

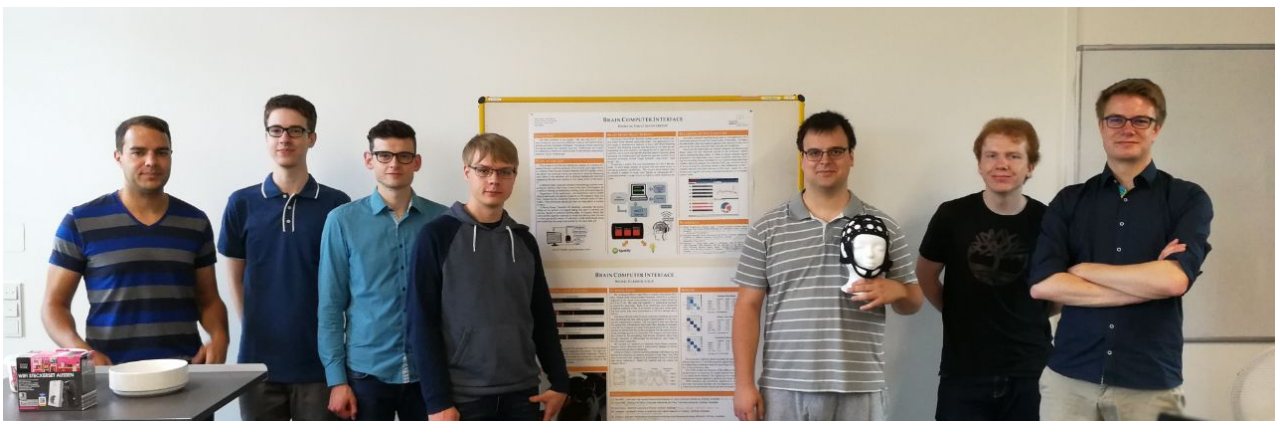


**Hochschule  
Augsburg** University of  
Applied Sciences

**Fakultät für  
Informatik**

# Wo issn des Hirn? 2.0

## Projektarbeitspezifische Dokumentation



Fabio Paul Aubele, Lukas Philipp Edlboeck, Michael Hammerl,  
Felix Kampfer, Florian Klein, Philipp Weber, Timo Wilhelm

Betreuer: Prof. Dr.-Ing. Hon. Dr. of ONPU Thorsten Schöler

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>3</b>
<b>2</b>	<b>Ergebnisse der Projektplanung</b>	<b>3</b>
2.1	Projektanforderungen . . . . .	3
2.2	Projektziele . . . . .	3
2.3	Stand der Technik . . . . .	4
2.4	Konzept . . . . .	4
<b>3</b>	<b>Implementierung</b>	<b>5</b>
3.1	Woche 1 - 2 Einführung Projektzielsetzung (15.03 - 29.03) . . . . .	5
3.2	Woche 2 - 5 Einarbeitung (29.03 - 12.04) . . . . .	5
3.3	Woche 5 - 12 Entwicklung der App und Klassifizierer (12.04 - 31.05) . . . . .	6
3.4	Woche 12 - 16 Fertigstellung und Projektvorbereitung(31.05 - 28.06) . . . . .	6
<b>4</b>	<b>Teamorganisation</b>	<b>6</b>
4.1	Software Engineering . . . . .	6
4.2	Milestones . . . . .	7
4.3	Projektplan . . . . .	7
4.4	Risklist . . . . .	8
4.5	Aufgabenverteilung . . . . .	9
<b>5</b>	<b>Tooling</b>	<b>9</b>
5.1	Project organisation tools . . . . .	9
5.2	Development toolchain . . . . .	10
<b>6</b>	<b>Zusammenfassung</b>	<b>10</b>
6.1	Nennenswerte Entscheidungen . . . . .	10
6.2	Gesammelte Erfahrungen . . . . .	11
6.3	Verbesserungsbedarf / für zukünftige Gruppen . . . . .	11
<b>7</b>	<b>Addendum</b>	<b>12</b>
7.1	Abbildungsverzeichnis . . . . .	12

# 1 Einführung

In diesem Dokument möchten wir unser Projekt "Wo issn das Hirn 2.0" vorstellen, das wir im Sommersemester 2017 durchgeführt haben. Dabei ging es darum basierend auf der Arbeit der Vorgruppe ein Brain-Computer Interface weiter zu entwickeln und dieses um eine Home-Automation Komponente zu erweitern. Im Folgenden möchten wir unsere Vorgehensweise und die Einteilung der Arbeit erläutern. Dabei Möchten wir sowohl auf unsere Ziele als auch auf den Ablauf des Projekts eingehen. Außerdem erklären wir noch die Teamorganisation und geben einen Überblick über das eingesetzte Tooling. In einer Zusammenfassung sammeln wir noch einmal die wichtigsten Erkenntnisse.

