
Software Requirements Specification

für

OBS-Plugin

Version 2.0

Prepared by Dominik Deller, Julia Grötsch, Khang Ho, Philipp Weber

Universität Regensburg, Lehrstuhl für Medieninformatik

07.10.2018

Inhalt

1. Einführung.....	1
1.1. Zweck.....	1
1.2. Leserzielgruppe & Aufbau.....	1
1.3. Produktumfang.....	1
1.4. Referenzen	1
2. Überblick	1
2.1. Produktumriss	1
2.2. Hauptfunktionen.....	2
2.3. Zielgruppen und Charakteristika.....	2
2.4. Betriebssystem	3
2.5. Vorgaben in Design und Implementierung	3
2.6. Benutzerdokumentation	3
2.7. Annahmen und Abhängigkeiten.....	3
3. Anforderungen an externe Schnittstellen	4
3.1. User Interfaces	4
3.1.1 Start.....	4
3.1.2 Startseite Testleiter	4
3.1.3 Startseite Proband	6
3.2. Hardware Interfaces	6
3.3. Software Interfaces	6
3.4. Kommunikations-Interfaces.....	7
4. Systemkomponenten	7
4.1. Streaming des Testpersonen-Screens.....	7
4.2. Screen Recording	7
4.3. Task modelling.....	8
4.4. Aufzeichnung von Tastaturaktivität.....	8
4.5. Aufnahme der Mimik der Testperson	9
4.6. Zeitsensible Anmerkungen.....	9
4.7. Kommunikation mit der Testperson.....	9
5. Other Nonfunctional Requirements.....	10
1.1 Performance Requirements	10
1.2 Safety Requirements	10
1.3 Security Requirements	11
1.4 Software Quality Attributes	11
1.5 Business Rules	11
1.6 Other Requirements	11

Revision History

Version	Datum	Geändert	Durch
1.0	11.06.2018	Initialdokument	Alle
2.0	01.10.2018	Finales Dokument	Alle

1. Einführung

1.1. Zweck

Mit der vorliegenden Software soll es möglich sein, unter Einbindung der Open Broadcaster Software¹ (im Folgenden OBS genannt) einen (Remote-)Usability-Test durchzuführen, ohne auf kostenintensive Spezial-Software angewiesen zu sein. Die Funktionen sollen sich an bereits bestehenden Usability-Testing-Software-Produkten (wie z.B. Morae²) orientieren.

Ziel des Projekts ist die Fertigstellung von einer ersten Version der Software, welche die im Folgendem näher beschriebenen Anforderungen erfüllen soll.

1.2. Leserzielgruppe & Aufbau

Das Dokument richtet sich an Personen, die die Implementierung der Software nachvollziehen wollen, an Entwickler, welche ein Interesse an der Vorgehensweise bei der Entwicklung und Einbindung einer Usability-Testing-Umgebung in OBS haben, sowie an Benutzer, welche die Software für eigene Usability-Tests verwenden wollen.

Im Dokument werden zunächst die Hauptfunktionen und Zielgruppe des Projekts vorgestellt, sowie grundsätzliche Vorgaben für die Entwicklung erläutert. Danach werden Schnittstellen vorgestellt, welche die Software integriert. Die Hauptfunktionen der Software werden im weiteren Verlauf erläutert, sowie ein Überblick über die nicht-funktionalen Anforderungen gegeben.

1.3. Produktumfang

Die entwickelte Software soll es dem Benutzer ermöglichen, durch OBS (Remote-)Usability-Tests durchzuführen. Da sich die Funktionen an bereits bestehenden Usability-Testing-Software-Produkten orientieren werden, soll es durch die vorliegende Lösung möglich sein, solche Tests professionell, aber ohne finanziellen Aufwand durchzuführen.

