72] UmfrageOnline. Professionelle online umfragen erstellen, leicht gemacht. <https://www.umfrageonline.com/>, 2007 - 2023. Accessed on: 27.12.2022.
- [73] R. Van Solingen, V. Basili, G. Caldiera, and H. D. Rombach. Goal question metric (gqm) approach. *Encyclopedia of software engineering*, 2002.
- [74] Visure. Requirements engineering: Schritt für schritt. <https://visuresolutions.com/de/Blog/Requirements-Engineering-Prozess/>, 2022. Accessed on: 27.12.2022.
- [75] Visure. What is a requirements validation: Definition, process & tools. <https://visuresolutions.com/blog/requirements-validation/>, 2022. Accessed on: 27.12.2022.
- [76] R. Wirdemann. *Scrum mit user stories*. Carl Hanser Verlag GmbH Co KG, 2022.
- [77] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012.
- [78] M. Wynne, A. Hellesoy, and S. Tooke. *The cucumber book: behaviour-driven development for testers and developers*. Pragmatic Bookshelf, 2017.