## 2 Ergebnisse der Projektplanung

### 2.1 Projektanforderungen

Bevor wir mit unserem Projekt starten konnten haben wir uns zunächst einmal mit dem bestehenden System der Projektgruppe des Vorsemesters vertraut gemacht und nachvollzogen wie das System im Allgemeinen arbeitet und funktioniert. Anschließend haben wir uns gemeinsam mit unserem Projektbetreuer Herr Prof. Dr. Thorsten Schöler, Gedanken bezüglich möglicher Projektanforderungen und möglichen Arbeitspaketen gemacht, da uns zunächst nicht bewusst war, in welche Richtung wir letztendlich das bestehende Projekt fortsetzen und weiterentwickeln wollten. Die ersten Ansätze die hierbei entstanden sind, beschäftigten sich in erster Linie mit der Verbesserung der bestehenden Algorithmen zur Signalerkennung- und Verarbeitung der Hardware. Der bereits existierende Ansatz einer Support Vektor Maschine hätte hierbei durch ein Neuronales Netzwerk ersetzt werden sollen. Eine weiterer Ansatz bestand dann darin, die bisher genutzte Hardware zu optimieren und zu verbessern. Umgesetzt werden sollte das ganze dahingehend, dass man die Länge der Kabel entsprechend anpasst, sodass vorhandene Interferenzen nicht mehr allzustark zur Geltung kommen und mögliche Signalübertragungen verfälschen bzw. zu stark beeinflussen. Ein letzter Ideenvorschlag bestand letztlich noch in der Entwicklung alternativer Anwendungen für das OpenBCI. Wir hatten beispielsweise das Manövrieren eines Rollstuhls in Betracht gezogen, um körperlich eingeschränkten Personen das Alltagsleben zu erleichtern. Eine nahezu vollständig gelähmte Person hätte hierbei mit Gedankenübertragung den Rollstuhl nach links- und rechts steuern und vor- oder zurückbewegen können. Bevor wir mit der Auswahl konkreter Anforderungen starten konnten mussten wir uns also noch überlegen, was letztendlich überhaupt machbar und umsetzbar ist. Wir mussten uns also neben Kriterien wie Zeit- und Arbeitsaufwand auch eine interne Projektgruppenstruktur überlegen, mit der wir unsere aus den Anforderungen hervorgehenden Ziele erreichen könnten.

Nach zahlreichen Überlegungen sind wir dann zu folgenden Ergebnissen gekommen, bzw. haben uns diese Anforderungen festgelegt:

- Überarbeiten bzw. Neuentwickeln bestehender Algorithmen und Classifier um die Effizienz und Erkennungsrate der Signale zu verbessern.
- Entwickeln einer App-Anwendung zum Anteuern von Geräten mit Anwendungskontext Smart-Home
- Verbessern der Elektrodenhalter und der Anbringungsmethodik von Haube und Elektroden zur Verbesserung der Messgenauigkeit

Auch aus zeitlichen Gründen haben wir uns entschieden, die uns zur Verfügung stehende Zeit in erster Linie mit den oben genannten Punkten auszufüllen und uns auf die Ausarbeitung dieser Aspekte zu konzentrieren.

### 2.2 Projektziele

Durch Definition unserer Anforderungen haben wir uns auch entsprechende Ziele gesetzt. Wir wollten in erster Linie ein möglichst stabil-laufendes System entwerfen, dass die Probleme der Vorgängergruppe hinsichtlich

der Datenübertragung- und Genauigkeit angeht und verbessert. Mit der Entwicklung besserer Classifier für das Training der Daten und Optimierung der Rahmenbedingungen während der Aufnahme (Eliminierung von Störfaktoren: Mobiltelefone, Licht- und Helligkeitsverhältnisse und Lautstärkepegel) sollte dieses Ziel verfolgt und erreicht werden. Mit Hilfe von Fachliteratur und professionellem Rat beim Klinikpersonal des örtlichen Krankenhauses wollten wir uns zudem nütlichen Rat einholen.