1.4. Referenzen

- OBS Plugin API Documentation. (2018). Retrieved from <https://github.com/jp9000/OBS/wiki/OBS-Plugin-API-Documentation>
- Morae. (2018). Retrieved from <https://www.techsmith.de/morae.html>
- Open Broadcaster Software | Home. (2018). Retrieved from <https://obsproject.com/>
- Bootstrap. (2018). Retrieved from <http://getbootstrap.com/>

2. Überblick

2.1. Produktumriss

Die vorliegende Software wurde entwickelt, um unter Zuhilfenahme der Streaming-Software OBS, Usability-Tests durchzuführen. Der Hauptfokus liegt hierbei auf der Möglichkeit, die Tests nicht nur

¹ <https://obsproject.com/de>

² <https://www.techsmith.de/morae.html>

vor Ort sondern auch per *remote* abzuhalten. Es handelt sich um ein Open Source Project und wurde für das Betriebssystem Windows entwickelt.

2.2. Hauptfunktionen

Bei den Hauptfunktionen muss zwischen Testleiter- und Testperson unterschieden werden, da je nach gewählter Rolle verschiedene Funktionalitäten zur Verfügung stehen. (Ist nichts Näheres vermerkt, steht eine Funktionalität beiden Seiten zur Verfügung.)

Webpage:

- Auswahl der Benutzerrolle
 - Angabe eines Namens
 - Auswahl zwischen Testleiter- oder Testpersonen-Ansicht
 - Starten der Anwendung durch "Los geht's!".
- Tutorial
- Stream (Testleiter-Seite)
 - OBS-Stream auf der Webpage
- Testmanager (auf Testleiter-Seite)
 - Auswahl eines Tests aus der Datenbank
 - Modellierung eines neuen Tests
 - Titel des Tests angeben
 - Task hinzufügen
 - Test speichern
- Tasks
 - Bereitstellen von Tasks durch den Testleiter (auf Testleiter-Seite)
 - Weitschaltung der einzelnen Tasks
 - Anzeigen des aktuellen Task (auf Testperson-Seite)
 - Aufzeichnen der Task-Completion-Time (auf Testperson-Seite)
 - Starten eines Task
 - Beenden eines Task
- Annotationen (auf Testleiter-Seite)
 - Möglichkeit zeitsensitive Ereignisse während und nach dem Test zu erfassen
 - Speicherung der Annotation
- Chat
 - Eingabe der Nachricht
 - Übermittlung in Echtzeit an das Gegenüber

OBS: (auf Testperson-Seite)

- Aufzeichnung des Bildschirms der Testperson
- Aufzeichnung von Tastatur Events durch Einbinden des Python-Skripts
- Aufzeichnung der Mimik des Probanden
- Audioaufzeichnung

2.3. Zielgruppen und Charakteristika

Typische Anwender, sind:

- Studenten oder Dozenten der Informatik, welche sich mit Usability Test auseinandersetzen und diese auch durchführen.

- Fortgeschrittene oder professionelle Anwender, die dieses Plugin für tiefgehende Tests verwenden möchten.
- Softwareentwickler, die sich mit der Thematik befassen und dieses Projekt weiterentwickeln möchten.

2.4. Betriebssystem

- Windows 8
- Windows 10

2.5. Vorgaben in Design und Implementierung

Die Software wird einerseits aus einer Webpage bestehen, andererseits aus einem Python-Skript, welches in OBS eingebunden werden muss.

Für die Webpage wurde das Bootstrap-Framework benutzt. Dies soll eine plattformübergreifende Benutzbarkeit der Software ermöglichen.

Als Entwicklungsumgebung wurde für die Homepage Atom³ verwendet.

Für den Server werden die Webserver Applikationen NGINX⁴ und Apache2⁵ verwendet. Hier wird NGINX als RTMP-Server dienen und die Übertragung des Streams an den verwendeten JW-Player⁶ abwickeln. Der verwendete JW-Player basiert auf Flash-Technik der Firma Adobe Systems⁷.

Neben den oben genannten Punkten wird eine MySQL⁸ Datenbank zum Einsatz kommen, in welcher Informationen zu den genutzten Tasks hinterlegt werden. Über das Webinterface von phpMyAdmin⁹ soll diese Datenbank konfiguriert und verwaltet werden.

