

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

Automatisierung von Guidelines für Videoerstellung im Requirements Engineering

Automating Guidelines for Video
Production in Requirements Engineering

Bachelorarbeit

im Studiengang Informatik

von

Janko Happe

Prüfer: Prof. Dr. Kurt Schneider
Zweitprüfer: Prof. Dr. Joel Greenyer
Betreuer: M. Sc. Oliver Karras

Hannover, 28.09.2018

Zusammenfassung

Automatisierung von Guidelines für Videoerstellung im Requirements Engineering

Eine zentrale Schwierigkeit im Requirements Engineering besteht darin, ein gemeinsames Verständnis zwischen Entwickler und Kunde zu schaffen. Eine aus dieser Schwierigkeit resultierende Methode ist die Erstellung von anforderungsbezogenen Videos durch die Softwareentwickler, da es sich bei Videos um einen sehr reichhaltigen Kommunikationsweg handelt. Eine zentrale Anwendung von Videos ist die Visualisierung von Visionen oder zukünftigen Szenarien. Die Videos werden dem Kunden vorgespielt, um zu verifizieren, dass diese Vision mit der Idee des Kunden übereinstimmt beziehungsweise um Missverständnisse zu erkennen. Aus vorherigen Arbeiten gingen Vorteile von Videos gegenüber textuellen Use Cases hervor. Dennoch handelt es sich hierbei nicht um eine im Requirements Engineering etablierte Methode. Ein Grund dafür könnte die fehlende Erfahrung der Softwareentwickler in der Videoproduktion sein. Der Aufwand für die Produktion der Videos ist womöglich zu hoch oder die Qualität der Videos ist nicht zufriedenstellend.

In dieser Arbeit wird ein prototypischer Videorecorder entwickelt, der den Nutzer dabei unterstützen soll gute Videos zu erstellen. Um Anhaltspunkte dafür zu erhalten was "gute Videos" ausmacht, wird sich mit Guidelines zur Videoproduktion auseinandergesetzt. Anschließend werden Konzepte erstellt, um diese Guidelines durch den Videorecorder zu automatisieren.

In einer Evaluation wurde der Prototyp mit einer üblichen Handykamera verglichen. Dabei wurden von den Probanden eine statistisch signifikant höhere subjektive Videoqualität und ein geringerer geschätzter Produktionsaufwand bei Benutzung des Prototyps angegeben.

Abstract

Automating Guidelines for Video Production in Requirements Engineering

A key difficulty in requirements engineering is to create a shared understanding between developer and customer. One from this difficulty resulting method is the creation of requirement related videos by the software engineers, since video is a very rich mode of communication. A central application of videos is the visualization of visions or future scenarios. The videos are played to the customer to verify that this vision matches the customer's idea or to recognize misunderstandings. Previous works have shown advantages of video over textual use cases. Nevertheless, this is not a method established in requirements engineering. One reason for this could be the lack of experience of software developers in video production. The effort needed to produce the videos may be too high or the quality of the videos not satisfactory.

In this work a prototype of a video recorder is developed, which supports the user in creating good videos. To get an idea about what constitutes "good videos", guidelines for video production were analyzed. Subsequently concepts were created to automate these guidelines through the video recorder.

In an evaluation the prototype was compared with a standard mobile phone camera. The subjects indicated a statistically significantly higher subjective video quality and a lower estimated production effort when using the prototype.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Motivation	1
1.2. Zielsetzung	2
1.3. Struktur der Arbeit	4
2. Grundlagen	5
2.1. Requirements Engineering	5
2.1.1. Requirements Analysis	6
2.1.2. Requirements Management	7
2.2. Videos im Requirements Engineering	8
2.3. Verwandte Arbeiten	10
3. Anforderungsermittlung	12
3.1. Anwendungsfall	12
3.2. Guidelines zur Videoproduktion	13
3.3. Basisanforderungen	15
3.4. Leistungsanforderungen	18
3.4.1. Anforderungen aus betrachteten Anwendungen	18
3.4.2. Konzepte zur Automatisierung von Guidelines	19
3.5. Begeisterungsanforderungen	24
4. Prototypische Umsetzung	25
4.1. Technische Grundlagen	25
4.2. Eingrenzung der implementierten Anforderungen	25
4.3. Design- und Entwurfsentscheidungen	27
4.3.1. Szenario Struktur	27
4.3.2. Videoplayer und Export	28
4.3.3. Überprüfung der Umgebung durch Sensorik	29
4.3.4. Dauer und Pausieren der Aufnahme	32
4.4. Einschränkungen	33
5. Evaluation	34
5.1. Vorbereitung mit GQM	34
5.2. Planung	35
5.3. Durchführung	37
5.3.1. Population	37
5.3.2. Ablauf einer Sitzung	37

Inhaltsverzeichnis

5.4. Auswertung	38
5.4.1. Ergebnisse der Evaluation	38
5.4.2. Bewertung der Ergebnisse	40
5.4.3. Bedrohung der Validität	42
5.4.4. Gewonnene Erkenntnisse aus dem praktischen Einsatz	43
6. Fazit und Ausblick	44
6.1. Fazit	44
6.2. Ausblick	45
A. Anforderungsermittlung: Features aus Kameraanwendungen	46
B. Evaluation: Abstraction Sheet	47
C. Inhalte der CD	48

1. Einleitung

1.1. Motivation

Durch die technische Entwicklung der letzten Jahre, und besonders durch die Etablierung von Smartphones und Social Media, ist die Benutzung von Videos für Privatpersonen inzwischen deutlich einfacher und verbreiteter geworden. Videos sind ein guter Weg, um Informationen verständlich zu vermitteln [1]. Im Requirements Engineering geht es um die Erhebung von Anforderungen für ein zu entwickelndes System. Ein zentrales Ziel ist hier das Erreichen eines gemeinsamen Verständnisses zwischen Kunden und Entwicklern [2][3]. Eine Herausforderung ist dabei die so genannte *Symmetry of Ignorance* [4]: Die Entwickler kennen sich nicht mit der Domäne der Kunden aus, und andersherum kennen sich die Kunden nicht mit Softwareentwicklung und den bestehenden Möglichkeiten aus. Deswegen ist es besonders essentiell für die erfolgreiche Zusammenarbeit, zunächst ein gemeinsames Verständnis zu schaffen. Videos können hier ein sinnvolles Werkzeug sein, was bereits durch bestehende Arbeiten gezeigt wurde [5].

Videos werden zum Teil zur Darstellung von Szenarien im *Scenario-based Requirements Engineering* eingesetzt. Die Szenarien können in den Punkten *Vergegenwärtigung zur Exploration des Designs*, *Anforderungserhebung* und *Validierung* eine Rolle spielen [6]. Im Folgenden werden diese Szenarien in Videoform der Einfachheit halber auch als *Vision Videos* bezeichnet. Vision Videos können von Entwicklerteams erstellt werden, nachdem die Kunden ihre Anforderungen geäußert haben. Diese Methode dient dazu zu überprüfen, ob die Anforderungen richtig Verstanden wurden, Missverständnisse aufzulösen und möglicherweise auch weitere versteckte Anforderungen zu entdecken, an die die Kunden zunächst nicht gedacht haben. Brill et al. [5] haben bereits in einer Studie gezeigt, dass Vision Videos vergleichbar oder besser dazu geeignet sind, um Missverständnisse in den Anforderungen zu vermeiden, als die traditionellen Use Cases [5]. Szenarien in Videoform sind weniger abstrakt, als formale, in der Softwareindustrie übliche Notationen. Sie sind für Außenstehende leichter verständlich und greifbarer. Es können auf diese Weise Herausforderungen, wie missverstandene Terminologie oder unterschiedliche Vorstellungen der Realität, die bei Anforderungen in reiner Textform auftreten können, weitestgehend eliminiert werden [7].

Trotz ihrer Vorteile, im Bezug auf das Schaffen eines klaren, gemeinsamen Verständnisses, sind Videos bis heute keine etablierte Dokumentationsoption in den Best Practices des Requirements Engineerings [8][3]. Motivation dieser Arbeit ist die Annahme, dass durch einen erleichterten Prozess der Videoerstellung, Softwareent-

wickler eher dazu geneigt wären, Vision Videos in ihren Arbeitsablauf aufzunehmen. Neben dem höheren Aufwand ist es vermutlich eine Hürde, dass Softwareentwickler nicht zwangsläufig Erfahrung mit der Erstellung von Videos haben und somit der Prozess erschwert wird [9]. Ihnen fehlt die nötige Unterstützung und es würde sich wahrscheinlich nicht rentieren spezielles Personal zur Videoproduktion einzustellen. Vor allem würde dies dazu führen, dass eine weitere Partei die Anforderungen verstehen muss und somit wieder mehr Raum für Missverständnisse entsteht. Es ist also ein Ansatz zu bevorzugen, der die Softwareentwickler selber bei der Produktion von Videos unterstützt. Ein solcher Ansatz könnte ein Videorecorder-Tool sein, welches Guidelines zur Videoproduktion automatisiert. Ein solches Tool könnte ohne große Kosten oder Aufwand die Erstellung von Videos erleichtern und somit die Schwelle für Softwareentwickler senken, um diesem Medium eine größere Chance zu geben.

1.2. Zielsetzung

Ziel dieser Arbeit ist die Implementierung und Evaluation eines Prototyps einer Software zur Aufnahme von Videos, die Nutzer ohne größere Kenntnisse zur Videoproduktion dabei unterstützt, gute Videos zu erstellen.

Nach dem Kano-Modell (siehe Abbildung 1.1) spielen in der Produktentwicklung drei verschiedene Faktoren eine Rolle für die Kundenzufriedenheit: *Basisfaktoren*, *Leistungsfaktoren* und *Begeisterungsfaktoren* [10]. Auch ohne einen tatsächlichen Kunden, möchte ich mich an diesem Modell orientieren, um möglichst die umfassenden Anforderungen für ein gutes Produkt abzudecken.

Basisfaktoren sind selbstverständliche Anforderungen, die der Kunde nur wahrnimmt, wenn diese nicht erfüllt werden. Sie steigern also nicht die Kundenzufriedenheit, sondern senken sie nur bei Nichterfüllung. In diesem Fall sind die Basisanforderungen all jene Anforderungen, die jeder Videorecorder erfüllen sollte. Dazu sollen vor allem verschiedene vorhandene Apps verglichen und eine Schnittmenge von Features gebildet werden, die allgemein benötigt zu werden scheinen.

Leistungsfaktoren sind dem Kunden bewusste Anforderungen, die je nach Erfüllungsgrad sowohl für Zufriedenheit als auch für Unzufriedenheit sorgen können. Hier sind dies die Anforderungen zur Automatisierung und Unterstützung des Nutzers, die über einen üblichen Videorecorder hinausgehen. Dafür müssen zunächst allgemeine Guidelines zur Videoproduktion zusammengetragen werden. Zu diesen Guidelines sollen dann Konzepte zur Automatisierung durch die Software erstellt werden. Mit Automatisierung ist gemeint, dass die Anwendung dem Nutzer gewisse Vorgaben zur Aufnahme des Videos macht. Wenn zum Beispiel nach einer Guideline kein Zoom verwendet werden soll, dann könnte das bedeuten, dass Zoom in der Anwendung deaktiviert wird. Eine Alternative wäre, dass beim Zoomen eine Meldung erscheint, die den Nutzer darauf hinweist, dass er vielleicht besser keinen Zoom benutzen und dafür mehr Schnitte machen sollte. Die genauen Konzepte zu den Guidelines werden im Verlauf der Arbeit noch diskutiert.

1. Einleitung

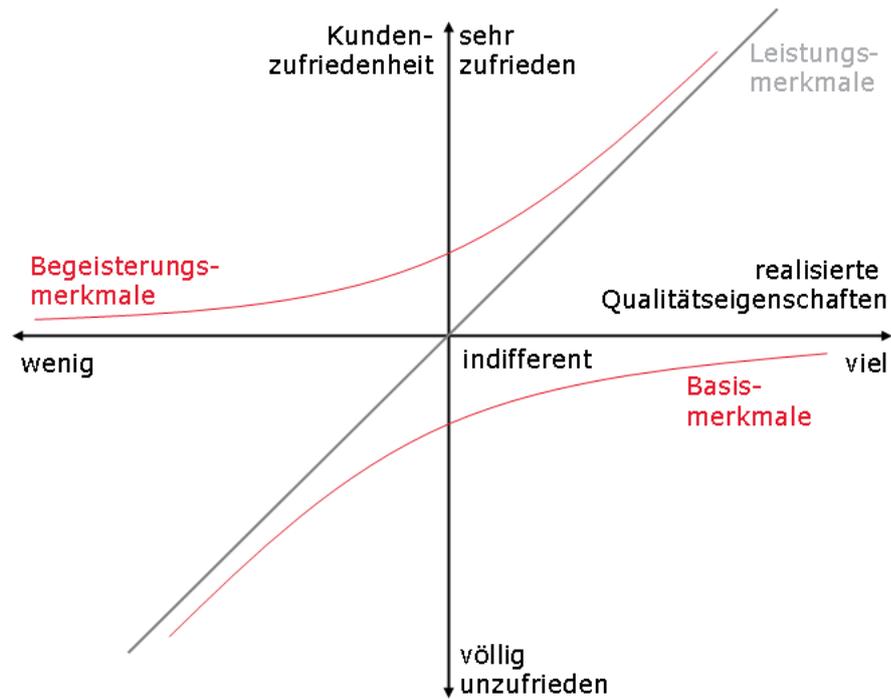


Abbildung 1.1.: Kano-Modell [10]

Begeisterungsfaktoren sind quasi das Gegenstück zu den Basisfaktoren. Der Kunde rechnet nicht mit ihnen oder weiß vielleicht gar nicht, dass diese Aspekte möglich sind, weshalb sie nur zu einer höheren Zufriedenheit führen können. Diese besonderen Anforderungen sind schwer zu planen, gerade wenn es keinen Kunden gibt. In diesem Fall werden Ideen zur Unterstützung des Nutzers welche über die Basis- und Leistungsfaktoren hinausgehen als solche bezeichnet.

Nachdem die Anforderungen festgelegt wurden, sollen ausgewählte Konzepte prototypisch implementiert und evaluiert werden. Die implementierte Software soll zur Evaluation von potentiellen Nutzern getestet werden. Dabei gilt es zu überprüfen, ob diese die Automatisierung als angenehm beziehungsweise hilfreich empfinden. Dies könnte der Fall sein, wenn mindestens eine der folgenden Hypothesen sich bewahrt:

1. Durch eine anwendungsseitige Unterstützung kann der Produktionsaufwand von Vision Videos gesenkt werden.
2. Durch eine anwendungsseitige Unterstützung bei der Produktion kann die Qualität von Vision Videos gesteigert werden.

1.3. Struktur der Arbeit

In diesem Abschnitt soll ein kurzer Überblick über den Aufbau der Arbeit gegeben werden. Diese Arbeit gliedert sich in sechs Kapitel.

In Kapitel 1 soll die Motivation der Arbeit deutlich gemacht werden und dem Leser eine ungefähre Vorstellung des Inhaltes der restlichen Arbeit vermittelt werden.

In Kapitel 2 werden Grundlagen erklärt, welche für das Verständnis der restlichen Arbeit notwendig sind. Dazu werden zunächst die einzelnen Bestandteile des Requirements Engineerings erläutert. Anschließend wird speziell auf Videos im Requirements Engineering und auf verwandte Arbeiten mit Bezug dazu eingegangen.

In Kapitel 3 werden Guidelines zur Videoproduktion zusammengetragen. Auf dessen Basis, und zusätzlich aus häufigen Features von vorhandenen Kameraanwendungen, werden dann Anforderungen für einen Videorecorder erhoben, welcher den Nutzer unterstützt die Guidelines einzuhalten.

In Kapitel 4 wird die prototypische Implementation eines solchen Videorecorders gezeigt.

In Kapitel 5 wird der zuvor erstellte Prototyp evaluiert und mit einer üblichen Handykamera verglichen, um den Nutzen in Bezug auf die Zielsetzung der Arbeit zu untersuchen.