Weiterhin wollten wir eine Applikation entwickeln, die es uns ermöglicht, die Daten aus den Aufnahmen und OpenVibe, dazu zu nutzen um Geräte und Anwendungen ansteuern zu können. Wir haben uns hierbei speziell dafür entschieden Spotify in unsere App-Anwendung zu integrieren, um Songs aus einer Playlist abspielen, stoppen und zwischen einzelnen Liedern hin- und herwechseln zu können. Außerdem sollte mit der App eine Steckdose angesteuert, sprich ein- und ausgeschaltet werden können.

Zuletzt wollten wir noch die bestehenden Elektrodenhalter modifizieren und entsprechend anpassen. Uns ist anfangs, beim Anbringen der Haube aufgefallen, das die Kabel wahrlos am Kopf herumhingen, was unserer Meinung nach das Messergebnis auch ein wenig beeinflussen könnte. Aus diesem Grund wollten wir spezielle Elektrodenhalter entwickeln, auf denen wir die Kabel auf der Haube einhängen können, um den Kabelsalat im Nackenbereich zu reduzieren.

Für die Umsetzung der einzelnen Ziele haben wir letztendlich auch die Verwendung von Plots in Betracht gezogen, um unsere Daten uns einfach visualisieren zu können.

## 2.3 Stand der Technik

Vom technischen Standpunkt aus standen uns bei der Durchführung des Projekts das OpenBCI-Kit samt Board und USB-Dongle sowie Elektroden und den dazugehörigen Kabelsträngen zum Empfangen der gemessenen Hirnströme zur Verfügung. Diese wurden dann im OpenBCI, der dazugehörigen Software, visualisiert.

Darüberhinaus stand uns haben wir auch OpenVibe genutzt, mit dem wir uns eingene Classifier über die bereitgestellte Schnittstelle zusammenbauen und darüber hinaus eigene Classifier in Python schreiben konnten, die wir dann letztlich dort eingebunden haben.

## 2.4 Konzept

Das Konzept unserer Projektarbeit entsprach etwa dem Endprodukt der vorherigen Projektgruppe (Elektroden messen die Gehirnströme eines Subjekts, und werden von einem Signalverarbeitungsprogramm so weiterverarbeitet, dass ein daraus folgendes Steuerungssignal irgendwie die Umgebung beeinflussen kann) mit einigen Änderungen: Der Fokus soll auf SSVEP liegen, das Eingangssignal soll stabiler werden und der Klassifizierungsalgorithmus sollte verfeinert werden. Außerdem wollten wir neue Anwendungsmöglichkeiten ausprobieren, z.B. die Steuerung eines Rollstuhls.