Sämtliche Kommunikation zwischen Frontend (Webpage) und Backend (Server) soll durch die Programmierungssprache PHP¹⁰ abgewickelt werden.

2.6. Benutzerdokumentation

Mit der Software wird eine Dokumentation zur Inbetriebnahme des Systems bereitgestellt. Es müssen einige Softwarepakete installiert werden, welche in der Dokumentation näher erläutert werden. Zudem findet der Nutzer auf der Startseite der Webpage ein Tutorial, welches die Hauptfunktionen näher erläutert.

2.7. Annahmen und Abhängigkeiten

Bei den Abhängigkeiten ist zu unterscheiden, in welcher Rolle die Software benutzt wird. Ist man der Testleiter, benötigt man lediglich einen Webbrowser (mit Internetzugriff, um die Webpage zu öffnen). Die Testperson benötigt eine Python Installation auf ihrem Rechner, da das Skript zur Einbindung in OBS mit Python entwickelt wurde. Außerdem muss die Open Broadcaster Software auf dem zu verwendenden Gerät vorhanden sein.

³ <https://atom.io/>

⁴ <https://www.nginx.com/>

⁵ <https://httpd.apache.org/>

⁶ <https://www.jwplayer.com/>

⁷ <https://www.adobe.com/de/>

⁸ <https://www.mysql.com/de/>

⁹ <https://www.phpmyadmin.net/>

¹⁰ <http://php.net/>

3. Anforderungen an externe Schnittstellen

3.1. User Interfaces

3.1.1 Start

Regensburger Usability Testing Plattform

Willkommen

Bitte geben Sie einen Namen an

☒ Ich bin der Testleiter
☐ Ich bin der Testteilnehmer

➔ Los gehts!

🎓 Tutorial

Auf der Startseite hat der Benutzer die Möglichkeit, einen Namen einzugeben, seine Rolle auszuwählen, sowie ein Tutorial aufzurufen. Der Name soll angegeben werden, um bei den späteren Aufzeichnungen des Chats die User-Rollen nachzuvollziehen.

Abbildung 1: Login

3.1.2 Startseite Testleiter

Regensburger Usability Testing Plattform

flowplayer
© 2008-2018 Flowplayer Ltd

00:00 00:00

Testmanager

[Einen bestehenden Test wählen >](#)
[Einen neuen Test anlegen >](#)
[Einen freien Test starten >](#)

Sie sind: Julia

Ihre Nachricht:

Annotations

Submit

Abbildung 2: Startseite Testleiter

Den Großteil des Bildschirms des Testleiters nimmt der Container für den Video-Stream ein. Darunter findet der Benutzer die Eingabemöglichkeit für Annotationen. Diese werden mit einem Zeitstempel in einer .txt Datei abgelegt und können am Ende des Tests heruntergeladen werden. Rechts unten auf der Webpage ist der Chat zu finden. Hier können Testleiter und Testperson miteinander kommunizieren. Das Versenden einer Nachricht erfolgt durch Drücken der Eingabetaste. Rechts oben befindet sich der Testmanager. Hierauf soll im Folgenden eingegangen werden.

Testmanager

Einen bestehenden Test wählen >

Einen neuen Test anlegen >

Einen freien Test starten >

Abbildung 3: Testmanager

Der Benutzer hat die Möglichkeit, einen bestehenden Test zu wählen, einen neuen Test anzulegen, oder einen freien Test zu starten. Letzterer wird über den Chat moderiert, daher wird darauf nicht näher eingegangen.