In Kapitel 6 werden schließlich die Ergebnisse der vorherigen Kapitel zusammengefasst und ein Ausblick für weiterführende Arbeiten geschaffen.

2. Grundlagen

In diesem Kapitel sollen einige Grundlagen dargelegt werden, die für das weitere Verständnis der Arbeit von Nöten sind.

2.1. Requirements Engineering

Wie im vorherigen Kapitel beschrieben, ist das Thema dieser Arbeit Teil des Requirements Engineerings. Requirements Engineering ist laut IREB (International Requirements Engineering Board) folgendermaßen definiert [11]:

Requirements Engineering ist ein systematischer und disziplinierter Ansatz zur Spezifikation und zum Management von Anforderungen mit den folgenden Zielen:

1. Die relevanten Anforderungen zu kennen, Konsens unter den Stakeholdern über die Anforderungen herzustellen, die Anforderungen konform zu vorgegebenen Standards zu dokumentieren und die Anforderungen systematisch zu managen.
2. Die Wünsche und Bedürfnisse der Stakeholder zu verstehen und zu dokumentieren.
3. Die Anforderungen zu spezifizieren und zu managen, um das Risiko zu minimieren, dass das System nicht den Wünschen und Bedürfnissen der Stakeholder entspricht.

Die Ziele aus der Definition können in zwei Gebiete unterteilt werden: Requirements Analysis und Requirements Management. Bei der Analyse handelt es sich vereinfacht gesagt um die Erhebung und Interpretation der Anforderungen der Kunden und beim Management um den Umgang mit Änderungen und Nachverfolgbarkeit. Genauer können nach Börger et al. [12] diese beiden Gebiete in sieben weitere Komponenten, wie in Abbildung 2.1 zu sehen ist, unterteilt werden [12]. Im Folgenden werden die einzelnen Komponenten, basierend auf Rupp [13] und Pohl [14], genauer erläutert.

2. Grundlagen

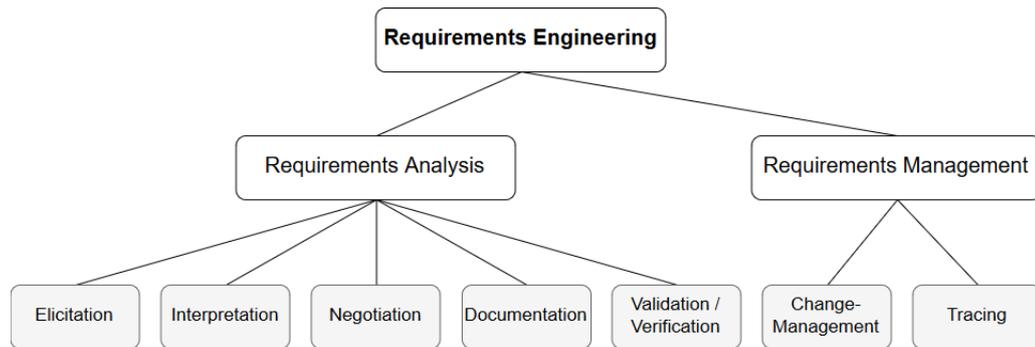


Abbildung 2.1.: Referenzmodell des Requirements Engineerings [12]

2.1.1. Requirements Analysis

Elicitation

Die üblicherweise erste Phase der Anforderungsanalyse wird als *Elicitation* oder *Anforderungserhebung* bezeichnet. In dieser Phase geht es zunächst um das Sammeln von Informationen. Der erste Schritt ist das Ermitteln aller sogenannten *Stakeholder*, den verschiedenen Interessengruppen, mit unterschiedlichem Bezug zum Projekt. Dazu gehören zum Beispiel die Kunden, aber auch ein späterer Nutzer des Systems ist ein Stakeholder. Danach muss versucht werden, die genauen Interessen dieser Stakeholder festzustellen. Dies kann über verschiedene Wege geschehen, wie zum Beispiel Interviews, Workshops oder Fragebögen. Aus dieser Phase resultieren zunächst eine Liste der Stakeholder und die so genannten Rohanforderungen, die noch keine feste Struktur haben.

Interpretation

In der nächsten Phase müssen die gesammelten Informationen interpretiert werden, um daraus genaue Anforderungen zu formulieren. Die gesammelten Rohanforderungen können sehr abstrakt und unkonkret sein. Deswegen ist es nötig sie für die weitere Verwendung zu strukturieren. Dazu werden sie zunächst klassifiziert, zum Beispiel in funktionale und nicht-funktionale Anforderungen. Funktionale Anforderungen sind all jene Anforderungen, die festlegen, was das Produkt leisten soll. Nicht-funktionale Anforderungen gehen darüber hinaus und legen die Qualität einer Leistung fest (zum Beispiel die maximale Reaktionszeit eines Systems).

Nach dieser Strukturierung werden ähnliche oder gleiche Anforderungen, Widersprüche und Zusammenhänge identifiziert und im Zweifel mit dem Kunden abgeklärt.

Negotiation

In der dritten Phase geht es darum, die zuvor festgestellten Konflikte zu lösen. Die Interessen der verschiedenen Stakeholder können sehr stark von einander abweichen, was der wohl häufigste Grund für einen Konflikt ist. Bei der Negotiation geht es darum, die Blickpunkte der verschiedenen Stakeholder genau zu betrachten und Konflikte zwischen diesen so gut wie möglich aufzulösen. Dazu müssen die Widersprüche und die Quellen aus denen diese resultieren den Kunden möglichst klar präsentiert werden. Dann können zum Beispiel die Stakeholder ihre Anforderungen priorisieren, um so einen Kompromiss untereinander zu finden. Für diese Phase sind besonders Workshops wichtig, an denen möglichst die verschiedenen Stakeholder teilnehmen.

Documentation

Alle gesammelten Informationen müssen so gut wie möglich dokumentiert werden. Dieser Prozess sollte von der ersten Phase an das ganze Projekt begleiten. Alle Anforderungen sollten unmissverständlich niedergeschrieben werden, da sie auch von anderen Personen verstanden werden müssen. Dazu sollten auch die Gründe für bestimmte Entscheidungen festgehalten werden. Eine Technik die hier zum Einsatz kommen kann ist das *Neurolinguistische Programmieren*. Ziel dieser Technik ist es, Formulierungen unmissverständlich zu machen und möglichst wenig Interpretationsspielraum zu lassen.

Validation/Verification

Bei der Validierung und Verifizierung wird überprüft, ob die dokumentierten Anforderungen korrekt sind. Es wird einerseits validiert, dass die Anforderungen den Vorstellungen des Kunden entsprechen und andererseits verifiziert, dass sie nicht den Rohanforderungen aus der Elicitation Phase widersprechen. Die Validierung erfolgt durch Analyse der Rohanforderungen und daraus gewonnenen Anforderungen und die Verifizierung durch Kundengespräche.

2.1.2. Requirements Management

Change Management

Das Change Management beginnt erst nach der Anforderungsanalyse und dient der Änderung von Anforderungen. Anforderungen können sich die ganze Dauer des Projekts über verändern und dieser Prozess sollte deswegen möglichst unkompliziert sein, um keine größeren Unterbrechungen zu verursachen. Änderungswünsche müssen an ein hierfür gebildetes Gremium - das *Change Control Board* - gerichtet wer-

den. Das Gremium sollte möglichst sowohl aus Kunden, als auch aus Entwicklern bestehen, damit Entscheidungen besser abgewogen werden können. Dieses Gremium beurteilt dann den Wunsch und entscheidet darüber. Jegliche Wünsche und Entscheidungen werden dokumentiert.

Tracing

Beim Tracing geht es um die Verfolgbarkeit der Änderungen von Anforderungen und deren Zusammenhängen. Es soll gewährleistet sein, dass zu jedem Zeitpunkt im Projekt die Entwicklung der Anforderungen und die Gründe dafür einsehbar sind. Diese Rückverfolgung der Anforderungen bis hin zur Elicitation nennt sich auch *Pre-tracing*. Außerdem soll durch *Post-tracing* ermöglicht werden, die aus einer Anforderung resultierenden Auswirkungen auf das entwickelte System, ebenfalls zu verfolgen. Anforderungen, Entscheidungen und andere Artefakte werden durch *Trace Links* verbunden, wodurch ein Graph entsteht. Dieser lässt sich zur Optimierung der Entscheidungs- und Implementierungsprozesse analysieren.

2.2. Videos im Requirements Engineering

Videos sind laut Ambler [1] der reichhaltigste Kommunikationsweg (siehe Abbildung 2.2). Das spricht dafür, dass sie auch im Requirements Engineering eine sinnvolle Ergänzung sein können.

Es gibt verschiedene Anwendungsmöglichkeiten für Videos im Requirements Engineering. Sie können zum Beispiel bei der Aufnahme und Analyse eines Workshops zum Einsatz kommen [15][16]. Die Entwickler können nicht immer am Workshop teilnehmen. Deswegen ist es wichtig den Workshop genau festzuhalten. Bei einem schriftlichen Protokoll können jedoch Informationen verloren gehen. Eine Videoaufnahme ist die Form der Dokumentation, die einer tatsächlichen Teilnahme am Workshop am nächsten kommt.

Hier soll es aber eher um den Einsatz mit dem Zweck gehen, Szenarien zu dem zu entwickelnden System darzustellen. *Scenario-based Requirements Engineering* ist eine Technik, bei der die Spezifikation von *Szenarien* unterstützt wird. Ein Szenario ist üblicherweise ein beispielhafter, erfolgreicher Durchlauf eines Use Cases. Die Szenarien können in verschiedenster Form festgehalten werden. Eine Möglichkeit sind Videos. Szenarien werden im Requirements Engineering zur *Vergegenwärtigung der Exploration des Designs, Anforderungserhebung* und *Validierung* eingesetzt [6]. Generell sind Vision Videos wohl aber vor allem der Validierung und Verifizierung, wie in 2.1.1 beschrieben, zuzuordnen. Nach der Anforderungserhebung werden von den Softwareentwicklern Videos erstellt. Diese sollen dem Kunden die Vision des Systems anschaulich vermitteln [7]. Der Kunde kann dann entscheiden, ob seine eigene Vision damit übereinstimmt und die geäußerten Anforderungen richtig verstanden

2. Grundlagen

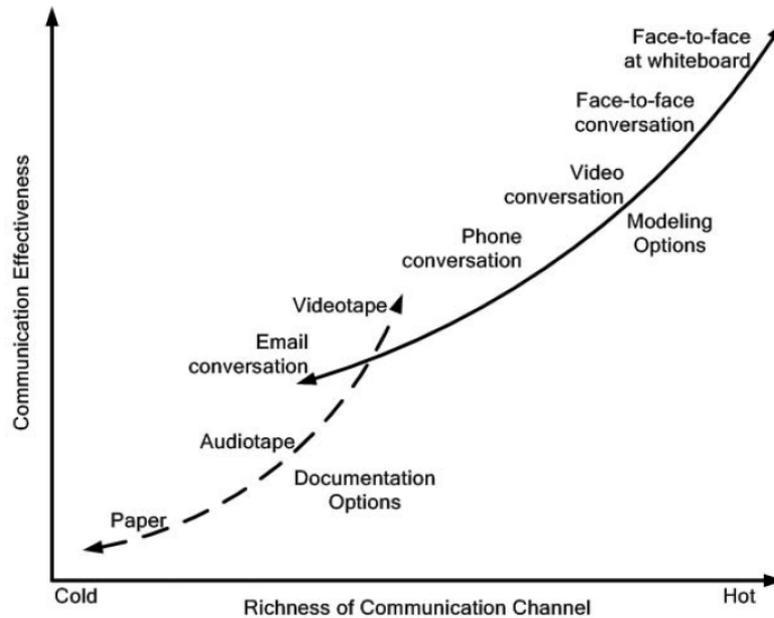


Abbildung 2.2.: Modes of Communication [1]

wurden. So können Missverständnisse aufgelöst werden und eventuell auch versteckte Anforderungen entdeckt werden, an die der Kunde zunächst nicht gedacht hat.

Laut Creighton et al. [7] gibt es fünf Hauptherausforderungen in der Verhandlung der Anforderungen:

1. Erwartungen: Die Erwartungen der Stakeholder an ein neues System können schwer zu definieren sein.
2. Änderungen: Stakeholder vertrauen auf altgewohnte Systeme. Dies kann zu Problemen bei der Einführung von neuen Ideen führen.
3. Kommunikation: Stakeholder können verschiedene Wahrnehmungen der Realität haben. Die Herausforderung besteht darin Anforderungen so zu formulieren, dass sie gleich verstanden werden und andere Perspektiven bewusst werden.
4. Sprache: Stakeholder können unterschiedliche Hintergründe haben. Daraus kann resultieren, dass sie unter dem gleichen Begriff etwas verschiedenes verstehen.
5. Vision: Die Stakeholder müssen von einer gemeinsamen Vision des Systems überzeugt werden. Die Schwierigkeit hier ist es, die Details ausreichend zu beschreiben und trotzdem die Essenz des Systems deutlich zu machen.

Der Einsatz von Videos kann in Betracht dieser Herausforderungen hilfreich sein, da Videos weniger abstrakt und unmissverständlicher sind, als formale Notationen

2. Grundlagen

[7][5]. Außerdem zeigen Videos viele kleinere Details, die vielleicht für eine Person selbstverständlich sind, bei denen aber eine zweite Person eine andere Vorstellung hatte.

Es gibt keine Regeln zur Aufnahme von Vision Videos, weshalb diese verschiedenste Form haben können. Sie können Animationen enthalten oder auch eine Art Mockup in Videoform sein [17]. Es können aber auch Realverfilmungen sein, worauf sich diese Arbeit konzentrieren wird. Die erstellten Aufnahmen können anschließend noch mit Text angereichert werden, oder wie Creighton et al. [7] es vorsehen, sogar eine gewisse Interaktivität erhalten, bei der der Zuschauer Elemente im Video auswählen und so genauere Informationen erhalten kann. Generell sind Vision Videos heutzutage noch keine verbreitete Technik im Requirements Engineering [3]. Es gibt aber bereits einige Arbeiten, die sich mit dem Einsatz von Videos beschäftigen. Auf diese wird in 2.3 genauer eingegangen.

2.3. Verwandte Arbeiten

In *Software Cinema—Video-based Requirements Engineering* beschreiben Creighton et al. [7] eine Technik zur Darstellung von Requirements in Videoform. In dieser Technik sollen die Videos jedoch bearbeitet werden, um textuelle Anforderungen mit einzubauen. Es soll eine gewisse Interaktivität geschaffen werden. Für diese Technik werden ein Videoproduzent und ein Analytiker gebraucht. Außerdem wird nicht genauer auf die Aufnahme des Videos eingegangen, sondern lediglich auf die Bearbeitung. Damit geht die Arbeit von Creighton et al. schon einen Schritt weiter im Prozess der Videoproduktion, als es hier vorgesehen ist.

In *Supporting Requirements Elicitation by Tool-Supported Video Analysis* untersuchen Karras et al. [16] ebenfalls die Benutzung eines Tools zur Unterstützung bei der Nutzung von Videos im Requirements Engineering. Hier handelt es sich jedoch nicht um Vision Videos, sondern um Aufnahmen von Workshops. Das Tool lässt den Nutzer Notizen mit einem bestimmten Zeitpunkt im Video verbinden. Dadurch kann bei Bedarf einfach auf die Stelle im Video zugegriffen werden. Dies ist im Gegensatz zu dem Thema dieser Arbeit nicht der Validierung zuzuordnen, sondern der Elicitation. Außerdem geht es auch hier weniger um die Videoproduktion an sich, sondern um die Nutzung der erstellten Aufnahmen.

In *Video as a By-Product of Digital Prototyping: Capturing the Dynamic Aspect of Interaction* analysieren ebenfalls Karras et al. [17] ein Tool zur Erstellung von Videos bei dem Durchlauf von Mockups. Es werden interaktive Prototypen erstellt und die Nutzung dieser wird aufgezeichnet. Eine Evaluation zeigte, dass die auf diese Weise erstellten Videos zu einem schnelleren Verständnis führen als statische Mockups. Hier soll auch ein Programm Softwareentwickler bei der Videoerstellung unterstützen, es handelt sich aber im Gegensatz zu dieser Arbeit nicht um Realver-

2. Grundlagen

filmungen.