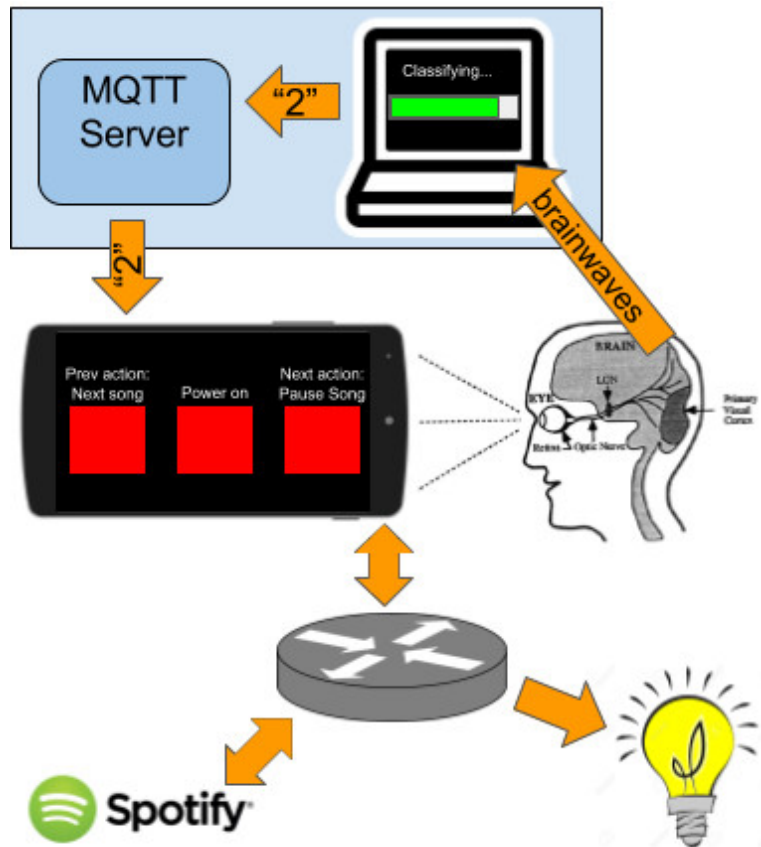


Abbildung 1: Übersichtsbild des fertigen Projekts

### 3 Implementierung

#### 3.1 Woche 1 - 2 Einführung Projektzielsetzung (15.03 - 29.03)

Nach dem ersten Treffen, bei dem sich das Team kennengelernt hat und der aktuelle Stand des Projektes erklärt wurde, wurden ein Brainstorming gemacht, um die Projektziele festzulegen zu können. Dabei wurde überlegt ob es sinnvoll ist, mit den Motorcortex so wie den Visualcortex gleichzeitig weiter zu arbeiten. Später im Projekt wurde dann entschieden nur den Visualcortex zu benutzen. Des weiteren wurde überlegt, für welchen Anwendungsfall die Daten benutzt werden können. Dabei wurde entschieden, dass eine Smart Home Anwendung geschrieben werden soll, die mithilfe von einer MQTT-Anbindung an den Klassifikator angeschlossen wird. Nach dem Erstellen des Anwendungsfalls wurde noch beschlossen, dass der aktuell verwendete Klassifizierer verbessert oder durch einen deep learning Ansatz verbessert werden sollte. Außerdem wurde noch beschlossen, dass die Qualität der Daten die aufgenommen werden, verbessert werden sollen indem zum Beispiel die Kabel gekürzt werden oder die Elektrodenhalter verbessert werden sollen.

#### 3.2 Woche 2 - 5 Einarbeitung (29.03 - 12.04)

Nachdem die Projektziele definiert wurden, fing die Gruppe damit an sich in das alte System einzuarbeiten. Dabei wurde der notwendige Softwarestack aufgesetzt und versucht das alte System testweise auszuführen. Da dabei Probleme auftraten da das System an manchen Stellen nicht funktionierte und es bei dem anbringen der Elektroden an dem Kopf so wie das auswerten der Gehirnsignale Probleme gab, traf sich die aktuelle Projektgruppe mit der Vorgängergruppe. Diese zeigte uns wie die Elektroden richtig an dem Kopf anzubringen sind und wie das alte System funktioniert. Außerdem erklärten sie uns den Unterschied zwischen den Signalen

von den zwei Gehirnkortexen. Da es bei dem visuelle Cortex mehr erkennbare Signale gab als bei dem Motorcortex, hat sich die Gruppe dazu entschieden sich auf den visuellen Cortex zu konzentrieren.

Da das System für die Benutzung antrainiert werden musste, wurden Testdaten benötigt, die für das Training benutzt werden können. Da es am Anfang Probleme gab eigene Testdaten zu erzeugen, suchten wir nach anderen Testdaten. Diese waren jedoch meistens in einem Format das für unsere Zwecke nicht sinnvoll war. Daher erstellten wir Konverter die die gefundenen Daten in unser Format umwandeln können.

### **3.3 Woche 5 - 12 Entwicklung der App und Klassifizierer (12.04 - 31.05)**

Nachdem das alte System verstanden wurde und benutzt werden konnte, wurde mit der Umsetzung der Projektziele begonnen. Für die Smart Home Anwendungen wurden Steckdosen mit einer Wlan Schnittstelle besorgt. Damit diese direkt von unseren System angesprochen werden konnten, musste ein Skript entwickelt werden das Steuersignale an die Steckdosen schicken kann. Neben dem Ansteuern der Steckdosen wurde mithilfe der API von Spotify eine Skript geschrieben, das es ermöglicht Musik abzuspielen, ohne dabei Spotify öffnen zu müssen.