Wählen Sie einen Test

1	Test 1	30.8.2018	x
2	Test	3.9.2018	x
3	Test	3.9.2018	x

Neuer Test
Geben Sie einen Titel für den Test ein:

Titel

Tasks hinzufügen:

Task hinzufügen +

Test Speichern

Neuer Test
Geben Sie einen Titel für den Test ein:

Test 4

Tasks hinzufügen:

1. Task 1 x
2. Task 2 x

Task 3

Task hinzufügen +

Test Speichern

Neuer Test
Der Test wurde gespeichert.

zurück zum Testmanager

Abbildung 4: Test wählen / Neuen Test anlegen

Tasks:
Test 1
task1: Task 1

weiter

Tasks:
Alle Tasks wurden erledigt

Test beenden & Ergebnisse herunterladen

Abbildung 5: Task / Task beenden

Wählt der Testleiter einen gespeicherten Test aus, erscheint anstelle des Testmanagers ein Widget in welchem die zu dem Test gehörenden Tasks angezeigt werden. Der Testleiter kann diese weiterschalten, wenn die Testperson eine Aufgabe erledigt hat. Sind alle Tasks erledigt, erscheint ein Button, über den der Testleiter den Test beenden und die zugehörigen Daten herunterladen kann.

3.1.3 Startseite Proband

Die Seite der Testperson besteht lediglich aus dem Chat, sowie der Anzeige der Tasks. Hier hat die Testperson die Möglichkeit, die Task-Completion-Time aufzuzeichnen.

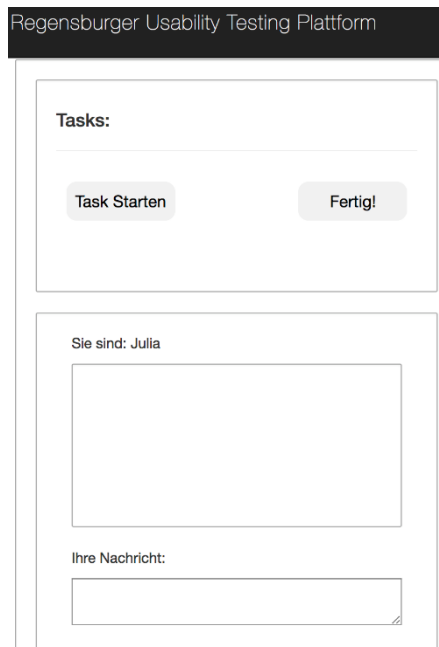


Abbildung 6: Startseite Proband

3.2. Hardware Interfaces

Für das Livestreamen sollten folgende Hardware Requirements eingehalten werden:

- Mindestens 2,3 GHz Dual-Core CPU
- 8 GB RAM Arbeitsspeicher
- Ordentliche Internet-Verbindung (v.a. Upload und Streamdelay)
- Kompatible Grafikkarte

3.3. Software Interfaces

Um die Plugins einzubinden, benötigt man die Software Open Broadcast Software (<https://obsproject.com/de>) sowie die jeweiligen Skripte, die über die Oberfläche von OBS eingebunden werden können. Dazu benötigt man jedoch eine kompatible Version von Python (3.5 oder neuere Version). Es gibt verschiedene Ansichten für die jeweiligen User dieser Software. Das User Interface des Testleiters beinhaltet einen Video Player zum Anzeigen für den Stream. Zusätzlich hat er noch eine „Annotation“ Fläche, in der er Kommentare für die Nachbearbeitung machen kann. Das UI des Probanden ist relativ simpel gehalten. Hier wird eine Fläche für den Chat sowie ein weiterer Reiter, für die Tasks bereitgestellt.

3.4. Kommunikations-Interfaces

Es wird eine Inter Verbindung benötigt, um den Livestream zu ermöglichen. Dabei ist vor allem die Upload Rate des Streamers von Bedeutung, um einen verzögerungsfreien Arbeitsablauf zu ermöglichen. Da es auf einen Server gehostet wird, ist die Kommunikation über den Chat möglich, zusätzlich kann über externe Quellen (Skype/Discord) eine mündliche Kommunikation stattfinden.

3.5. Serveranbindung

Als temporärer Hostserver wird Digital Ocean benutzt. Dort passieren alle wesentlichen Vorgänge der Software. Dieser Server muss bei der weiteren Nutzung selbst konfiguriert werden (kurze Anleitung entnehmen Sie bitte dem beigefügten Dokument: Server_Setup.pdf).

4. Systemkomponenten

In diesem Abschnitt werden die verschiedenen Funktionalitäten vorgestellt, die durch den Stakeholder genannt und gewünscht wurden.