In *Videos vs. Use Cases: Can Videos Capture More Requirements under Time Pressure?* führen Brill et al. [5] eine Studie zum Einsatz von Videos im Gegensatz zu Use Cases durch. Sie zeigen, dass Vision Videos vergleichbar oder besser dazu geeignet sind, um Missverständnisse in den Anforderungen zu vermeiden. Sie schreiben jedoch: "*We give no guidance for creating good videos – this remains future work.*" [5, S. 2] Hier schließt diese Arbeit thematisch an. Im nächsten Kapitel werden Guidelines zur Videoproduktion beschrieben.

3. Anforderungsermittlung

In diesem Kapitel geht es darum, die Anforderungen für ein Videorecorder-Tool, das den Nutzer bei der Erstellung von Videos unterstützt, zu erheben. Zunächst soll der Anwendungsfall dieses Tools genauer beschrieben werden, um den Kontext der Anforderungen besser verständlich zu machen. Anschließend werden Guidelines zur Videoproduktion zusammengetragen, welche die Basis für die Anforderungen bilden. Danach werden die Anforderungen, aufgeteilt in Basis-, Leistungs- und Begeisterungsanforderungen, aufgestellt.

3.1. Anwendungsfall

Wie zuvor beschrieben ist das Thema dieser Arbeit der Verifikation des Requirements Engineerings zuzuordnen. Die Nutzer des Systems bestehen aus einem Team aus Softwareentwicklern, welches sich am Anfang eines Projektes befindet. Die Softwareentwickler haben bereits Rohanforderungen erhoben und diese analysiert. Als nächstes wollen sie durch den Kunden verifizieren, dass diese richtig interpretiert wurden und ihre Vision des Systems mit der des Kunden übereinstimmt. Hierzu möchten sie ein oder mehrere Videos erstellen, in denen die aufgestellten Anforderungen zu erkennen sind. An dieser Stelle kommt das Videorecorder-Tool zum Einsatz. Das Tool läuft auf einem mobilen Gerät (Smartphone oder Tablet) mit integrierter (Front- und Rück-)Kamera, integriertem Mikrofon und Touchscreen. Es wird davon ausgegangen, dass Kamera und Mikrofon eine ausreichende Aufnahmequalität vorweisen. Das Team filmt damit das gewünschte Szenario (zum Beispiel mit Hilfe eines Papierprototyps). Eine Person bedient die Kamera und je nach Szenario werden eine oder mehrere Personen als Schauspieler eingesetzt. Beim Filmen unterstützt das Tool die Person an der Kamera dabei Guidelines zur Videoproduktion, welche zu diesem Zweck im nächsten Teil gesammelt werden sollen, einzuhalten. Nachdem die Videos aufgenommen wurden können sie nach Belieben bearbeitet und mit textuellen Informationen angereichert werden. Für diesen Schritt bietet das Tool keine direkte Hilfestellung. Anschließend können die Videos dem Kunden vorgespielt werden, um Missverständnisse und möglicherweise zusätzliche Anforderungen zu erkennen. Außerdem kann das Video möglicherweise zudem noch zum Zweck der Dokumentation genutzt werden.

3.2. Guidelines zur Videoproduktion

In diesem Abschnitt sollen verschiedene Guidelines zur Videoproduktion zusammengetragen werden. Solche Guidelines können sich je nach Art des Videos stark unterscheiden. Gerade bei Videos zu Unterhaltungszwecken gibt es keine festen Regeln, da es sich eher um eine Kunstform handelt. Es gibt bisher keine spezifischen Guidelines für Vision Videos, weshalb versucht werden muss, vergleichbare Arten zu betrachten. Bei Vision Videos handelt es sich um informative Videos, bei denen es vor allem darum geht, ein möglichst klares Verständnis zu schaffen. Damit lassen sich die zu betrachtenden Guidelines schon stark einschränken.

Massive Open Online Course (MOOC) Videos sind Videos, die als Teil von Online Kursen, besonders von Universitäten, zu Lehrzwecken angeboten werden. Hier existieren bereits Guidelines [18][19]. Außerdem herangezogen wurden die Guidelines der Association For Recorded Sound Collections (ARSC) [20] und den Universitäten Boise State [21], Leeds [22] und Massachusetts Institute of Technology [23] und das *Video Production Handbook* von Millerson und Owens [24]. Betrachtet wurden nur Guidelines in direktem Bezug auf die Aufnahme der Videos. Guidelines zur Auswahl von Equipment oder Nachbearbeitung wurden vernachlässigt, da diese tendenziell nicht zu "automatisieren" sind, beziehungsweise dies einen anderen Schritt der Videoproduktion betrifft. Die Verbleibenden Guidelines können grob in vier Kategorien eingeteilt werden: Bild, Ton, Perspektive/Bewegung und Dauer. Im Folgenden werden diese Kategorien und die dazugehörigen Guidelines genauer beschrieben.

Kategorie 1: Bild

Das Bild ist wahrscheinlich der wichtigste Aspekt eines Videos. Die darzustellende Szene sollte möglichst klar zu erkennen sein. Folgende Guidelines tragen dazu bei:

1. Das Bild sollte gut beleuchtet sein. [20][21][22][23][24]
Man sollte dazu an einem hellen Ort drehen und die Kamera nicht in Richtung eines Fensters halten.
2. Das Bild sollte gut fokussiert sein. [20]
3. Der Hintergrund sollte simpel und ordentlich sein. [21][23][24]
Es sollten keine Objekte im Hintergrund den Zuschauer von den relevanten Geschehnissen ablenken.

Kategorie 2: Perspektive/Bewegung

Generell sollte auch hier für möglichst geringe Ablenkung gesorgt werden. Das heißt unnötige Bewegung ist generell zu vermeiden.

3. Anforderungsermittlung

1. Die Kamera sollte so ruhig wie möglich gehalten werden. [20][21][24]
Die Kamera sollte nicht zittern und nicht mehr als nötig bewegt werden. Wenn die Kamera bewegt wird, sollte dies möglichst ruhig und langsam geschehen.
2. Zoom sollte vermieden werden. [20][22]
Zoom sollte nur dazu genutzt werden gezielt die Aufmerksamkeit auf etwas zu richten. Generell ist er aber eher zu vermeiden, da es sich um keine für den Menschen natürliche Bewegung handelt und er deswegen ablenkend wirken kann.
3. Das Video sollte im Landscape-Format aufgenommen werden (16:9 oder natives Seitenverhältnis). [20][22]
Videos sollten generell bevorzugt im Querformat aufgenommen werden, da dies dem menschlichen Sichtfeld besser entspricht.
4. Schräge Perspektiven (Dutch Tilt) sollten vermieden werden. [22]
Extreme Perspektiven, wie der *Dutch Tilt* werden dazu genutzt beim Betrachter einen unwirklichen oder desorientierenden Eindruck zu vermitteln. Dies ist im Fall von informativen Videos eher kontraproduktiv.

Kategorie 3: Audio

Der Ton sollte, wie das Bild, möglichst klar und verständlich sein.

1. Es sollte wenig/keine Umgebungsgeräusche geben. [20]
2. Es sollte möglichst ein externes Mikrofon verwendet werden. [22][23]
Dies betrifft zwar nicht die Aufnahme an sich, sondern das Equipment, es ist aber trotzdem zu berücksichtigen die Verwendung eines externen Mikrofons zu ermöglichen.
3. Es sollte nicht zu langsam gesprochen werden. [18]
Guo et al. [18] zeigten, dass schnelles Sprechen eine höhere Aufmerksamkeit beim Zuschauer bewirkt. In der Studie lieferten 185-254 Wörter pro Minute die besten Ergebnisse.

Kategorie 4: Dauer

1. Das Video sollte nicht zu lang sein. [18][19]
Guo et al. [18] und Lackner et al. [19] stimmen darüber überein, dass kürzere Videos besser sind, was die Aufrechterhaltung der Aufmerksamkeit des Betrachters angeht. Laut Guo et al. sollte es idealerweise kürzer als 6 Minuten sein. Bei Lackner et al. sind es 5-10 Minuten.
2. Es sollten 5-10 Sekunden vor und nach jeder Szene aufgezeichnet werden. [23]
Dies dient als eine Art Puffer um sicherzugehen, dass das Video nicht zu spät

3. Anforderungsermittlung

beginnt oder zu früh aufhört.

3. Schnelle Schnitte sollten vermieden werden. [22]
Dies betrifft zwar eher die Bearbeitung, aber kann auch schon bei der Aufnahme einzelner Szenen berücksichtigt werden.

3.3. Basisanforderungen

Basisanforderungen sind die grundlegenden Anforderungen an ein System, die als selbstverständlich angesehen werden. Nur bei Nichterfüllung fallen sie negativ auf.

Zu der Erhebung der Anforderungen wurden die Features von sieben Android Applikationen zur Videoaufnahme aus dem Google Play Store betrachtet. Es wurden nur Features in Bezug auf Videos beachtet. Die Aufnahme von Fotos und damit zusammenhängende Features wurden vernachlässigt. Die Features sind in Tabelle 3.1, nach Häufigkeit sortiert, aufgelistet. Sie wurden in drei Kategorien eingeteilt. Jene Features die in mindestens sechs von sieben Anwendungen vorhanden waren, werden als Basisanforderungen angesehen, da sie in (fast) jedem Videorecorder vorhanden sind und somit zu erwarten sind. Auf die Features aus der zweiten Kategorie wird später eingegangen. Die restlichen Features werden vernachlässigt, da es hier zunächst um die Features gehen soll, welche für einen Videorecorder wichtig sind. Die Basis-Features sollen im Folgenden genauer betrachtet und für den hier vorhandenen Anwendungsfall, unter Einbezug der Guidelines zur Videoproduktion (siehe Abschnitt 3.2), priorisiert werden. Die Priorisierung erfolgt auf einer Skala von “sehr niedrig” bis “sehr hoch”.

Priorisierung für den Anwendungsfall

Videoaufnahme

Die Aufnahme von Videos ist die grundlegende Funktion eines Videorecorders. Dies gilt auch für den Anwendungsfall, weshalb dieses Feature von höchster Priorität ist.

Resultierende Anforderung:

[R101] Das System muss dem Nutzer die Möglichkeit bieten Videos aufzuzeichnen.

Priorität: sehr hoch

Kameraauswahl

Im Fall der betrachteten Androidanwendungen betrifft die Kameraauswahl die Front- und Rückkamera. Die Frontkamera wird üblicherweise dazu genutzt sich selber aufzunehmen. Es ist zu vermuten, dass im Fall von Vision Videos für gewöhnlich eine

3. Anforderungsermittlung

Feature	Häufigkeit	Kategorie
Videoaufnahme	7/7	1
Kameraauswahl	7/7	
Zoom	7/7	
Blitz	7/7	
Tap to focus	6/7	
Auswahl der Auflösung	6/7	
Auswahl von Seitenverhältnissen	5/7	2
Raster (Hilfslinien)	5/7	
Zeitraffer	5/7	
Pausieren der Aufnahme	4/7	
Manuelle Belichtung	4/7	
Filter/Effekte	3/7	3
Auswahl des Mikrofons	3/7	
Neigungsanzeige	3/7	
Weißabgleich	3/7	
Einstellung einer Maximaldauer	2/7	
Zeitlupe	1/7	
Ohne Ton aufnehmen	1/7	
Geotagging	1/7	

Tabelle 3.1.: Features von Kamera-Apps

Person zum Filmen vorhanden ist und somit eher die Rückkamera von Nutzen ist. Die Auswahl der Kamera könnte jedoch auch den Einsatz einer externen Kamera ermöglichen. Insgesamt bietet dieses Feature also eine etwas größere Flexibilität, ist aber nicht von höchster Priorität.

Resultierende Anforderung:

[R102] Das System muss dem Nutzer die Möglichkeit bieten die Kamera zur Aufzeichnung auszuwählen.

Priorität: mittel

Zoom

Zoom ist definitiv eine Basisanforderung eines Videorecorders, die immer vorhanden ist. An dieser Stelle widerspricht der Basisanforderung aber eine Guideline aus Abschnitt 3.2, die besagt, dass Zoom vermieden werden soll. Auf dieses Feature wird noch im nächsten Abschnitt, in Bezug auf die Leistungsanforderungen, genauer eingegangen.

3. Anforderungsermittlung

Blitz

Aus den Guidelines zur Videoproduktion ging hervor, dass Videos in gut beleuchteter Umgebung aufgenommen werden sollten. Die Beleuchtung durch den Blitz ist eine Möglichkeit, um bei Dunkelheit die Lichtverhältnisse zu verbessern. Ob der Blitz eines üblichen mobilen Gerätes ausreicht, um bei Dunkelheit eine gute Videoqualität zu erreichen ist jedoch fraglich. In Einzelfällen könnte es aber ein nützliches Feature sein.

Resultierende Anforderung:

[R103] Das System sollte dem Nutzer die Möglichkeit bieten den Blitz während der Aufnahme zu aktivieren.

Priorität: mittel

Tap to focus

Es wird davon ausgegangen, dass die Kamera bereits einen integrierten Autofokus besitzt. Eine zusätzliche Möglichkeit durch Tippen auf einen Bereich im Bild zu fokussieren wäre dennoch sinnvoll für den Fall, dass der Autofokus nicht auf das gewünschte Objekt fokussiert. Außerdem wird bei diesem Feature zusammen mit dem Fokus auch die Belichtung angepasst.

Resultierende Anforderungen:

[R104] Das System muss dem Nutzer die Möglichkeit bieten den Fokus durch tippen auf das Bild anzupassen.

Priorität: hoch

[R105] Das System muss dem Nutzer die Möglichkeit bieten die Belichtung durch tippen auf das Bild anzupassen.

Priorität: hoch

Auswahl der Auflösung

Grundsätzlich sollten Vision Videos eine möglichst hohe Bildqualität aufweisen, damit die Objekte möglichst klar zu erkennen sind. Der einzige Grund die Qualität niedriger einzustellen, wäre der geringere Speicherverbrauch. Da aber mit eher kurzen Videos im Bereich von wenigen Minuten zu rechnen ist, ist dieses Feature für den Anwendungsfall mit einer niedrigen Priorität einzuordnen.

Resultierende Anforderung:

[R106] Das System muss sollte dem Nutzer die Möglichkeit bieten die Auflösung für das aufzunehmende Video auszuwählen.

Priorität: niedrig

3.4. Leistungsanforderungen

Leistungsfaktoren sind dem Kunden bewusste Anforderungen, die je nach Erfüllungsgrad sowohl für Zufriedenheit als auch für Unzufriedenheit sorgen können.

In diesem Abschnitt sollen einerseits die Features aus Kategorie 2 der betrachteten Android-Applikationen untersucht werden. Andererseits sollen Konzepte zur Automatisierung der Guidelines aus Abschnitt 3.2 erstellt werden und dazu passende Anforderungen aufgestellt werden.

3.4.1. Anforderungen aus betrachteten Anwendungen

Auswahl von Seitenverhältnissen

Eine häufiges Feature aus den betrachteten Anwendungen ist die Auswahl der Seitenverhältnisse. Laut Guidelines sollten die Seitenverhältnisse 16:9 oder den nativen Verhältnissen des Gerätes entsprechen. Dieses Feature wird also nicht benötigt.

Raster

Ein optionales Raster kann hilfreich sein, um leichter eine gute Bildkomposition zu erhalten. Eine Person ohne Erfahrung kann möglicherweise mit dem Raster in dieser Hinsicht weniger anfangen als ein professioneller Videoproduzent. Das Raster kann außerdem aber dafür genutzt werden die Linien parallel zum Horizont zu halten und somit sicherzugehen keine schräge Perspektive zu erhalten, was der 4. Guideline aus der Kategorie Perspektive/Bewegung entspricht (siehe Abschnitt 3.2). Im Bezug auf einen Videorecorder, der den Nutzer unterstützen soll könnte es also ein sinnvolles Feature sein.