Für das Steuern wurde eine App entworfen, die einerseits die Steckdosen so wie die Spotify Anbindung steuern kann andererseits aber auch die visuelle Anzeige besitzt mit der, über die Klassifizierer, die App gesteuert wird. Bei der visuellen Anzeige handelt es sich um drei, in unterschiedlichen Frequenzen, flackernde Boxen. Neben dem Entwickeln der App für das steuern der Smart Home Anwendungen wurde auch der Klassifizierer verbessert. Am Anfang wurde noch der, von der vorherigen Gruppe benützte, Naive Base Ansatz verwendet worden. Unser erster Ansatz war es ein Multilayer Neural Network zu benützten. Dieses war schon in Open-Vibe implementiert. Da jedoch das Training zu lange dauerte und das Ergebnis nicht die Erwartungen erfüllte, wurde ein anderer Ansatz verfolgt. Die Support Vector Maschine, die wir als nächsten Ansatz ausprobiert haben, war schneller als das Multilayer Neural Network jedoch waren die Ergebnisse nicht viel besser als bei dem Multilayer Neural Network Ansatz. Nach der Support Vector Maschine probierten wir ein Convolutional neural network aus. Mit diesen Ansatz wurden dann die, im Gegensatz zu den anderen Ansätzen das beste Ergebnis erzeugt.

### **3.4 Woche 12 - 16 Fertigstellung und Projektvorbereitung(31.05 - 28.06)**

Nachdem die App so wie die Klassifizierer entwickelt wurden, wurden die beiden Programme mit Hilfe eines MQTT-Servers zusammengefügt. Dabei zeigt die App die flackernden Boxen an und der Klassifizierer sendet die ausgewerteten Daten, über MQTT, an die App.

Nachdem das System komplett war wurde dieses getestet. Dabei sind noch Fehler bei der App erkannt worden. Nachdem diese gefixt wurden, war das System fertig.

Danach wurden die Plakate für den Projekttag erstellt. Dabei hat sich eines der Plakate mit den Klassifizierer beschäftigt und das andere mit der Smart Home Anwendung.

## **4 Teamorganisation**

### **4.1 Software Engineering**

Unsere Softwareentwicklungsmethodik bestand hauptsächlich aus einem agilen Ansatz und Prototyping. Innerhalb der Gruppe trafen wir uns wöchentlich einmal um die Fortschritte, Ziele und Probleme zu besprechen. Dies geschah in Form eines Daily-Scrum-Meeting und hatte das Ziel agile Methodiken in die Planung einzuführen. Wir entschieden uns jedoch für einen wöchentlichen Rhythmus da dies besser zu der Arbeitsweise eines Semesterprojektes passt. Die Treffen liefen dabei immer gleich ab. Anfangs erzählt jeder Teilnehmer, was er über die Woche erledigt hat, welche Probleme er dabei hatte und was er sich für die nächste Woche vornimmt. Danach wurde eine allgemeine Diskussionsrunde gestartet, in welcher offene Punkte, Organisatorisches oder allgemeine Probleme nochmals tiefer besprochen wurden. Dies half uns das Projekt anpassbar und überschaubar zu halten.

Um die gewünschte Transparenz in einem agilen Projekt zu bieten, haben wir für jedes Treffen einzelne Berichte erstellt, welche die besprochenen Punkte beinhaltet. Innerhalb des Projektes dokumentierten wir wichtige Fortschritte stets in das GitLab Wiki oder unter die entsprechenden Issues. Wichtige Dokumente wie die Risklist oder den allgemeinen Projektplan wurden anfangs angefertigt und während des ganzen Projektes über auf einem aktuellen Stand gehalten.

Innerhalb der Projektgruppe haben wir mithilfe des GitLab-Issue-Systems alle zu erledigenden Aufgaben gesammelt und uns diese gegenseitig zugewiesen. Bei Fortschritt oder auftretende Problemen konnten wir uns mithilfe der Kommentarfunktion koordinieren. Dies half uns bei aufeinander aufbauenden Anforderungen und unterstützte uns erneut Transparenz und Anpassbarkeit zu bieten.

Bei der Implementierung der verschiedenen Anforderungen entschieden wir uns für einen Ansatz des Prototyping. So wurden anfangs vorerst rudimentäre Prototypen erstellt und diese dann Schritt für Schritt erweitert und verbessert. Dies hatte den Vorteil, dass wir uns innerhalb der wöchentlichen Projekttreffen neu abstimmen konnten, falls es irgendwo in eine falsche Richtung ging. Dies ermöglichte das Projekt stets anpassbar und agil zu halten. Insbesondere bei einem recht wissenschaftlichen Vorhaben wie dieses, bat sich ein solcher Ansatz an. Hier kam es nämlich öfter vor erreichte Ergebnisse zu verwerfen und sich in eine andere Richtung zu entwickeln.