4.1. Streaming des Testpersonen-Screens

Um das Testing durchzuführen, muss gewährleistet werden, dass der Stream ordnungsgemäß funktioniert. Dazu muss der Screen des Probanden gestreamed werden. Da es sich um die grundlegende Funktion handelt, ist die Priorität dieser Funktionalität sehr hoch, da ohne diese Funktion der ganze Ablauf nicht möglich ist.

Folgende Use Cases müssen durch diese Funktion abgedeckt sein:

- As User I want to be able to see the subject's screen during the test in realtime.

Damit das Streaming ordnungsgemäß funktioniert, müssen die Hardware Requirements erfüllt sein, sowie eine stabile Internetverbindung vorhanden sein. Hierbei ist die Upload Rate des Streamers wichtig, da es sonst zu Verzögerungen sowie Verschlechterung der Stream Qualität kommen kann, was den Ablauf stören würde. Ebenso muss eine stabile Internetverbindung auf der anderen Seite vorhanden sein. Eine verzögerungsfreie Streaming Methode ist momentan nicht möglich, weshalb eine Grundverzögerung von ca. 4-5 Sekunden mit einberechnet werden muss.

4.2. Screen Recording

Der Screen des Streamers muss aufgenommen werden, um nachträgliches Arbeiten zu ermöglichen, da bei einem Live Testing viele sensible Daten nicht direkt erkannt werden und erst in der Nacharbeit aufkommen. Da es sich hierbei um essentielle Daten handelt, ist auch diese eine Grundfunktion und besitzt daher eine hohe Priorität.

Folgende Use Cases müssen durch diese Funktion abgedeckt sein:

- As User I want to be able to save a recording of the subject's screen during the test
- As User I want to be able to record the subject's facial expressions during the test

OBS besitzt eine Recording Funktion, die man in den Einstellungen aktivieren kann. Dabei wird jeder Stream in ein gewähltes Videoformat lokal abgespeichert. Da ein solcher Test durchaus

länger andauern kann, kann es vorkommen, dass die Datei sehr groß wird und da der Stream beim Probanden läuft - aber beim Tester landen soll, ist dies nicht sinnvoll. Deshalb sollte der Tester entweder über OBS aufnehmen oder durch eine externe Software den Screen aufzeichnen. Eine Aufzeichnung des Probanden ist optional, vorausgesetzt der Proband ist damit einverstanden. OBS bietet die Möglichkeit eine „Face-Cam“ als Quelle anzugeben und in beliebiger Größe an einem Ort auf dem Stream zu platzieren. Dies könnte für manche Tests wichtig sein, muss aber auf Seiten der Probanden manuell eingestellt werden.

Ein Keylogger ist als Python Skript vorhanden und muss vor dem Streaming über OBS eingebunden werden (vgl. Dokumentation). Die Logs werden dann als Text Dateien lokal beim Probanden abgelegt. Diese müssen nach Beendigung des Tests an den Testleiter geschickt werden.

4.3. Task modelling

Die Tasks sind der wichtigste Bestandteil eines User-Testings und besitzen somit auch eine hohe Priorität. Das Hauptziel eines User Test ist es, die Funktionalitäten auf ihre Tauglichkeit zu überprüfen. Das Feedback der Probanden ist zu berücksichtigen und in das Finale Produkt mit ein zu beziehen. Die Tasks sollen hierbei wesentliche Funktionalitäten der Applikation abbilden und in einem Anwendungsfall soll herausgefunden werden, inwiefern der Proband das Nutzungsszenario erfolgreich absolviert hat.

Folgende Use Cases müssen durch diese Funktion abgedeckt sein:

- As User I want to be able to model tasks to show them on the tester's screen
- As User I want to be able to edit tasks

Die Tasks werden auf der Homepage angezeigt und können nur vom Testleiter hinzugefügt und editiert werden. Der Proband kann die Tasks sich anzeigen lassen, während der Testleiter diese hinzufügt oder editiert. Hierfür existieren auf der Homepage verschiedene Reiter.