Resultierende Anforderung:

[R201] Das System muss dem Nutzer die Möglichkeit bieten Hilfslinien zur Orientierung über das Bild zu legen.

Priorität: hoch

Zeitraffer

Zeitraffer wird eingesetzt, um einen längeren Prozess in kurzer Zeit zu zeigen. Bei Vision Videos ist es im Normalfall sinnvoller den Prozess möglichst real darzustellen. In den betrachteten Anwendungen wurde bei Benutzung von Zeitraffer das Video stumm geschaltet. Im nächsten Abschnitt wird noch genauer auf die Beschleunigung des Videos inklusive Audio eingegangen.

Pausieren der Aufnahme

Durch die Möglichkeit die Aufnahme zu pausieren kann der Nutzer Schnitte einbauen, ohne das Video im Nachhinein zu bearbeiten. Dies ist hilfreich für den gesamten Prozess der Videoproduktion.

Resultierende Anforderung:

[R202] Das System muss dem Nutzer die Möglichkeit bieten die Aufnahme zu pausieren und fortzusetzen.

Priorität: hoch

Manuelle Belichtung

Wie zuvor beim Fokus wird davon ausgegangen, dass eine automatische Belichtung durch die Kamera vorhanden ist. Eine weitere Option zum Anpassen der Belichtung ist bereits in dem Feature "Tap to focus" integriert. Eine zusätzliche manuelle Belichtungsoption ist nicht notwendig, da versucht wird ein natürliches Bild zu erzeugen.

Resultierende Anforderung:

[R203] Das System sollte dem Nutzer die Möglichkeit bieten die Belichtung manuell anzupassen.

Priorität: sehr niedrig

3.4.2. Konzepte zur Automatisierung von Guidelines

In diesem Abschnitt sollen Anforderungen aus den Guidelines zur Videoproduktion abgeleitet werden. Diese Anforderungen erhalten alle eine hohe Priorität, da sie die Essenz des Tools ausmachen und keine genaueren Anhaltspunkte dazu vorhanden sind, welche Guideline wichtiger ist als eine andere. Sie werden im Folgenden erneut in die Kategorien aus Abschnitt 3.2 eingeteilt.

Kategorie 1: Bild

Beleuchtung

Das Video sollte in einer gut beleuchteten Umgebung aufgenommen werden. Abgesehen vom Einsatz von Blitz zur Beleuchtung lässt sich dies vom System nicht beeinflussen und der Blitz kann tendenziell eine natürliche Beleuchtung nicht ersetzen. Deswegen erscheint es am sinnvollsten den Nutzer auf eine zu geringe Beleuchtung aufmerksam zu machen.

3. Anforderungsermittlung

Resultierende Anforderung:

[R204] Das System muss fähig sein ein schlecht beleuchtetes Bild zu erkennen und den Nutzer darauf hinweisen.

Fokussierung

Das Bild sollte gut fokussiert sein. Dies wird bereits durch die Basisanforderungen abgedeckt, da ein Videorecorder einen Autofokus besitzen sollte und somit die Kamera bereits automatisch versucht ein gut fokussiertes Bild zu generieren. Idealerweise sollte zusätzlich auch manuell mit "Tap to focus" fokussiert werden können.

Hintergrund

Der Hintergrund des Bildes sollte ordentlich sein und nicht ablenken. Diese Guideline scheint schwierig umzusetzen zu sein. Das System müsste erkennen, ob der Hintergrund unordentlich ist. Möglicherweise ließe sich dies mit Hilfe von Deep Learning bewerkstelligen, aber es erscheint nahezu unmöglich zu erkennen was zum Hintergrund gehört und was Teil der Szene ist oder ab wann ein Hintergrund unordentlich ist.

Kategorie 2: Perspektive/Bewegung

Ruhige Kamera

Die Kamera sollte möglichst ruhig gehalten werden. Idealerweise könnte das System das Bild automatisch stabilisieren. Dies würde aber das Speichern des Videos stark verlangsamen, die Qualität verringern und am Rand müsste ein Teil des Bildes abgeschnitten werden. Insgesamt erscheint diese Maßnahme für den Zweck übertrieben, da die meisten Vision Videos wahrscheinlich nicht in einer Umgebung aufgenommen werden, in der ein Zittern der Kamera nicht zu vermeiden ist. Eine einfachere Alternative wäre das Zittern zu registrieren und den Nutzer darauf hinzuweisen. So müsste der Nutzer zwar selber für eine ruhige Kamera sorgen, es würden aber keine weiteren Nachteile entstehen.

Resultierende Anforderung:

[R205] Das System muss fähig sein ruckartige Bewegungen zu erkennen und den Nutzer darauf hinweisen.

Zoom

Zoom sollte vermieden werden. Der einfachste Weg diese Guideline umzusetzen wäre keinen Zoom zu implementieren. Wenn in manchen Fällen aber doch Zoom benötigt

3. Anforderungsermittlung

ist, wäre dies vielleicht problematisch. Ein Kompromiss wäre es den Zoom standardmäßig zu deaktivieren, dem Nutzer aber eine Einstellungsmöglichkeit zu geben, um diesen zu aktivieren.

Resultierende Anforderungen:

[R206] Das System muss dem Nutzer die Möglichkeit bieten zu zoomen.

[R206.1] Das System sollte dem Nutzer die Möglichkeit bieten Zoom zu aktivieren und zu deaktivieren.

[R206.2] Standardmäßig sollte der Zoom deaktiviert sein.

Landscape

Videos sollten im Landscape-Format aufgenommen werden. Dies lässt sich umsetzen, indem die Anwendung nur auf dieses Format ausgerichtet wird und bei einer Drehung nicht in das Hochformat wechselt. Dies hält zwar nicht zwingend den Nutzer davon ab im Hochformat zu filmen, aber sollte reichen um zu verdeutlichen, dass dieses Format bevorzugt werden sollte.

Resultierende Anforderungen:

[R207] Das System sollte für das Landscape-Format optimiert sein.

[R207.1] Das System sollte keinen Wechsel ins Hochformat ermöglichen.

Dutch Tilt

Perspektiven mit starker Schräglage sollten vermieden werden. Dazu könnte das System die Neigung des Geräts überprüfen und den Nutzer darauf hinweisen, wenn diese eine festgelegte Grenze überschreitet.

Resultierende Anforderung:

[R208] Das System muss fähig sein eine starke Neigung zu erkennen und den Nutzer darauf hinweisen.

Kategorie 3: Audio

Umgebungsgeräusche

Es sollte bei der Aufnahme möglichst wenig/keine Umgebungsgeräusche geben. Das System könnte überprüfen, ob es konstante Geräusche gibt, die eine festgelegte Lautstärkegrenze überschreitet und den Nutzer darauf hinweisen.

Resultierende Anforderung:

[R209] Das System muss fähig sein konstante Umgebungsgeräusche zu registrieren und den Nutzer darauf hinweisen.

3. Anforderungsermittlung

Externes Mikrofon

Es wird empfohlen für die Aufnahme ein externes Mikrofon zu verwenden, um eine bessere Tonqualität zu erhalten. Es sollte möglich sein aufzunehmen, auch wenn kein externes Mikrofon vorhanden ist, aber es ist sinnvoll die Nutzung zu ermöglichen.

Resultierende Anforderung:

[R210] Das System sollte dem Nutzer die Möglichkeit bieten das Mikrofon zur Aufnahme auszuwählen.

Sprechgeschwindigkeit

Falls in dem Video gesprochen wird, sollte die Sprechgeschwindigkeit nicht zu langsam sein. Ein Feature aus den betrachteten Android-Applikationen war der Zeitraffer, bei dem allerdings der Ton stumm geschaltet wurde. Eine Lösung könnte ein Zeitraffer inklusive Audio sein. Das System könnte durch eine Spracherkennung die WPM (words per minute) erkennen und diese auf einen bestimmten Wert anpassen. Es wäre allerdings ein Nachteil, dass das Video in dem Fall nicht mehr Originalgeschwindigkeit hätte oder, würde man das Video nicht beschleunigen, Ton und Video nicht mehr synchron wären. Dadurch scheint diese Lösung nicht praktikabel zu sein.

Kategorie 4: Dauer

Gesamtlänge

Die Länge des Videos sollte eine bestimmte Dauer nicht überschreiten, um die Aufmerksamkeit des Zuschauers nicht zu verlieren. Laut Guo et al. [18] handelt es sich dabei um 6 Minuten. Hier wäre eine Möglichkeit eine maximale Aufnahmedauer festzulegen. Es wäre eventuell aber ungünstig, wenn die Aufnahme von alleine stoppt und der Nutzer noch weiter filmen wollte (eventuell mit der Intention das Video danach zuzuschneiden). Es wäre also wohl sicherer den Nutzer nur auf das Überschreiten einer empfohlenen Zeit hinzuweisen.

Resultierende Anforderung:

[R211] Das System muss fähig sein eine Überschreitung einer Aufnahmedauer von 6 Minuten zu erkennen und den Nutzer darauf hinweisen.

Puffer

Es sollen vor und nach jeder Szene 5-10 Sekunden aufgenommen werden. Dies ließe sich eventuell realisieren, indem die Kamera dauerhaft läuft, aber immer nur die letzten 5-10 Sekunden speichert, solange keine Aufnahme durch den Nutzer gestartet wurde. So könnte noch ein Puffer vor und nach der vom Nutzer gestarteten

3. Anforderungsermittlung

Aufnahme aufgezeichnet werden. In diesem Fall müsste die Anwendung aber auch das Zuschneiden der Videos ermöglichen, um den richtigen Start- und Endpunkt anschließend wählen zu können. Das Konzept wird noch einmal in Abbildung 3.1 in Form eines Zeitstrahls verdeutlicht.

Resultierende Anforderungen:

[R212] Das System sollte dauerhaft Video aufzeichnen.

[R212.1] Wenn keine Aufnahme durch den Nutzer gestartet wurde, sollte das System die Aufnahmen bis auf die letzten 5-10 Sekunden löschen.

[R212.2] Das System sollte bei Aufnahmestart durch den Nutzer 5-10 Sekunden Puffer vor dem Start und nach dem Ende der Aufnahme speichern.

[R213] Das System muss dem Nutzer die Möglichkeit bieten aufgenommene Videos zu schneiden.

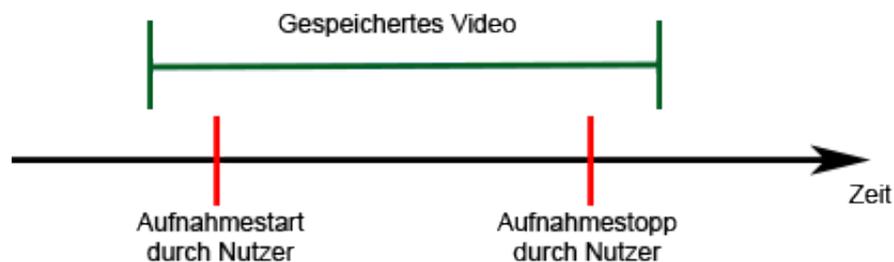


Abbildung 3.1.: Puffer

Schnitte

Schnelle Schnitte sollen vermieden werden. Die Anwendung könnte verhindern, dass bei einer sehr kurzen bisherigen Aufnahmedauer die Aufnahme pausiert wird, da ein Pausieren der Aufnahme zu einem Schnitt im Video führt. Es kann kein Einfluss darauf genommen werden, ob der Nutzer in der Nachbearbeitung schnelle Schnitte einfügt, aber so können zumindest schnelle Schnitte während der Aufnahme verhindert werden. Als minimale Aufnahmezeit werden hierfür 5 Sekunden gewählt. 5 Sekunden sollten nicht zu lang sein, da im Zweifel ein Paar Sekunden Puffer nicht schaden, aber lang genug um hektische Schnitte zu verhindern.

Resultierende zusätzliche Bedingung:

[R202] Wenn eine Aufnahmezeit von 5 Sekunden für das aktuelle Video überschritten ist, muss das System dem Nutzer die Möglichkeit bieten die Aufnahme zu pausieren.

3.5. Begeisterungsanforderungen

Als Begeisterungsfaktoren werden Anforderungen bezeichnet, die über die Anforderungen des Kunden hinausgehen. Da es hier keinen Kunden gibt, wird die Bezeichnung hier für Ideen verwendet, welche über die Basisanforderungen eines Videorecorders und die Guidelines zur Videoproduktion hinausgehen. Hier wird eine Idee zur Strukturierung der Videos beschrieben.

Szenario Struktur

Die Idee besteht darin, die Vision Videos mit der bereits vorhandenen Technik des *Scenario-based Requirements Engineering* zu verbinden. Ein Szenario ist üblicherweise ein erfolgreicher Durchlauf eines Use Cases, der in einzelne Schritte aufgeteilt ist. Man könnte zu jedem einzelnen Schritt des Szenarios ein kurzes Video aufnehmen und diese Videos dann zusammenfügen. Dies hätte vor allem den Vorteil, dass es einfacher wäre bei einer Änderung des Use Cases, auch das dazugehörige Video anzupassen. Man könnte Schritte vertauschen oder nur einzelne Schritte ersetzen. Außerdem können in ähnlichen Szenarien identische Schritte vorkommen. Diese müssten dann nicht mehrfach aufgenommen werden. Durch diese Struktur würde die Anwendung einen stärkeren Bezug zum Requirements Engineering erhalten.

Resultierende Anforderungen:

[R301] Das System muss dem Nutzer die Möglichkeit bieten ein Video in mehrere Schritte zu strukturieren.

[R301.1] Das System muss fähig sein mehrere Videos zusammenzufügen.

[R301.2] Das System muss dem Nutzer die Möglichkeit bieten die Reihenfolge der Videos zu ändern.

[R301.3] Das System muss dem Nutzer die Möglichkeit bieten einzelne Videos zu löschen.

[R301.4] Das System muss dem Nutzer die Möglichkeit bieten einzelne Videos zu ersetzen.

4. Prototypische Umsetzung

Im Rahmen dieser Arbeit wurde ein prototypischer Videorecorder auf Basis der zuvor erhobenen Anforderungen erstellt.

Im Folgenden wird dieser Prototyp als **ViViRecorder** (Vision Video Recorder) bezeichnet.

4.1. Technische Grundlagen

Der ViViRecorder wurde mit der Programmiersprache Java für ein Microsoft Surface entwickelt. Grundsätzlich funktioniert das Programm durch die Plattformunabhängigkeit von Java auch auf anderen Geräten, es ist aber nicht sichergestellt, dass die volle Funktionalität hier vorhanden ist. Als Ausgangspunkt für den ViViRecorder wurde der Videorecorder aus dem von Karras entwickelten Requirements Video Analyzer (ReqVidA) genutzt [25]. Zur Erstellung der grafischen Benutzeroberfläche wurde das JavaFX Framework verwendet. Für den Zugriff auf Kamera und Mikrofon wurde JavaCV, eine Java-Schnittstelle zu OpenCV und FFmpeg benutzt. Außerdem wurde für den Zugriff auf die Sensorik des Microsoft Surfaces jni4all verwendet, um ein Java Native Interface (JNI) zu einer C#-Library herzustellen. Für Nutzerdaten zu Szenarien (Name, Reihenfolge der Schritte etc.) wird eine SQLite-Datenbank erstellt, um diese bei Neustart der Anwendung nicht zu verlieren.

4.2. Eingrenzung der implementierten Anforderungen

Es war zeitlich nicht möglich im Rahmen dieser Arbeit alle aufgestellten Anforderungen zu implementieren. Hier soll die Auswahl der implementierten Anforderungen begründet werden.

Einige Anforderungen waren bereits durch Karras Videorecorder [25] erfüllt: Es konnten bereits Videos aufgenommen werden. Außerdem war eine Auswahl von Kamera und Mikrofon möglich. Somit waren Anforderungen [R101], [R102] und [R210] bereits umgesetzt. Da das Microsoft Surface keinen integrierten Blitz besitzt schied [R103] aus.