## 4.2 Milestones

Das Projektteam hat über das Semester hinweg einige Ereignisse als signifikanten Projektfortschritt deklariert. Diese Milestones sind allesamt auch in dem unteren Projektplan als aufgelistet.

- Recherche über das Thema und den Fortschritt der letzten Projektgruppe
- Kennenlernen von Datasets und vergleichen der verschiedenen Tools zum Erstellen von classifier
- Erstellen eines 'Standalone classifier'
- Entwickeln der Smart Home App
- Zusammenfügen der Smart Home App und des classifier
- Demo der entwickelten Anwendung

## 4.3 Projektplan

Um den Überblick des Zeitmanagements innerhalb des Projektes zu behalten, wurde ein Gantt-Diagramm erstellt und über den Zeitraum des Projektes entsprechend erweitert. Dies half uns bei der Planung der verschiedenen Arbeiten und dem setzen von Milestones und Deadlines.

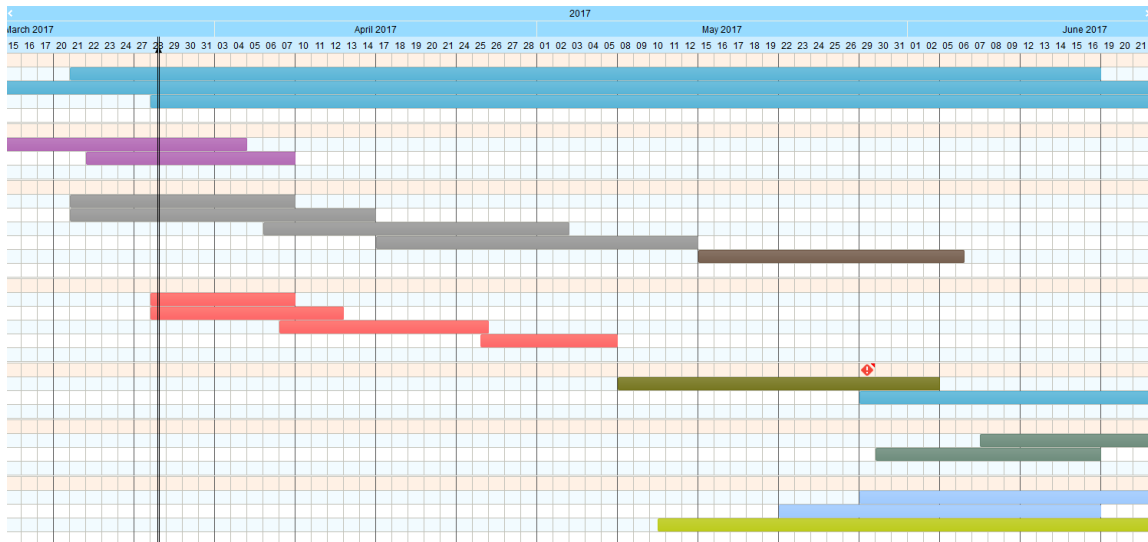


Abbildung 2: Erstellter Projektplan

Der gesamte Plan, wurde mit dem kostenlosen online Tool TomsPlanner erstellt und kann über diesen [Link](#) angezeigt werden.

#### 4.4 Risklist

Innerhalb unseres Projektes hat das Team eine Abschätzung über mögliche Risiken, sowie Möglichkeiten mit diesen umzugehen, erstellt. Diese Tabelle wurde zentral in unser Wiki gehalten und wurde über unsere Arbeit hinweg aktualisiert. In folgendem sieht man die anfängliche Version der Risklist.

##### Risklist

risk	probability	impact	overall	priority
time management	9	10	90	very high
installing the electrodes	9	8	72	high
hardware fail	7	10	70	high
theme missing	6	10	60	medium
wrong implementation	5	8	40	low
lack of quality	4	9	36	low
data loss	3	9	27	low
staff shortages	2	5	10	negligible
conflicts between project members	1	6	6	negligible
robustness	0	5	0	negligible

Abbildung 3: Erstellte Risklist, anfängliche Version

Als mögliche Gegenmaßnahmen für die oben definierten Risikos haben wir uns folgende überlegt:

- **time management:** Befolgen und aktualisieren des Gant-Diagramm: [Link](#).
- **installing the electrodes:** Die letzte Projektgruppe und wenn möglich Dr. Naumann kontaktieren.
- **hardware fail:** Die vorhandene Hardware vorsichtig behandeln und neue Hardware bestellen.