4.4. Aufzeichnung von Tastaturaktivität

Alle Aktivitäten sollten aufgenommen werden, zwar sind diese nahezu komplett durch das Recording abgedeckt, aber die Tastatur Events sollten nochmals separat aufgezeichnet werden, um eine nachträgliche Datenauswertung zu vereinfachen. Diese sollen zusätzlich noch mit „Timestamps“ ergänzt werden, um die Datenauswertung zusätzlich zu vereinfachen. Außerdem sollten die Timestamps so gekennzeichnet werden, dass beispielsweise Taskrelevante Daten schneller gefiltert werden können.

Folgende Use Cases müssen durch diese Funktion abgedeckt sein:

- As User I want to get keylogs of all activity of the subject during the test
- As User I want to get an overview of hotspots/main areas of interest of the subject
- As User I want to be able to log a timeline for the test

Alle relevanten Daten werden im Regelfall lokal beim Probanden abgespeichert. Diese müssen nach Abschluss des Tests an den Testleiter übermittelt werden. Diese sollten nicht all zu groß werden, da es sich hierbei lediglich um eine .txt Datei handeln wird.

4.5. Aufnahme der Mimik der Testperson

Der Gesichtsausdruck eines Probanden soll nach dessen Erlaubnis aufgezeichnet werden können. Gesichtsausdrücke geben Informationen über die Gemütslage des Probanden wieder, was für User Interface spezifische Angelegenheiten wichtig sein kann.

Folgende Use Cases müssen durch diese Funktion abgedeckt sein:

- As User I want to be able to record the subject's facial expression during the test

Hierfür wird eine Webcam benötigt. Man kann in OBS eine Szene erstellen und diese so platzieren, dass Sie nur wenig Workspace in Anspruch nimmt.

4.6. Zeitsensible Anmerkungen

Es soll dem Nutzer möglich sein, Metadaten und Kommentare während des Tests zu machen. Diese sollten möglichst einfach hinzugefügt werden, damit der Fokus auf dem Test bleiben kann. Ebenfalls werden diese Informationen meist vergessen, wenn man diese nicht direkt mitnotiert, weshalb diese Funktionalität ebenfalls wichtig ist.

Folgende Use Cases müssen durch diese Funktion abgedeckt sein:

- As User I want to be able to add time-sensitive metadata to the recording of the test
- As User I want to be able to log a timeline for the test

Diese Funktion muss in wenigen Schritten erfolgen, da man diese Kommentare während des Tests anfertigt und den Workflow der getesteten Person sowie des Testers nicht unnötig lange unterbrechen will.

4.7. Kommunikation mit der Testperson

Die Kommunikation zwischen Testperson und Testleiter spielt eine wesentliche Rolle für einen erfolgreichen Test. Vor allem hinsichtlich des Remote-Testings, ist die Kommunikation umso wichtiger, da man sich nicht persönlich gegenübersteht, sondern durch eingeschränkte Möglichkeiten kommunizieren muss. Daher muss der Chat reibungslos funktionieren, um weder den Leiter noch den Probanden abzulenken, während diese den Test absolvieren.

Folgende Use Cases müssen durch diese Funktion abgedeckt sein:

- As User I want to be able to have a live-chat with the subject during the test

Zusätzlich zum Chat wäre es ratsam, über eine Voice Kommunikation nachzudenken. Diese kann über externe Softwares realisiert werden und würde effizienter funktionieren als ein Chat. Man müsste seinen Workflow bei einem Chat unterbrechen damit man schreiben kann, während bei einer Voice Kommunikation keine Schreibzeit benötigt wird und man mehr Information in einer kürzeren Zeit übermitteln kann.