Es wurde sich auf die Basisanforderungen, die Leistungsanforderungen mit mindestens mittlerer Priorität, sowie die aus den Guidelines resultierenden Anforder-

4. Prototypische Umsetzung

rungen und die eine Begeisterungsanforderung zur Szenario-Struktur konzentriert. Auch diese Anforderungen mussten aber noch ein wenig eingegrenzt werden. Bei Tap to focus ([R104], [R105]) handelte es sich um eine Basisanforderung. Microsoft bietet eine Möglichkeit zur Implementierung dieses Features für Universal Windows Platform (UWP) Apps über die FocusControl-Klasse aus dem Namespace Windows.Media.Devices an. Dies funktioniert jedoch nur in den Programmiersprachen C++ und C#. JavaCV enthielt keine Möglichkeit den Fokus der Kamera zu verstellen. Unter diesen Umständen erschien die Implementierung dieses Features zu aufwändig, da für einen Prototypen der Autofokus genügen sollte. Des Weiteren wurden die Anforderungen zum Puffer ([R212],[R213]) aus zeitlichen Gründen vernachlässigt. Zoom wurde nicht implementiert, womit zumindest die Verfeinerung [R206.2] erfüllt ist, da standardmäßig der Zoom nicht aktiv sein soll. Die zusätzliche Einstellung, um den Zoom zu aktivieren erschien für einen Prototypen jedoch nicht notwendig, da laut Guidelines Zoom vermieden werden soll und die Erfüllung der Guidelines im Fokus steht. Das Programm ist außerdem auf das Landscape-Format ausgelegt. Es wird jedoch nicht verhindert, dass die Anwendung sich dreht, da es nicht trivial ist den Wechsel der Bildschirmausrichtung einer Java-Anwendung auf Windows zu verhindern. Da sie aber nicht für das Hochformat optimiert ist, ist sie so kaum benutzbar. Somit ist Anforderung [R207] auch erfüllt, allerdings ohne die Verfeinerung [R207.1]. Im Folgenden sind noch einmal die übrigen Anforderungen aufgelistet, auf die sich konzentriert wurde. Die Umsetzung dieser Anforderungen wird im nächsten Abschnitt genauer erläutert.

[R201]	Das System muss dem Nutzer die Möglichkeit bieten Hilfslinien zur Orientierung über das Bild zu legen.
[R202]	Wenn eine Aufnahmezeit von 5 Sekunden für das aktuelle Video überschritten ist, muss das System dem Nutzer die Möglichkeit bieten die Aufnahme zu pausieren.
[R204]	Das System muss fähig sein ein schlecht beleuchtetes Bild zu erkennen und den Nutzer darauf hinweisen.
[R205]	Das System muss fähig sein ruckartige Bewegungen zu erkennen und den Nutzer darauf hinweisen.
[R208]	Das System muss fähig sein eine starke Neigung zu erkennen und den Nutzer darauf hinweisen.
[R209]	Das System muss fähig sein konstante Umgebungsgeräusche zu registrieren und den Nutzer darauf hinweisen.
[R211]	Das System muss fähig sein eine Überschreitung einer Aufnahmedauer von 6 Minuten zu erkennen und den Nutzer darauf hinweisen.
[R301]	Das System muss dem Nutzer die Möglichkeit bieten ein Video in mehrere Schritte zu strukturieren.
[R301.1]	Das System muss fähig sein mehrere Videos zusammenzufügen.
[R301.2]	Das System muss dem Nutzer die Möglichkeit bieten die Reihenfolge der Videos zu ändern.
[R301.3]	Das System muss dem Nutzer die Möglichkeit bieten einzelne Videos zu löschen.
[R301.4]	Das System muss dem Nutzer die Möglichkeit bieten einzelne Videos zu ersetzen.

Tabelle 4.1.: Anforderungen

4.3. Design- und Entwurfsentscheidungen

In diesem Abschnitt sollen die implementierten Features sowie die dazugehörigen Design- und Entwurfsentscheidungen genauer beschrieben werden. In Abbildung 4.1 ist für einen ersten Überblick die Hauptansicht der Anwendung zu sehen. Der Videorecorder besteht zunächst aus einem Livestream der Kamera. An der rechten Seite befinden sich drei ToggleButtons zum Starten, Stoppen und Pausieren der Aufnahme und zum Ein- und Ausschalten eines Rasters ([R201]). Das Raster bietet eine optionale Hilfe, um Anhaltspunkte für die Perspektive zu erhalten. Es ist weiß mit schwarzem Schatten, um auf jedem Hintergrund erkennbar zu sein. Außerdem ist es leicht transparent, damit nichts verdeckt wird und es nicht zu sehr ablenkt. Unter dem Livestream in der Mitte befinden sich Auswahlmöglichkeiten für Kamera und Mikrofon. Auf die zusätzlichen Funktionen wird im Weiteren genauer eingegangen.

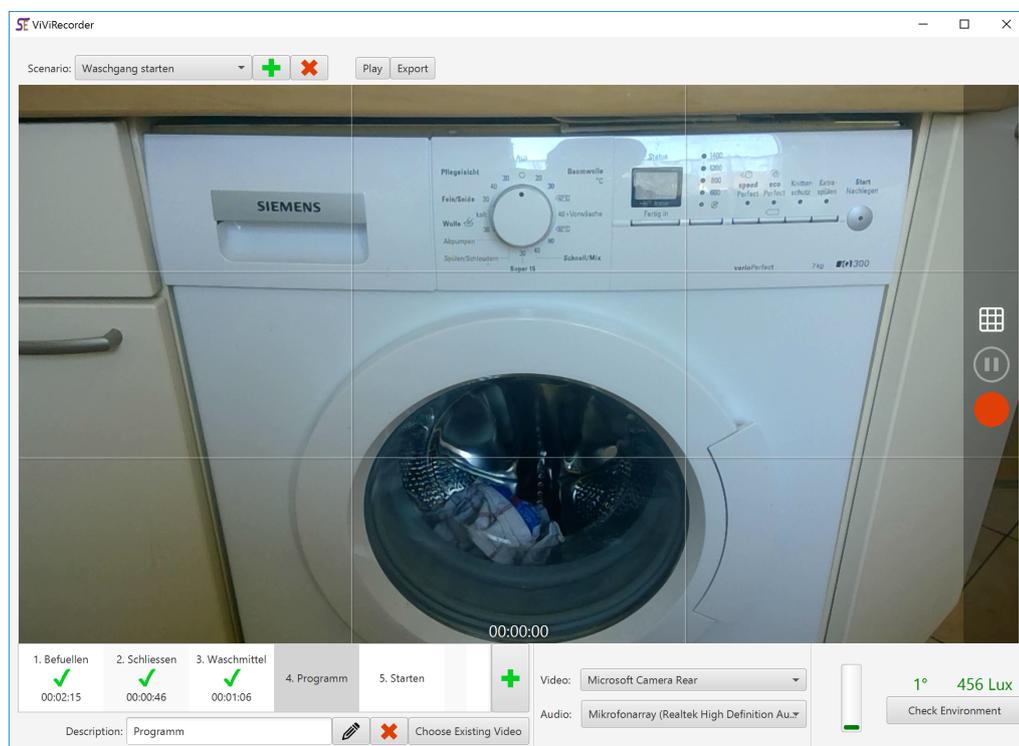


Abbildung 4.1.: Hauptansicht des ViViRecorders

4.3.1. Szenario Struktur

Die Videos lassen sich in einzelne Schritte eines Szenarios einteilen ([R301]). Dafür lässt sich zunächst oben, über dem Livestream (siehe Abbildung 4.1), ein Szenario auswählen bzw. hinzufügen oder löschen. Unten links werden dann die einzelnen Schritte des gewählten Szenarios angezeigt. Hier lassen sich Schritte hinzufügen, um-

4. Prototypische Umsetzung

benennen, löschen und die Reihenfolge der Schritte durch Drag and Drop anpassen. Wenn nun ein Schritt ausgewählt ist, kann entweder ein Video zu diesem Schritt in dem Programm aufgenommen, oder ein vorhandenes Video ausgewählt werden. Ist ein Video aufgenommen beziehungsweise ausgewählt, so wird ein grüner Haken unter dem Schritt angezeigt. Um ein neues Video für einen Schritt aufzunehmen muss der Schritt ausgewählt werden und einfach erneut aufgenommen oder ein Video ausgewählt werden. So kann die Reihenfolge der Schritte leicht angepasst oder ein einzelner Schritt ausgetauscht werden, wenn das Szenario sich verändert. Außerdem können bei ähnlichen Szenarien Schritte mehrfach verwendet werden, ohne dass sie erneut aufgenommen werden müssen. Die Strukturierung des Videos soll mit diesem Aufbau möglichst simpel sein und dem Nutzer keine große zusätzliche Arbeit bereiten.

4.3.2. Videoplayer und Export

In Abbildung 4.2 ist der Videoplayer zu sehen, welcher sich bei drücken des Play-Buttons über dem Livestream (siehe Abbildung 4.1) öffnet. Hier werden die bereits aufgenommenen Videos in ihrer festgelegten Reihenfolge abgespielt, um dem Nutzer eine Vorschau über den aktuellen stand des Szenarios zu geben. Links neben dem Videoplayer befindet sich eine Liste der Schritte. Durch klicken auf einen Schritt wird das gesamte Video von diesem Schritt aus abgespielt, oder bei deaktiviertem Autoplay auch nur der einzelne Schritt, falls der Nutzer diesen genauer betrachten möchte.

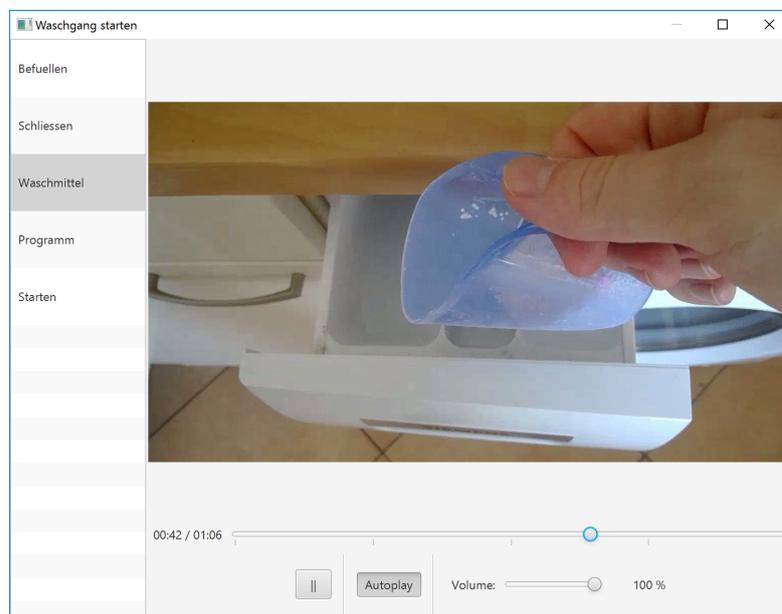


Abbildung 4.2.: Videoplayer

Ist der Nutzer zufrieden mit der Aufnahme, so kann er das Video über den Export-Button, direkt neben dem Play-Button, exportieren, um eine einzelne MP4-Datei zu dem Szenario zu erhalten ([R301.1]).

4.3.3. Überprüfung der Umgebung durch Sensorik

Die Überprüfung der Umgebung beinhaltet eine Überprüfung von Beleuchtung ([R204]), Umgebungsgeräuschen ([R209]), dem Winkel, in dem das Gerät gehalten wird ([R208]) und ob das Gerät ruhig gehalten wird ([R205]). Das Programm soll den Nutzer auf diese Aspekte aufmerksam machen und so dafür sorgen, dass die zugehörigen Guidelines besser eingehalten werden. Zur Überprüfung der genannten Aspekte wurden Lichtsensor, Mikrophon, Ausrichtungssensor und Gyrometer verwendet. Der Lichtsensor befindet sich an der Vorderseite des Surfaces neben der Frontkamera und gibt einen Wert zur Beleuchtungsstärke in Lux aus.

Mit dem Mikrophon wird ein Wert in Pascal gemessen, welcher hier für bessere Vergleichswerte in Dezibel umgerechnet wird.

Der Ausrichtungssensor liefert eine Quaternion, welche die Drehung des Gerätes beschreibt. Diese Quaternion lässt sich in die Eulerschen Winkel um die Achsen x, y und z (siehe Abbildung 4.3) umrechnen.

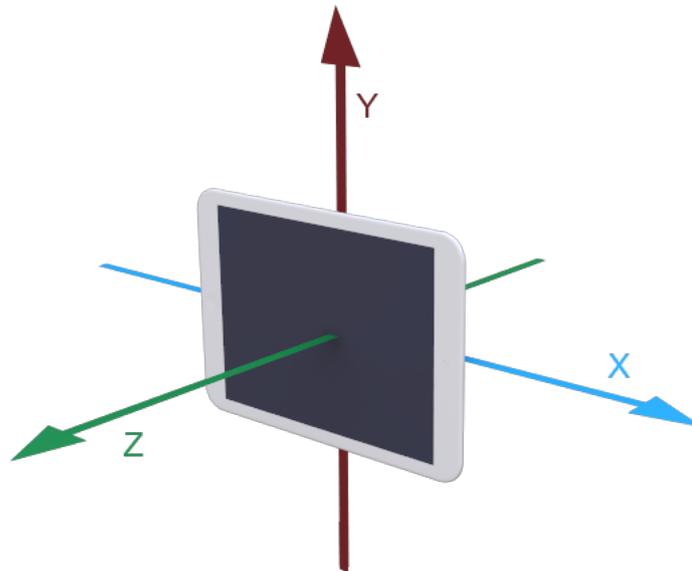


Abbildung 4.3.: Eulersche Winkel

Eine Drehung um die y-Achse ist notwendig bei einer freien Bewegung im Raum. Auch eine Drehung um die x-Achse kann sinnvoll sein, wenn das zu filmende Objekt sich nicht auf Höhe der Kamera befindet. Eine Drehung um die z-Achse liefert jedoch

4. Prototypische Umsetzung

lediglich ein schräges Bild. Dies soll laut Guidelines vermieden werden. Es wird also nur dieser Winkel weiter betrachtet.

Beim Gyrometer wird die Geschwindigkeit in die drei Richtungen der Achsen aus Abbildung 4.3 zurückgegeben. Das Gyrometer wird dazu genutzt, um ein Wackeln des Gerätes zu erkennen. Bei einem Wackeln ändert sich die Richtung, in die sich das Gerät bewegt. Es kann auch im Normalfall nötig sein beim Filmen die Richtung zu wechseln. Bei mehreren aufeinander folgenden, schlagartigen Richtungswechseln ist aber von einem Wackeln auszugehen.

Grenzwerte für Umgebungstests

Zum Überprüfen der Umgebung mussten für die verschiedenen Sensoren passende Grenzwerte gefunden werden, welche nicht überschritten werden sollten. Zum Finden dieser Grenzwerte wurde an drei verschiedenen Orten eine Minute lang gefilmt, wobei die Messungen der Sensoren geloggt wurden. Alle Werte wurden an einem sonnigen Tag gemessen. Bei den drei Orten handelt es sich um eine Wohnung, einen ruhigen, leicht schattigen Garten und eine stark befahrene Straße. Die Ergebnisse der Messungen sind in Tabelle 4.2 zu sehen.

	Wohnung	Garten	Straße
Min. Helligkeit	47 Lux	784 Lux	2612 Lux
Max. Helligkeit	3038 Lux	6260 Lux	107537 Lux
Max. Lautstärke	47.5 dB	52.6 dB	87 dB
Max. Winkel	4.7°	6.59°	6.6°

Tabelle 4.2.: Testwerte

Laut National Optical Astronomy Observatory (NOAO) ist außerdem der Empfohlene Luxwert für Warenhäuser, Wohnungen, Theater und Archive 150 Lux [27]. Dieser Wert wurde als untere Grenze gewählt, da davon auszugehen ist, dass es zum Filmen nicht dunkler sein sollte, als an den genannten Orten. Da bei der Messung auf der Straße ein Lux-Wert von über 100.000 erreicht wurde, und dies nicht zu hell erschien, wurde keine obere Grenze gewählt.