- **theme missing:** Erstellen eines deutlichen Plans, was alle grundlegenden Fragen beantwortet.

## 4.5 Aufgabenverteilung

Anfänglich haben wir unser Team in zwei Teilgruppen aufgeteilt, welche sich jeweils um das Erreichen eines großen Zieles gekümmert haben:

Team	Personen
Neuronales Netzwerk	Timo Wilhelm, Felix Kampfer, Phillip Weber, Lukas Edlböck
Smart Home Applikation	Florian Klein, Fabio Aubele, Michael Hammerl

Weiterhin hatten wir innerhalb der Projektgruppe einige Rollen vergeben:

Rolle	Person
Dokumentation	Michael Hammerl
Team-Leiter	Felix Kampfer
Git-Management	Lukas Edlböck
Protokollant	Felix Kampfer
Risikomanagement	Fabio Aubele
Foto-Dokumentation	Florian Klein

Abschließend sollen die erledigten Aufgaben und die jeweils zuständigen Personen aufgezeigt werden:

Aufgaben	Personen
Allgemeine Recherche zum Thema	alle Projektteilnehmer
Kontakt zum Klinikum Augsburg	Phillip Weber
Nachvollziehen der Ergebnisse der Vorgruppe	Fabio Aubele, Florian Klein
Ansteuern der Steckdosen (testweise und in der App)	Michael Hammerl, Florian Klein
Spotify (Ansprechen der API, Programmieren der Funktionalität)	Fabio Aubele
App-Programmierung allgemein (UI, Animator, Konfigurationsseite)	Fabio Aubele, Michael Hammerl
Erstellung der Datenkonverter	
Zusammenführen der Systeme (über MQTT)	Timo Wilhelm, Felix Kampfer, Phillip Weber, Lukas Edlböck, Fabio Aubele
Elektrodenhülsen (Design, 3D-Druck)	Florian Klein
Erstellen und Drucken von Plakaten	Timo Wilhelm, Felix Kampfer, Phillip Weber, Lukas Edlböck, Fabio Aubele
Dokumentation (Risklist, Projektplan, Entwürfe, etc.)	alle Projektteilnehmer

## 5 Tooling

### 5.1 Project organisation tools

- *GitLab Community Edition* hosted at the datacentre of the University of Applied Sciences Augsburg as version control for source code and development files.
- *WhatsApp Messenger* service for cross-platform instant messaging.

- *OwnCloud* hosted at the datacentre of the University of Applied Sciences Augsburg for sharing of big binary files, documents and media content.

## 5.2 Development toolchain

- *OpenViBE* software platform for design, test and use of Brain-Computer Interfaces and real-time EEG acquisition visualization and processing.
- *OpenBCI GUI 2.0* TO visualize, record, and stream data from the OpenBCI Board.
- *Lua* scripting language for implementation of custom OpenVibe data-pipeline elements, primarily used for configuration and acquisition.
- *Python* programming language for processing of real-time data from OpenVibe, primarily used for classifier training, testing and custom data visualization.
- *scikit-learn* machine learning library for Python, to design and train the statistical classifiers.
- *Lasagne* python library to design neural networks in the Python programming language with Theano.
- *nolearn* wrapper library around lasagne to a common interface compatible with scikit-learn
- *Theano* numerical computation library for Python, that features deep integration with NumPy and allows for efficient evaluation of multi-dimensional array operations on CPU or GPU.
- *imbalanced-learn* re-sampling functions to balance supervised learning datasets.
- *NumPy* high-level mathematical functions for large multi-dimensional data arrays.
- *GCC* compiler suite for compilation of Theano computational graph structure of the neural network.
- *OpenBlas / Intel-MKL* highly optimized low-level Basic Linear Algebra Subprograms for common linear algebra operations.
- *MinGW* open-source tool set for development on windows environments, including a port of the GNU Compiler Collection (GCC)
- *Mosquitto* open source message broker implementing the Message Queue Telemetry Transport (MQTT) protocol for cross-device communication over local networks.

## 6 Zusammenfassung

### 6.1 Nennenswerte Entscheidungen

Bereits zu Beginn des Projekts wurde die Gruppe vor eine wichtige Entscheidung gestellt. Es standen zwei Ansätze der Auswertung der Gehirnströme zu Verfügung. Die Möglichkeiten sowie Limitierungen der beiden Varianten wurden zunächst mit dem Betreuer erörtert. Nachdem diese Diskussion nochmals in einer Telefonkonferenz mit dem Betreuer der Partnerhochschule der Ulster University, Dr. Leo Galway, vertieft wurde, war sich das Projektteam in der Entscheidung einig. Aufgrund der vielfältigeren Anwendungs- und Entwicklungsmöglichkeiten fiel die Wahl auf den Schwerpunkt SSVEP.