5. Other Nonfunctional Requirements

1.1 Performance Requirements

Die Kernfunktion des Projekts besteht darin, einen Usability-Test mithilfe der OBS Software durchzuführen. Dieses wird hierbei für die Echtzeitübertragung eines Videostreams verwendet. Daher ist eine schnelle Internetverbindung der beteiligten Geräte vorausgesetzt. Auch sollten die für die Tests verwendeten Rechner daher für das Verwenden der Software geeignet sein. OBS selbst besitzt folgende Anforderungen¹¹:

Windows:

- DirectX 10.1 compatible GPU
- Windows 7 or newer

macOS:

- Intel CPU (PPC is not supported)
- OpenGL 3.2 compatible GPU
- macOS 10.10 or newer

Linux/Unix

- OpenGL 3.2 compatible GPU
- X window system

Obwohl die offiziellen Anforderungen dazu keine Angaben machen, sollten 8GB RAM und ein aktueller Prozessor vorausgesetzt werden.

1.2 Safety Requirements

Da es sich hierbei um Software handelt, ist eine Gefährdung durch die Benutzung so gut wie ausgeschlossen. Allerdings sollte hier dennoch die folgende Warnung vor photosensitiven Anfällen beachtet werden, auch wenn die Software keine unüblichen Lichteffekte verwendet:

„Bei manchen Personen kann es zu epileptischen Anfällen oder Bewusstseinsstörungen kommen, wenn sie bestimmten Blitzlichtern oder Lichteffekten ausgesetzt werden.

Diese Personen können bei der Benutzung von Computern einen Anfall erleiden. Es können auch Personen davon betroffen sein, deren Krankheitsgeschichte bislang keine Epilepsie aufweist und die nie zuvor epileptische Anfälle gehabt haben.

Falls bei Ihnen oder einem Ihrer Familienmitglieder unter Einwirkung von Blitzlichtern mit Epilepsie zusammenhängende Symptome (Anfälle oder Bewusstseinsstörungen) aufgetreten sind, wenden Sie sich bitte vor der Verwendung an Ihren Arzt.“

¹¹ <https://obsproject.com/wiki/System-Requirements>

1.3 Security Requirements

Die Software zeichnet den Tester sowie seine Eingaben während des Tests auf. Alle eventuell relevanten Datenschutzgesetze sind dabei zu berücksichtigen und die Testperson ist von diesem Umstand in Kenntnis zu setzen und muss ihm zustimmen.

Je nach verwendeter Software oder Serverstruktur zur Verarbeitung des Videostreams sollte auch auf die Datensicherheit geachtet werden, da dort ebenfalls die Datenschutzrechte des Getesteten verletzt werden können.

Daher sind die aufgezeichneten Videos mit Umsicht und der angebrachten Sorgfalt im Hinblick auf die Datenschutzrichtlinien zu behandeln.

1.4 Software Quality Attributes

Die Software soll im Rahmen des Kurses eine funktionierende Version des OBS Tools inklusive funktionierender Plugins oder Skripte beinhalten.

Ein funktionierender Server sowie dessen Implementation ist dahingegen nicht vorgesehen.

Für eine Benutzung des Tools ist daher ein geeigneter Server bereitzustellen, der den benötigten Voraussetzungen gerecht wird.

Die Plugins/Skripte sind verwendbar mit einer Standard OBS Version und können zu jeder neuen Installation des Programms hinzugefügt werden.

1.5 Business Rules

Bei der Benutzung des Tools können Testpersonen, vorgegebene Aufgaben bearbeiten und Tester können diesen Anweisungen oder neue Aufgaben zukommen lassen.

Die Funktionalitäten sind auf diese beiden Rollen beschränkt und können nicht vom jeweils anderen benutzt werden.

1.6 Other Requirements

OBS ermöglicht es lediglich, den aufgezeichneten Videostream bei der Darstellung zu verändern, bzw. extra Overlays oder Hinweise anzuzeigen.

Daher ist es nicht möglich auf dem Bildschirm der Testperson, auf dem gerade aufgezeichnet wird, auch neue Overlays mit testrelevanten Informationen oder Aufgaben darzustellen.

Das bedeutet, dass für die vorgesehene Benutzung des Tools zwei oder mehr Bildschirme benötigt werden.

Auf dem ersten wird das Verhalten der Testperson aufgezeichnet und auf dem zweiten kann der Tester mit der Testperson interagieren, etwa in dem er Anweisungen über das Tool an den Probanden schickt.