Für die Lautstärke erscheint es sinnvoll einen Wert zwischen Garten und Straße zu wählen, da der Garten ruhig war und die Straße eindeutig zu laut. Laut Industrial Noise Control (IAC) entsprechen 60 Dezibel zum Beispiel der Lautstärke einer Konversation in einem Restaurant, Hintergrundmusik oder einer Klimaanlage auf 30 Meter Entfernung [26].

Die Perspektive erschien in keinem der drei Videos zu schräg. Die in Tabelle 4.2 eingetragenen Maximalwerte traten jedoch auch jeweils nur kurz bei Bewegung auf. Als Grenzwert wurde mit 7° ein Wert, welcher leicht über den Messungen liegt, gewählt.

Beim Gyrometer ist davon auszugehen, dass bei zwei aufeinander folgenden Messungen im Normalfall nicht mehr als ein Richtungswechsel auf einer beliebigen Achse

4. Prototypische Umsetzung

vorkommen sollte. Um False Positives zu vermeiden, wurden aber als Grenzwert für ein Wackeln drei aufeinander folgende Richtungswechsel festgelegt.

Darstellung

Es sind drei Konzepte implementiert, an denen der Nutzer die genannten Faktoren überprüfen kann. Die erste Möglichkeit ist der “Check Environment”-Button. Bei Betätigung dieses Buttons öffnet sich ein Dialog und es werden die verschiedenen Aspekte überprüft (siehe Abbildung 4.4).

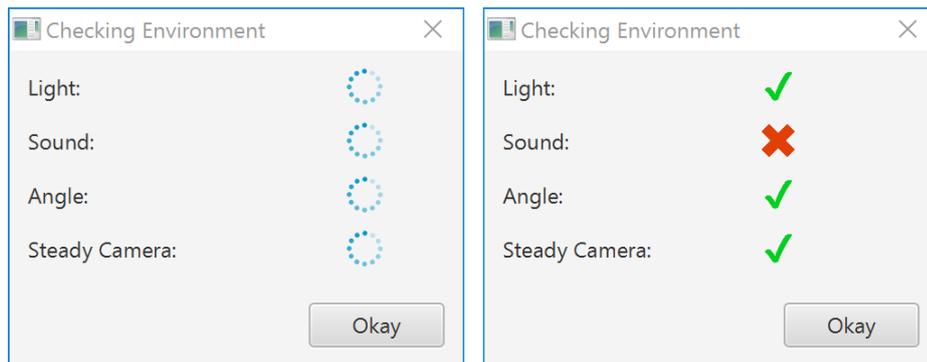


Abbildung 4.4.: Umgebungsüberprüfung vor Aufnahme

Der Nutzer kann so vor der Aufnahme überprüfen, ob die Umgebung und Ausrichtung des Gerätes in Ordnung sind. Es wird für 5 Sekunden überprüft, ob im Durchschnitt die Grenzwerte nicht überschritten werden. Anschließend erhält der Nutzer ein Feedback zu den vier Aspekten und kann im Zweifel entscheiden, ob er etwas an diesen verbessern kann.

Des Weiteren werden die Faktoren auch vor und während der Aufnahme live überprüft. Der Nutzer kann rechts unter dem Livestream Lautstärkelevel, Winkel und Helligkeit selber ablesen (siehe Abbildung 4.5). Bei Überschreiten eines Grenzwertes färben sich die Anzeigen rot, um dem auch während der Aufnahme dem Nutzer ein Feedback zu geben.



Abbildung 4.5.: Live Umgebungsüberprüfung

4. Prototypische Umsetzung

Die dritte Variante ist, dass das System überprüft, ob bei der Aufnahme ein Verstoß vorgekommen ist. Es wird überprüft, ob entweder ein Wackeln vorkommt, oder mindestens drei Sekunden am Stück einer der anderen Grenzwerte überschritten wird. Diese drei Sekunden wurden gewählt, damit nicht sofort ein Warnung angezeigt wird, auch wenn beispielsweise die Beleuchtung nur für eine Millisekunde schlecht war. Außerdem ist es besonders für die Umgebungsgeräusche relevant, da festgestellt werden soll, ob ein konstantes Hintergrundgeräusch vorhanden ist und dies möglichst nicht durch Geräusche, welche im Bezug auf die Aufnahme entstehen, ausgelöst werden sollte.

Wenn nun festgestellt wird, dass es während der Aufnahme zu dunkel, laut, die Perspektive zu schräg war oder ein Zittern vorhanden war, so öffnet sich bei Stoppen der Aufnahme ein Dialog (siehe Abbildung 4.6). Hier wird der Nutzer nun auf sämtliche Verstöße hingewiesen und kann dann für sich entscheiden, ob er das Video erneut aufnimmt. Zusätzlich wird eine Textdatei erzeugt, in der die Verstöße zusammen mit dem Zeitpunkt im Video aufgelistet werden, falls dem Nutzer unklar ist, an welcher Stelle diese aufgetreten sind. So soll der Nutzer während der Aufnahme nicht gestört oder eingeschränkt, aber dennoch unterstützt werden.

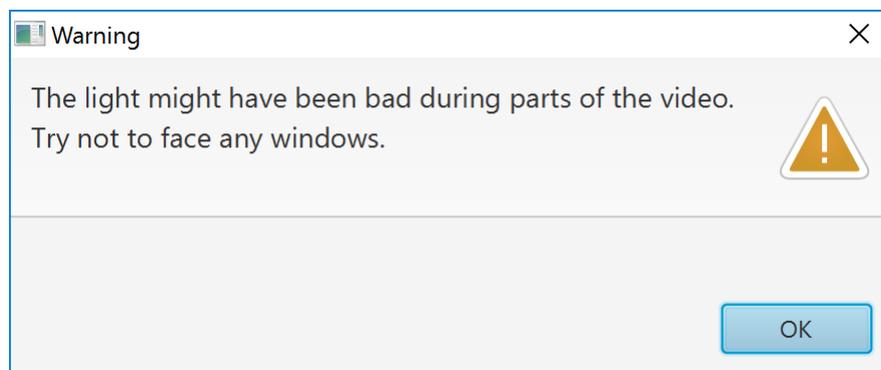


Abbildung 4.6.: Umgebungsüberprüfung bei Aufnahme

4.3.4. Dauer und Pausieren der Aufnahme

Die Aufnahme lässt sich durch den Pause-Button pausieren. Dieser Button ist erst nach 5 Sekunden Aufnahme aktiv, um zu schnelle Schnitte zu verhindern. Damit erfüllt er die Anforderung [R202].

Laut Guidelines zur Videoproduktion sollten Videos nicht zu lang sein. Für die Dauer der Aufnahme ([R211]) wurde die Guideline nach Guo et al. [18] gewählt, wonach das Video kürzer als 6 Minuten sein sollte. Nach 6 Minuten Aufnahme wird der Nutzer darauf hingewiesen, dass das Video möglicherweise zu lang wird (siehe Abbildung 4.7). Dazu wird die Dauer der bereits im Szenario vorhandenen Schritte zu der aktuellen Aufnahmezeit addiert. Es wird die Aufnahme aber nicht gestoppt. Es ist also möglich längere Videos aufzunehmen. Der Nutzer soll lediglich auf die

4. Prototypische Umsetzung

Guideline hingewiesen werden, ohne ihn zu stören oder zu stark einzuschränken. Außerdem kann der Nutzer in der Liste der Schritte die Dauer jedes einzelnen Schrittes sehen. Dadurch hat er einen Überblick wie lang das Gesamtvideo wird beziehungsweise welcher Schritt bei einer Überschreitung das Problem verursacht.

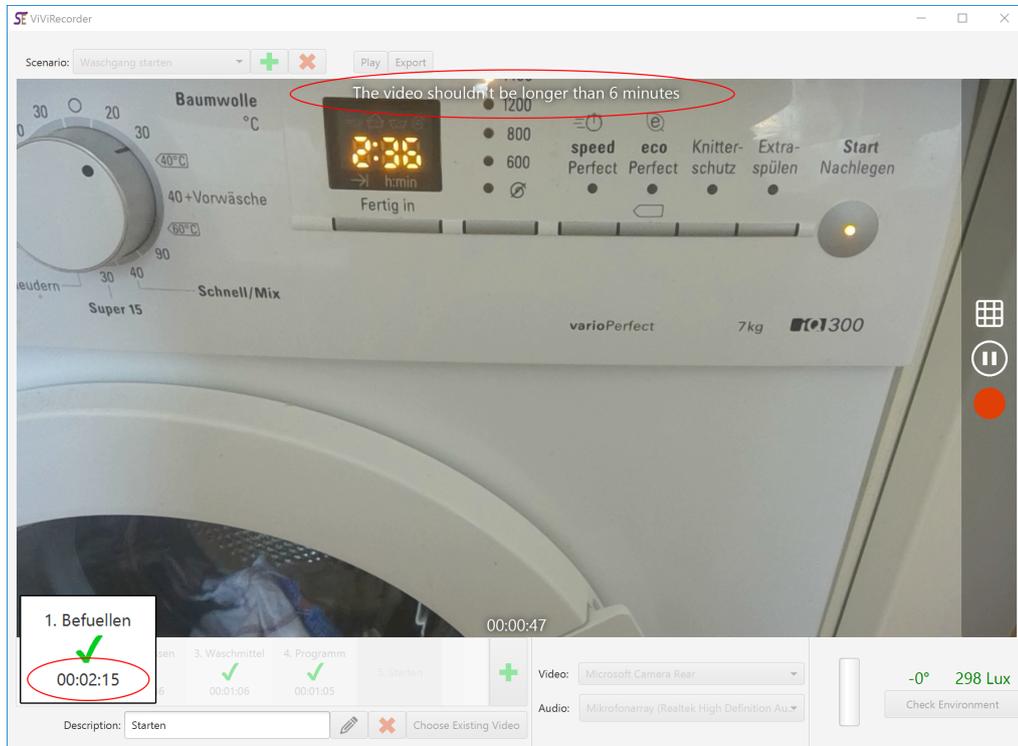


Abbildung 4.7.: Benachrichtigung bei Überschreitung von 6 Minuten

4.4. Einschränkungen

Wie bereits erwähnt wurde mit Hilfe von jni4net eine Schnittstelle erstellt, um auf die Sensorik des Gerätes zuzugreifen. Die erstellte Dynamic Link Library (DLL) funktioniert nur unter Windows. Damit die Sensoren auch unter anderen Betriebssystemen ausgelesen werden können, müssten noch zusätzliche Bibliotheken erstellt werden.

Die Helligkeitsüberprüfung ist auf die Rückkamera des Gerätes ausgelegt. Der Lichtsensor befindet sich auf der gleichen Seite wie die Frontkamera. Dies bedeutet, dass bei Benutzung der Frontkamera der Luxwert besonders hoch ist, wenn Licht in die Kamera scheint. Das Licht sollte aber auf das zu filmende Objekt, also in die entgegengesetzte Richtung scheinen. Eine Ausrichtung zu einem Fenster würde beispielsweise einen besonders guten Helligkeitswert liefern, was dem Gegenteil der Guideline entspricht.

5. Evaluation

Im Rahmen dieser Arbeit wurde der ViViRecorder - ein Prototyp eines Videorecorders - erstellt, der den Nutzer dabei unterstützen soll Guidelines zur Videoproduktion einzuhalten. Diesen Prototypen gilt es in diesem Kapitel zu evaluieren. Dazu soll die Evaluation zunächst mit Hilfe des GQM-Ansatzes (Goal-Question-Metric) [28] geplant werden. Anschließend wird die Evaluation durchgeführt und die Ergebnisse ausgewertet.

5.1. Vorbereitung mit GQM

Zunächst wurde ein Zielbaum aufgestellt, um das Ziel - eine vermehrte Nutzung von Vision Videos im Requirements Engineering - zu konkretisieren (siehe Abbildung 5.1).

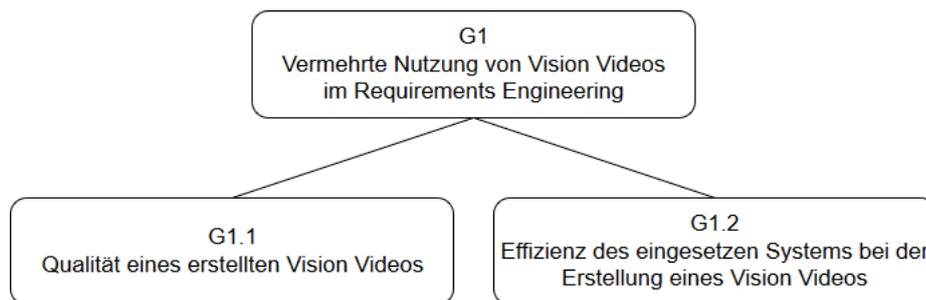


Abbildung 5.1.: Zielbaum

Es wurden die beiden Unterziele “Qualität eines erstellten Videos” und “Effizienz des eingesetzten Systems bei der Erstellung eines Vision Videos” aufgestellt. Beide Unterziele gilt es zu steigern, um zu dem Hauptziel zu gelangen. Nach Konkretisierung des Hauptzieles wurden Abstraction Sheets zu den Unterzielen erstellt, um die zu messenden abhängigen Variablen zu verdeutlichen. In Tabelle 5.1 ist beispielhaft das Abstraction Sheet zu Ziel G1.2 zu sehen. Das Abstraction Sheet zu G1.1 ist in Anhang B zu finden.

5. Evaluation

Zweck:	Qualitätsaspekt:	Betrachtungsgegenstand:	Perspektive:
Steigere	Effizienz	Videoproduktion	Requirements Engineer
Qualitätsfaktoren:		Einflussfaktoren:	
Zeitaufwand der Produktion eines Vision Videos		Hinweise zu Guidelines und Hilfestellung zur Strukturierung bei der Aufnahme eines Videos	
Ausgangshypothese:		Einflusshypothese:	
Aufgrund fehlender Erfahrung in der Videoproduktion ist der Workflow schlecht was zu einer niedrigen Effizienz bei der Erstellung eines Vision Videos führt.		Durch toolgestützte Unterstützung in Form von Hinweisen zu Guidelines und Strukturierung des Videos kann die Effizienz bei der Videoproduktion erhöht werden.	

Tabelle 5.1.: Abstraction Sheet: Effizienz bei der Videoproduktion

5.2. Planung

Zunächst müssen die unabhängigen Variablen für die Evaluation festgelegt werden. Dabei handelt es sich in diesem Fall um die verwendete Technik. Anschließend werden die Forschungsfragen aufgestellt und die abhängigen Variablen daraus abgeleitet. Zuletzt wird das Design und die genaue Vorgehensweise bei der Evaluation erläutert.

Techniken

1. ViViRecorder: Filmen eines Szenarios mit Hilfe des in dieser Arbeit entwickelten Prototypen.
2. Handykamera: Filmen eines Szenarios mit Hilfe einer üblichen Handykamera.

Forschungsfragen

1. Ist der Zeitaufwand für die Aufnahme eines Videos geringer bei Benutzung des ViViRecorders als bei Benutzung einer Handykamera?
2. Ist der geschätzte Zeitaufwand für die Nachbearbeitung eines Videos mit Hilfe des ViViRecorders geringer als mit Hilfe einer Handykamera?
3. Ist der subjektiv empfundene Aufwand bei der Produktion eines Videos geringer bei Benutzung des ViViRecorders als bei Benutzung einer Handykamera?
4. Ist die subjektiv empfundene Qualität eines erzeugten Videos mit Hilfe des ViViRecorders höher als mit Hilfe einer Handykamera?

Abhängige Variablen

1. Die Dauer für die Aufnahme der Videos.
2. Die geschätzte Zeit für die restliche Produktionsdauer des Videos nach Aufnahme.
3. Der subjektiv empfundene Aufwand für die Erstellung von Videos.
4. Die subjektiv empfundene Qualität der aufgenommenen Videos.