Da die vorherige Projektgruppe bereits nennenswerte Ergebnisse in diesem Bereich verzeichnen konnte, waren bereits wichtige grundlegende Erkenntnisse und Abläufe vorhanden. Für den weiteren Verlauf des Projekts mussten Schwerpunkte gewählt werden, die eine Weiterentwicklung bestehender Systeme oder die Umsetzung eigener Ideen betrifft. Im Bereich der Datenanalyse war es dem Team wichtig, die Genauigkeit und Performance der Classifier weiter zu entwickeln. Für die Demonstration möglicher Anwendungsfälle sollte

zudem ein neues, geeignetes Szenario entwickelt werden. In diesem Bereich war sich das Team schnell über das Thema Home Automation als Anwendungsbeispiel einig. Für die Auseinandersetzung mit diesen beiden Themengebieten wurde das Projektteam in zwei Gruppen unterteilt. Die Aufteilung geschah grundsätzlich auf freiwilliger Basis und konnte erfreulicherweise zur Zufriedenheit jedes Teilnehmers wunschgemäß durchgeführt werden.

## 6.2 Gesammelte Erfahrungen

Die Aufteilung in die zwei Untergruppen hat sich in dem Projekt bewährt, da wir innerhalb der Gruppen besser Treffen organisieren konnten, die für alle Teammitglieder passend waren. Allerdings muss man gegen Ende des Projekts Zeit für die Integration der Ergebnisse der Einzelgruppen einplanen bei denen alle Projektteilnehmer anwesend sind.

Der Besuch der Neurologie des Klinikum Augsburg hat uns enorm geholfen Probleme mit unserem Projektaufbau zu erkennen. Der Zeitpunkt für den Besuch gegen Mitte des Projekts war gut gewählt, um genug Vorwissen für konstruktive Fragen mitzubringen.

Die Mitschriften der Projekttreffen im Wiki hat sich beim Rekonstruieren des Projektablaufs als hilfreich erwiesen. Bei den Projekttreffen hat sich bewährt eine Runde nach dem Scrum Stand-Up Meeting Prinzip durchzuführen um alle Teammitglieder über den aktuellen stand und die Arbeitspakete sowie Probleme der Gruppe aufzuklären. Besonders wichtig war auch das Treffen mit der Vorgruppe um implizites Wissen auszutauschen und einen Schnellen Start ins Projekt zu ermöglichen, da viele Prozesse bei der EEG-Datenverarbeitung sehr komplex sind.

## 6.3 Verbesserungsbedarf / für zukünftige Gruppen

Wenn die Zielsetzung des Projekts feststeht, sollte ein Zeitplan mit Meilensteinen erstellt werden um zu erkennen ob das Projekt im Zeitplan liegt. Außerdem ist es entscheidend, die Issues im Git bis zum Ende des Projekts kontinuierlich zu Pflegen um zu vermeiden, dass Arbeitspakete untergehen. Wichtig ist, einige Wochen vor Ende des Projekts einen Feature-Freeze einzuplanen um sich auf die Dokumentation und Vorbereitung auf den Projekttag konzentrieren zu können.

Eventuell sollte das Wiki noch eine Unterteilung für die verschiedenen Projektgruppen aufweisen, um veraltete oder überholte Informationen besser zu Erkennen. Auch auf die Ordnerstruktur im Git Projekt sollte geachtet werden, damit alle Teammitglieder schnell auf die Relevanten Dateien zugreifen können. Auch eine aussagekräftige Benennung von aufgezeichneten Datensets ist entscheidend um diese auch in Zukunft weiter benutzen zu können. Seltener benötigte, größere Dateien wie Videos oder Bilder sollten in der OwnCloud statt im Git abgelegt werden um die Zeit für das Kopieren des Projekts zu reduzieren.

## 7 Addendum

### 7.1 Abbildungsverzeichnis

1	Übersichtsbild des fertigen Projekts . . . . .	5
2	Erstellter Projektplan . . . . .	8
3	Erstellte Risklist, anfängliche Version . . . . .	8