Zu testende Hypothesen

Die folgenden Nullhypothesen wurden aufgestellt, um die Forschungsfrage bezüglich der subjektiven Qualität der Videos zu beantworten:

H_{0_1} : Die Dauer für die Aufnahme der Videos bei Nutzung des ViViRecorders ist größer oder gleich als die Dauer für die Aufnahme der Videos bei Nutzung des Handys.

H_{0_2} : Die geschätzte Zeit für die restliche Produktionsdauer des Videos nach Aufnahme ist größer oder gleich bei Nutzung des ViViRecorders als bei Nutzung des Handys.

H_{0_3} : Der subjektiv Empfundene Aufwand für die Erstellung von Videos ist größer oder gleich bei Nutzung des ViViRecorders als bei Nutzung des Handys.

H_{0_4} : Die subjektive Videoqualität bei der Nutzung des ViViRecorders ist kleiner oder gleich als die subjektive Videoqualität bei der Nutzung des Handys.

Den Nullhypothesen entgegen stehen die Alternativhypothesen:

H_{1_1} : Die Dauer für die Aufnahme der Videos ist kleiner bei Nutzung des ViViRecorders als bei Nutzung des Handys.

H_{1_2} : Die geschätzte Zeit für die restliche Produktionsdauer des Videos nach Aufnahme ist kleiner bei Nutzung des ViViRecorders als bei Nutzung des Handys.

H_{1_3} : Der subjektiv Empfundene Aufwand für die Erstellung von Videos ist kleiner bei Nutzung des ViViRecorders als bei Nutzung des Handys.

H_{1_4} : Die subjektive Videoqualität bei Nutzung des ViViRecorders ist größer als die subjektive Videoqualität bei Nutzung des Handys.

Design

Die Evaluation folgt dem *Within-Subjects Design*, bei dem jeder Proband beide Techniken verwendet. Nachteil hierbei kann ein möglicher auftretender Lerneffekt sein, aber die Hoffnung ist einen besseren Vergleich zwischen den beiden Techniken zu erhalten, da die Evaluation in einem kleineren Rahmen ablaufen wird. Der Lerneffekt wird versucht durch ein *Cross-Design* zu verhindern, bei dem die Reihenfolge der genutzten Techniken variiert.

Die Probanden sollen ein einfaches Szenario mit beiden Techniken filmen und anschließend einen Fragebogen ausfüllen, um mit den gesammelten Daten die Forschungsfragen zu beantworten. Dabei stelle ich mich als Schauspieler zur Verfügung, da sonst für jeden Durchlauf noch ein weiterer Proband nötig wäre und der Schauspieler keinen größeren Einfluss auf die abhängigen Variablen haben sollte. Da Grundlage dieser Arbeit die Annahme ist, dass Software Engineers aus mangelnder Erfahrung mit Videoproduktion Vision Videos ablehnen, ist vor allem wichtig, dass die Probanden keine größere Erfahrung mit Videoproduktion haben. Aus Mangel an Zeit und Teilnehmern wird deswegen darauf verzichtet die Evaluation auf Software Engineers einzuschränken. Außerdem wird aus Zeitgründen kein vollständiges Video produziert. Es wird nur der Prozess der Aufnahme betrachtet und anschließend abgefragt, ob der Proband das Video noch schneiden oder nachbearbeiten würde und wenn ja, wie viel Zeit er schätzungsweise dafür benötigen würde.

5.3. Durchführung

5.3.1. Population

Aus zeitlichen Gründen haben an der Evaluation nur 6 Probanden aus unterschiedlichen Fachgebieten teilgenommen. Bei den Fachgebieten handelte es sich um Informatik, Maschinenbau, Medizin, Politikwissenschaft und Musikpädagogik. Zwei Probanden gaben an bereits Erfahrung mit Videoproduktion zu haben. In einem Fall wurde diese im Rahmen von Online-Streaming gesammelt, im anderen Fall im Rahmen von Schulprojekten. Somit hatten die Teilnehmer überwiegend keinen Bezug zum Requirements Engineering, aber auch keine professionelle Erfahrung mit der Videoproduktion, wodurch trotzdem getestet werden konnte, ob das erstellte Tool in diesem Fall eine Hilfestellung bieten kann. In Tabelle 5.2 sind noch einmal die Daten zu den einzelnen Probanden zu sehen.

5.3.2. Ablauf einer Sitzung

Als Szenario welches die Probanden filmen sollten wurde “Mensakarte aufladen und Essen kaufen” gewählt. Bei der Hälfte der Probanden wurde dafür eine reale Auf-

5. Evaluation

Proband	Fachgebiet	Erfahrung	Reihenfolge der Techniken
P1	Musikpädagogik	-	Handy, ViViRecorder
P2	Maschinenbau	-	ViViRecorder, Handy
P3	Maschinenbau	-	Handy, ViViRecorder
P4	Medizin	Schulprojekte	ViViRecorder, Handy
P5	Politikwissenschaften	Online-Streaming	Handy, ViViRecorder
P6	Informatik	-	ViViRecorder, Handy

Tabelle 5.2.: Population

ladestation genutzt, bei der anderen Hälfte ein Papier-Prototyp, da es nicht immer möglich war in der Mensa zu filmen. Das zu kaufende Essen und die Kasse mit Kartenscanner zum Bezahlen wurden in allen Fällen fiktiv dargestellt.

Den Probanden wurde erklärt, dass sie ein Video erstellen sollen, welches möglichst gut den Ablauf dieses Szenarios und die einzelnen Schritte die damit verbunden sind darstellen soll. Bei jedem Probanden gab es zwei Durchläufe. Es wurde abwechselnd zuerst mit der Handykamera (Nexus 4) und als zweites mit dem ViViRecorder gefilmt und andersherum. Vor Aufnahme mit dem ViViRecorder wurden noch kurz (etwa 1-2 Minuten) die Funktionen erklärt. Dann wurde bei beiden Durchgängen die Zeit gestoppt, bis die Probanden meinten fertig zu sein. Zum Schluss füllten die Probanden dann einen Fragebogen aus.

5.4. Auswertung

5.4.1. Ergebnisse der Evaluation

Zeitaufwand

Die Bearbeitungszeit (Task Completion Time) betrug im Durchschnitt mit dem Handy 5:00 Minuten und mit dem ViViRecorder 6:10 Minuten. Der Median ist in beiden Fällen etwas niedriger. Beim Handy beträgt er 4:30 Minuten und beim ViViRecorder 6:00 Minuten. Dies lässt sich durch einen Ausreißer unter den Probanden erklären, bei welchem beide Durchläufe etwas länger dauerten.

	Mittelwert	Standardabw.	Median
Alle	5:35 min	1:39 min	5:30 min
Handy	5:00 min	1:09 min	4:30 min
ViViRecorder	6:10 min	2:02 min	6:00 min

Nach dem Shapiro-Wilk-Test handelt es sich bei den Daten zu beiden Techniken um eine Normalverteilung. Damit ergibt sich der t-Test als Testmöglichkeit für statistische Signifikanz. Ein einseitiger t-Test mit einem Signifikanzlevel von 0.05 ergibt,

5. Evaluation

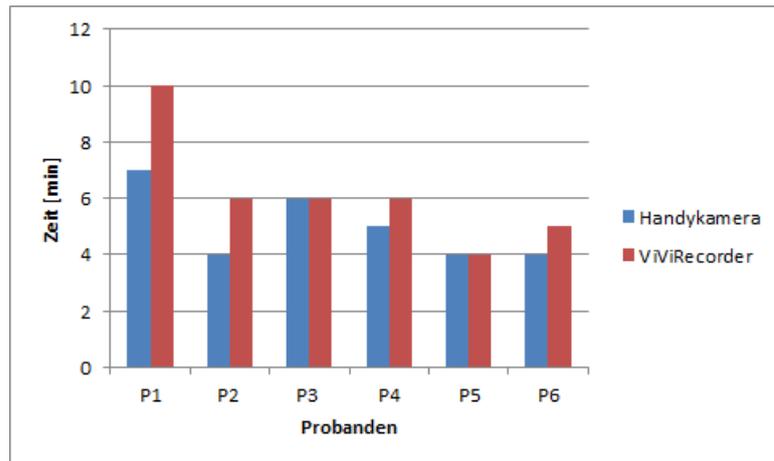


Abbildung 5.2.: Task Completion Time

dass die Bearbeitungsdauer mit dem ViViRecorder signifikant länger war, als die Bearbeitungsdauer mit dem Handy ($t = -2.444506$, $p = 0.029165$). Damit kann die Nullhypothese H_{0_1} nicht widerlegt werden.

Die Probanden wurden außerdem gefragt wie lang sie die Nachbearbeitungszeit für die Videos mit dem beiden verschiedenen Techniken einschätzen.

	Mittelwert	Standardabw.	Median
Alle	20:45 min	26:15 min	10:00 min
Handykamera	35:00 min	29:52 min	20:00 min
ViViRecorder	6:30 min	8:56 min	2:00 min

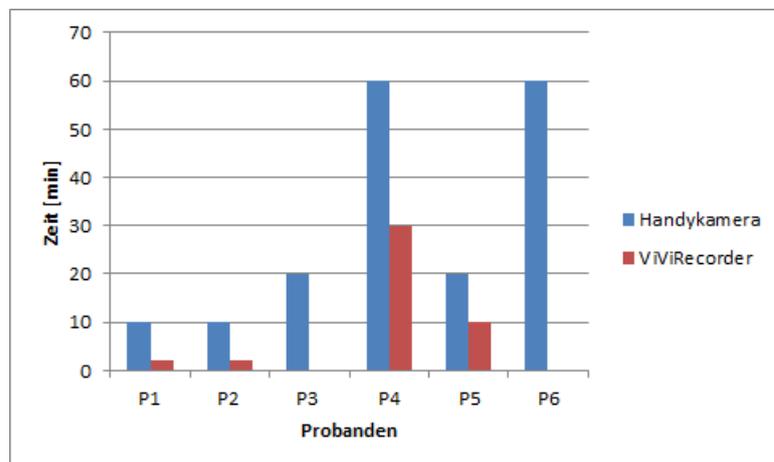


Abbildung 5.3.: Nachbearbeitungsdauer (geschätzt)

Ein Teilnehmer gab bei beiden Techniken ein Zeitintervall an. Für den ViViRecorder wurde die obere grenze dieses Intervalls gewählt und für das Handy die untere, um

weniger Spielraum für Ungenauigkeiten zu Gunsten des ViViRecorders zu lassen. Dennoch liegen sowohl Mittelwert als auch Median für den ViViRecorder deutlich unter den Werten für das Handy. Die Probanden gaben zusätzlich alle an, dass sie den Gesamtaufwand für die Produktion des Videos mit dem ViViRecorder geringer einschätzen. Die Daten wurden an dieser Stelle nicht auf statistische Signifikanz getestet, da es sich um Schätzwerte handelt. Es ist aber eine deutliche Tendenz zu den Alternativhypothesen H_{1_2} und H_{1_3} erkennbar.

Videoqualität

Die Probanden wurden nach einer subjektiven Beurteilung der Qualität der verschiedenen aufgenommenen Videos gefragt. Die Qualität wurde auf einer Skala von 1 bis 5 angegeben mit 5 = sehr gut und 1 = sehr schlecht. Der Median für die subjektive Qualität der mit dem ViViRecorder erstellten Videos beträgt 4 und ist damit eine Stufe über dem Median der Qualität der mit dem Handy erstellten Videos. Der Mann-Whitney-Test mit einem Konfidenzintervall von 95% ergibt, dass die subjektive Qualität mit ViViRecorder signifikant größer ist ($W = 30$, $p = 0.013443$). Es kann also die Alternativhypothese H_{1_4} "Die subjektive Videoqualität bei Nutzung des ViViRecorders ist größer als die subjektive Videoqualität bei Nutzung des Handys." angenommen werden.

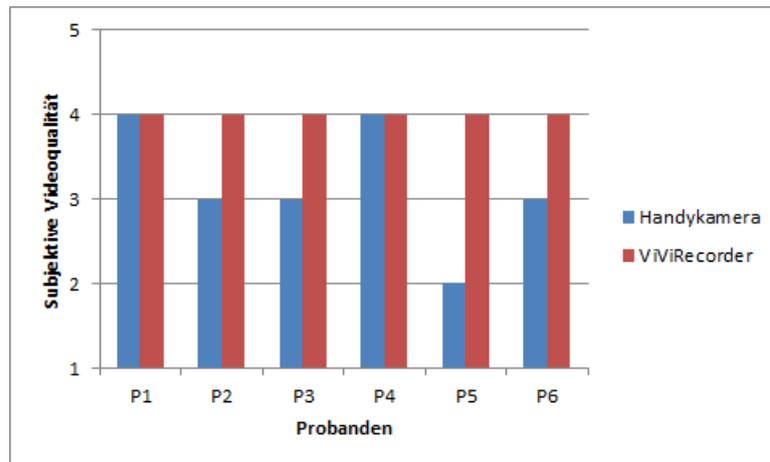


Abbildung 5.4.: Subjektive Videoqualität

5.4.2. Bewertung der Ergebnisse

Zunächst sollen die aufgestellten Forschungsfragen mit Hilfe der Ergebnisse beantwortet werden. Anschließend soll eine Schlussfolgerung daraus gezogen werden.

Beantwortung der Forschungsfragen

1. Ist der Zeitaufwand für die Aufnahme eines Videos geringer bei Benutzung des ViViRecorders als bei Benutzung einer Handykamera?

Die Bearbeitungszeit mit dem ViViRecorder war statistisch signifikant länger als die Zeit, welche mit dem Handy benötigt wurde. Diese Forschungsfrage lässt sich also verneinen. Es ist aber davon auszugehen, dass die Teilnehmer bereits zuvor eine Handykamera benutzt hatten, während der ViViRecorder zum ersten mal benutzt wurde. Etwas mehr Zeit wurde also möglicherweise benötigt, da die Teilnehmer noch nicht an den ViViRecorder gewöhnt waren. Außerdem kostet das Überprüfen der Umgebung vor dem Filmen Zeit. Bei Proband P1 wurden nach Hinweisen des ViViRecorders sogar Videos erneut aufgenommen.

2. Ist der geschätzte Zeitaufwand für die Nachbearbeitung eines Videos mit Hilfe des ViViRecorders geringer als mit Hilfe einer Handykamera?

Sämtliche Teilnehmer gaben eine geringere Zeit an, welche für die Nachbearbeitung mit dem ViViRecorder benötigt würde. Da es sich an dieser Stelle um Schätzwerte handelt, wurden diese nicht auf statistische Signifikanz untersucht. Der deutliche Unterschied von Median und Mittelwert lässt dennoch eine starke Tendenz zu Gunsten des ViViRecorders erkennen.

3. Ist der subjektiv empfundene Aufwand bei der Produktion eines Videos geringer bei Benutzung des ViViRecorders als bei Benutzung einer Handykamera?

Auf die Frage mit welcher Technik die Probanden den Gesamtaufwand für die Videoproduktion geringer einschätzen würden, gaben alle Probanden den ViViRecorder an. Dies bestätigt noch einmal den großen Unterschied in der Nachbearbeitung. Es ist anzunehmen, dass dies vor allem Resultat des automatischen Zusammenfügens der Schritte ist.

4. Ist die subjektiv empfundene Qualität eines erzeugten Videos mit Hilfe des ViViRecorders höher als mit Hilfe einer Handykamera?

Die Teilnehmer gaben eine statistisch signifikant höhere subjektive Qualität für die Videos an, welche sie mit dem ViViRecorder erstellten. Die Features des ViViRecorders scheinen also einen positiven Einfluss auf die Qualität der Videos genommen zu haben. Da es sich aber um ein subjektives Empfinden handelt, ist es schwer festzustellen welche Aspekte genau hierfür verantwortlich sind.

Schlussfolgerungen

Die Bearbeitungszeit war zwar bei Benutzung des ViViRecorders größer, der Gesamtaufwand für die Videoproduktion erschien den Teilnehmern aber dennoch geringer. Außerdem wurde die Bearbeitungszeit bei Benutzung des ViViRecorders durch Umgebungschecks und erneute Aufnahmen beeinflusst. Diese Faktoren erhöhen die Produktionsdauer, verfolgen aber das Ziel G1.1 eine höhere Videoqualität zu erzielen. Bei diesem zweiten Ziel war auch eine positive Tendenz zu erkennen, weshalb insgesamt ein positives Fazit aus der Evaluation gezogen werden kann.

5.4.3. Bedrohung der Validität

In diesem Abschnitt wird auf die Bedrohung der Validität eingegangen. Gemeint sind damit Einschränkungen der Gültigkeit der Ergebnisse.

Teilnehmerzahl

Durch die geringe Teilnehmerzahl ist die *Conclusion Validity* [29] bedroht. Um eine fundierte Schlussfolgerung zu ziehen wäre eine Studie in einem größeren Rahmen notwendig, da bei nur sechs Teilnehmern die Ergebnisse zu stark von Ausreißern beeinflusst werden. Mitunter wegen der geringen Teilnehmerzahl wurde das *Within-Subjects Design* gewählt, um dennoch einen direkten Vergleich der beiden Techniken zu erhalten.

Bezug zum Requirements Engineering

Die Teilnehmer hatten größtenteils keinen Bezug zum Requirements Engineering. Es wäre möglich, dass ein Requirements Engineer andere Anforderungen an ein Video stellt und deswegen zu einem anderen Ergebnis kommen würde. Es wurde versucht ein Szenario zu wählen, welches Bezug zu Software enthält und trotzdem für alle Teilnehmer leicht zu verstehen ist, um diese Bedrohung der *Construct Validity* [29] zu reduzieren.

Hardware

Die Evaluation bezieht sich auf die Software, die zur Videoaufnahme verwendet wird. Es bestehen jedoch auch Unterschiede in der verwendeten Hardware der beiden Techniken. So gaben zum Beispiel zwei Teilnehmer die Größe des Gerätes als einen Nachteil des ViViRecorders an.

5.4.4. Gewonnene Erkenntnisse aus dem praktischen Einsatz

Der meistgenannte Vorteil des ViViRecorders war das Einteilen des Videos in Schritte beziehungsweise das automatische Zusammenfügen dieser. Dies scheint also ein sinnvolles Feature zu sein, was sich auch an den Einschätzungen zu dem Produktionsaufwand des Videos erkennen lässt.

Ansonsten wurden drei der sechs Teilnehmer durch den ViViRecorder auf eine schlechte Beleuchtung hingewiesen. Zwei davon durch einen vorherigen Umgebungcheck und einer durch eine Benachrichtigung nach Aufnahme des Videos. Diese Hinweise erschienen auch sinnvoll. Die anderen Hinweise zur Umgebung traten nicht direkt in Erscheinung, aber störten demnach auch nicht. Sie könnten aber dennoch in Form der Live-Anzeigen relevant gewesen sein.

Eine weitere Beobachtung war, dass beim Filmen mit Schauspielern die Koordination, wann das Filmen startet und stoppt, eine Schwierigkeit darstellen kann. Dies verdeutlicht die Wichtigkeit von Anforderung [R212] einen Puffer am Anfang und Ende des Videos zu erstellen und besonders die Möglichkeit zu bieten Anfang und Ende in der Anwendung zuzuschneiden ([R213]). Eine weitere Idee, welche im Bezug zu diesem Problem aufkam, ist ein akustisches Signal bei Start und Stopp der Aufnahme.

6. Fazit und Ausblick

In diesem Kapitel soll ein abschließender Überblick über den Inhalt dieser Arbeit geschaffen werden. Anschließend soll noch ein Ausblick auf mögliche Erweiterungen des ViViRecorders erfolgen.

6.1. Fazit

Eine Schwierigkeit im Requirements Engineering stellt das Verständnis zwischen Kunde und Entwickler da. Dieses Problem beruht vor allem auf der Symmetry of Ignorance, eines beidseitigen fehlenden Verständnisses über das Fachgebiet des Anderen.

Aus vorherigen Arbeiten ging hervor, dass Videos ein geeigneter Kommunikationsweg sind, um Informationen verständlich zu vermitteln. Dies führte zu Vision Videos, einer Methode Anforderungen zur Verifizierung in Videoform darzustellen. Bis heute handelt es sich dabei jedoch nicht um eine etablierte Methode im Requirements Engineering. Eine mögliche Begründung dafür ist die fehlende Erfahrung der Softwareentwickler in der Videoproduktion, um effizient ein qualitativ hochwertiges Video zu erstellen.

In dieser Arbeit wurde sich deswegen mit Guidelines zur Videoproduktion auseinandergesetzt. Darauf basierend wurden Anforderungen für einen Videorecorder aufgestellt, welcher den Nutzer dabei unterstützt, die gesammelten Guidelines umzusetzen. Für die zusätzlichen notwendigen Anforderungen eines Videorecorders wurden häufige Features aus vorhandenen Kameraanwendungen für Android betrachtet.

Anschließend wurde anhand dieser Anforderungen ein Prototyp eines solchen Videorecorder - der ViViRecorder - erstellt und die umgesetzten Konzepte erläutert. Ein zentrales Feature des ViViRecorders ist die Umgebungsüberprüfung. Mit Hilfe der Sensorik der verwendeten Hardware überprüft der ViViRecorder die Umgebung auf Licht, Umgebungsgeräusche, Kameraausrichtung und ruhige Kamerahaltung. Der Nutzer wird darauf hingewiesen, wenn einer der empirisch gewonnen Grenzwerte bezüglich der genannten Aspekte überschritten wird. Außerdem bietet das Programm dem Nutzer die Möglichkeit ein Video in mehrere Schritte eines Szenarios zu unterteilen. Dies soll einen engeren Bezug zu den formalen Notationen des Requirements Engineerings schaffen. Zudem lässt sich die Reihenfolge der Schritte bearbeiten und einzelne Schritte austauschen. Dadurch lässt sich das Video leicht anpassen, wenn sich das zugehörige Szenario verändert.

Der ViViRecorder wurde anschließend in kleinem Rahmen evaluiert und mit einer

üblichen Handykamera verglichen. Die Probanden filmten mit beiden Techniken ein simples Szenario. Dabei wurden die *Task Completion Time* gemessen und weitere subjektive Daten über einen Fragebogen gesammelt. Ergebnis der Evaluation waren positive Tendenzen zu Gunsten des ViViRecorders, was sowohl Produktionsaufwand als auch Qualität der Videos angeht.

6.2. Ausblick

Hier wird auf einige Punkte zur Erweiterung beziehungsweise Verbesserung des ViViRecorders eingegangen.

Zunächst einmal wurden nicht alle aufgestellten Anforderungen implementiert. Diese nicht implementierten Anforderungen können als Erweiterung angesehen werden. Dabei stach beim praktischen Einsatz im Rahmen der Evaluation besonders die Option Videos in der Anwendung zu schneiden hervor.

Zusätzlich zu einer Einstellungsmöglichkeit für Zoom wären weitere Einstellungen um Features ein- und auszuschalten sinnvoll. Eine Idee dazu wäre bei Benachrichtigung zu einer nicht eingehaltenen Guideline eine “nicht erneut anzeigen” Option hinzuzufügen, damit der Nutzer nicht bei jedem Video den Hinweis erneut bekommt, wenn ihm dieser bereits bewusst ist.

Zudem wäre ein genaueres Feedback für den Nutzer wichtig, sowohl bei einem Umgebungsscheck vor dem Filmen, als auch bei den Benachrichtigungen danach. Die Zeitpunkte zu denen Verstöße gegen Guidelines registriert werden, werden zwar in einer Log-Datei gespeichert, es wäre aber sinnvoll diese Informationen für den Nutzer besser zugänglich zu machen, indem man die Zeitpunkte zum Beispiel im Videoplayer markiert.

Eine weitere Erweiterung könnte eine Möglichkeit sein, eine übergeordnete Struktur für Projekte zu schaffen, um bei mehreren Projekten nicht die Szenarien zu vermischen. Außerdem könnten Überlegungen dazu angestellt werden, die Struktur eines Szenarios weiter auszubauen. Es könnte Szenarien mit mehreren alternativen Abläufen geben, bei denen man sich dann im Videoplayer für einen bestimmten Pfad entscheiden kann. So könnte der Nutzer interaktiv die unterschiedlichen Abläufe eines Szenarios durchlaufen.

Es wäre zu dem sinnvoll die Grenzwerte für die Umgebungstests des ViViRecorders noch genauer zu evaluieren, da zu diesen noch wenige Daten vorliegen.

A. Anforderungsermittlung: Features aus Kameraanwendungen

↓ Feature	Kamera-App						
	#1	#2	#3	#4	#5	#6	#7
Videoaufnahme	x	x	x	x	x	x	x
Kameraauswahl	x	x	x	x	x	x	x
Zoom	x	x	x	x	x	x	x
Blitz	x	x	x	x	x	x	x
Tap to focus	x	x	x	x	x	x	
Auswahl der Auflösung	x	x	x	x	x		x
Auswahl von Seitenverhältnissen		x	x	x	x	x	
Raster	x	x	x	x	x		
Zeitraffer	x		x	x	x		x
Pausieren der Aufnahme		x	x	x	x		
Manuelle Belichtung	x		x		x	x	
Filter/Effekte				x	x	x	
Auswahl des Mikrofons		x	x	x			
Neigungsanzeige		x	x		x		
Weißabgleich					x	x	x
Einstellung einer Maximaldauer		x	x				
Zeitlupe	x						
Ohne Ton aufnehmen			x				
Geotagging			x				

- #1: OnePlus 3 Camera (Integrierte Kamera-App auf einem OnePlus 3)
- #2: HD Camera Pro (Google Play Store ID: photo.selfie.camera.hdcamera)
- #3: Open Camera (Google Play Store ID: net.sourceforge.opencamera)
- #4: Camera MX (Google Play Store ID: com.magix.camera_mx)
- #5: A Better Camera (Google Play Store ID: com.almalence.opencam)
- #6: HD Camera + (Google Play Store ID: com.toh.hdcamera)
- #7: HD Camera (Google Play Store ID: tools.photo.hd.camera)

B. Evaluation: Abstraction Sheet

Zweck:	Qualitätsaspekt:	Betrachtungsgegenstand:	Perspektive:
Steigere	Qualität	Vision Video	Requirements Engineer
Qualitätsfaktoren:		Einflussfaktoren:	
Anzahl der erfüllten Guidelines durch ein vom Requirements Engineer erstelltes Video		Hinweise zu Guidelines bei der Aufnahme eines Videos	
Ausgangshypothese:		Einflusshypothese:	
Aufgrund fehlender Erfahrung in der Videoproduktion ist die Qualität eines durch den Requirements Engineer erstellten Vision Videos gering.		Durch toolgestützte Unterstützung in Form von Hinweisen zu Guidelines kann die Qualität von Vision Videos erhöht werden.	

Tabelle B.1.: Abstraction Sheet: Qualität eines Vision Videos

C. Inhalte der CD

In diesem Anhang befindet sich eine Auflistung der Daten auf der beiliegenden CD.

1. Bachelorarbeit in digitaler Form (PDF-Datei, LaTeX-Projekt)
2. Evaluationsunterlagen bestehend aus:
 - a) Fragebogen
 - b) Ergebnisse der Evaluation als Excel Tabelle
 - c) Videos aus der Evaluation
3. Quellcode des ViViRecorders als Eclipse-Projekt

Literaturverzeichnis

- [1] S. W. Ambler,
Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process,
New York: John Wiley & Sons, Inc., 2002.
- [2] *29148-2011 - EISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes –Requirements engineering*,
IEEE, 2011.
- [3] S. Fricker, R. Grau und A. Zwingli,
Requirements engineering: best practice,
Springer, 2015.
- [4] G. Fischer,
Symmetry of ignorance, social creativity, and meta-design,
Elsevier, 2000.
- [5] O. Brill, K. Schneider und E. Knauss,
Videos vs. Use Cases: Can Videos Capture More Requirements under Time Pressure?,
Springer-Verlag Berlin Heidelberg, 2010.
- [6] A. Sutcliffe,
Scenario-based Requirements Engineering,
11th IEEE International Requirements Engineering Conference, 2003.
- [7] O. Creighton, M. Ott und B. Bruegge,
Software Cinema-Video-based Requirements Engineering,
14th IEEE International Requirements Engineering Conference, 2006.
- [8] O. Karras, J. Klünder und K. Schneider,
Enrichment of Requirements Specifications with Videos: Enhancing the Comprehensibility of Textual Requirements,
Zenodo, 2016.
- [9] O. Karras und K. Schneider,
Software Professionals are Not Directors: What Constitutes a Good Video?,
arXiv:1808.04986, 2018.

- [10] E. Sauerwein, F. Bailom, K. Matzler, H. H. Hinterhuber,
The Kano model: How to delight your customers,
International Working Seminar on Production Economics, 1996.
- [11] S. Bühne und A. Herrmann,
Handbuch Requirements Management nach IREB Standard,
IREB eV, 2015.
- [12] E. Börger, B. Hörger, D. L. Parnas und H. D. Rombach,
Requirements Capture, Documentation, and Validation,
Schloss Dagstuhl, Dagstuhl Report 242, 1999.
- [13] C. Rupp und die SOPHISTen,
Requirements-Engineering und -Management,
Carl Hanser Verlag , 2014.
- [14] K. Pohl,
Requirements engineering: fundamentals, principles, and techniques,
Springer Publishing Company, Incorporated, 2010.
- [15] S. A. Fricker, K. Schneider, F. Fotrousi, C. Thuemmler,
Workshop videos for requirements communication,
Springer, 2016.
- [16] O. Karras, S. Kiesling und K. Schneider,
Supporting requirements elicitation by tool-supported video analysis,
24th IEEE International Requirements Engineering Conference, 2016.
- [17] O. Karras, C. Unger-Windeler, L. Glauer und K. Schneider,
*Video as a By-Product of Digital Prototyping: Capturing the Dynamic
Aspect of Interaction*,
IEEE 25th International Requirements Engineering Conference Work-
shops, 2017.
- [18] P. J. Guo, J. Kim und R. Rubin,
*How Video Production Affects Student Engagement: An Empirical Stu-
dy of MOOC Videos*,
ACM, 2014.
- [19] E. Lackner, M. Kopp und M. Ebner,
How to MOOC? – A pedagogical guideline for practitioners,
10th International Scientific Conference "eLearning and Software for
Education", Bucharest, 2014.
- [20] N. Georgitis, K. Peach, and S. Rodriguez,
ARSC Video Production Guidelines,
[Online]. Available:
[http://www.arsc-audio.org/pdf/video_production_guidelines.
pdf](http://www.arsc-audio.org/pdf/video_production_guidelines.pdf)

- [21] Boise State University,
Video Guidelines and Quality Assurance,
[Online]. Available:
<http://comms.leeds.ac.uk/video/video-guidelines/>
- [22] University of Leeds,
Video Guidelines and Quality Assurance,
[Online]. Available:
<http://comms.leeds.ac.uk/video/video-guidelines/>
- [23] Massachusetts Institute of Technology,
Video Production Guide,
[Online]. Available:
<http://web.mit.edu/techtv/videoprodguide/videoprodguide.pdf>
- [24] G. Millerson, und J. Owens
Video Production Handbook,
Focal Press, 2008.
- [25] O. Karras,
Werkzeugunterstützte Analyse von Requirements-Workshop-Videos,
Masterarbeit, 2015.
- [26] IAC Acoustics,
Comparitive Examples of Noise Levels,
[Online]. Available:
<http://www.industrialnoisecontrol.com/comparative-noise-examples.htm>
- [27] National Optical Astronomy Observatory,
Recommended Light Levels (Illuminance) for Outdoor and Indoor Venues,
[Online]. Available:
https://www.noao.edu/education/QLTkit/ACTIVITY_Documents/Safety/LightLevels_outdoor+indoor.pdf
- [28] R. van Solingen, V. Basili, G. Caldiera und H. D. Rombach,
Goal Question Metric (GQM) Approach,
John Wiley & Sons, Inc., 2002.
- [29] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wessln,
Experimentation in Software Engineering,
Springer Science & Business Media, 2012.

Erklärung der Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit 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 28.09.2018

Janko Happe