



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Masterthesis

Silke Katharina Termer

Umsetzung verschiedener Konzepte zur  
Aufnahme und Auswertung betrieblicher  
Rückmeldedaten für eine flexible Fertigung

Silke Katharina Termer

Umsetzung verschiedener Konzepte zur Aufnahme  
und Auswertung betrieblicher Rückmeldedaten für  
eine flexible Fertigung

Masterthesis eingereicht im Rahmen der Masterprüfung  
im Masterstudiengang Automatisierung  
am Department Informations- und Elektrotechnik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. -Ing. Florian Wenck  
Zweitgutachter : Prof. Dr. -Ing. Marc Hensel

Abgegeben am 6. März 2019

**Silke Katharina Termer**

**Thema der Masterthesis**

Umsetzung verschiedener Konzepte zur Aufnahme und Auswertung betrieblicher Rückmeldedaten für eine flexible Fertigung

**Stichworte**

Manufacturing Execution Systems (MES), Enterprise Resource Planning (ERP), Unternehmensleitebene, Prozessebene, Speicherprogrammierbare Steuerung (SPS), Webseite, OPC UA, Durchlaufdiagramme, Datenbank

**Kurzzusammenfassung**

Diese Masterarbeit umfasst die Konzeption und Realisierung eines Systems, das die Rückmeldedaten einer flexiblen Fertigung aufnimmt und auf einer Webseite visuell darstellt. Dafür werden die Rückmeldedaten in Durchlaufdiagrammen und in tabellarischer Form dargestellt. Die Rückmeldedaten werden über die Steuerungen gewonnen und mittels OPC UA Protokoll übertragen. Der Webserver ist auf einem Raspberry Pi implementiert. Hier werden auch die Rückmeldedaten mittels OPC UA empfangen und in einer Datenbank gespeichert.

**Silke Katharina Termer**

**Title of the paper**

Implementation from different concepts for recording and analysing confirmation data for a flexible fabrication

**Keywords**

manufacturing execution systems (MES), enterprise resource planning (ERP), corporate management level, process level, programmable logic controller, website, OPC UA, input/ output diagram, data base

**Abstract**

Inside this report the construction and realisation of a system to recording and visualise the confirmation data from the flexible fabrication is described. The visualization will show the confirmation data in a table and in an input/ output diagram. The recorded data will be transmitted with the OPC UA protocol. The website is implemented at a Raspberry Pi. On the Raspberry Pi also the confirmation data will be received and saved in a data base.

## **Danksagung**

An dieser Stelle möchte ich mich bei allen bedanken, die mich während meines Studiums, und im Besonderen der Anfertigung dieser Masterarbeit, unterstützt und motiviert haben.

Zuerst gebührt mein Dank Herrn Prof. Dr. -Ing. Florian Wenck für die gute Betreuung und die Möglichkeit, diese Masterarbeit zu erstellen. Ein weiterer Dank geht an Herrn Prof. Dr. -Ing. Marc Hensel der die Zweitkorrektur dieser Arbeit übernommen hat. Ebenfalls möchte ich mich bei den wissenschaftlichen Mitarbeitern des Fachbereiches Automatisierungstechnik bedanken, die mir bei technischen Problemen in der Modellfabrik und sowie Bestellungen immer zur Seite standen. Weiter danke ich allen Korrekturlesern meiner Arbeit.

Außerdem möchte ich mich bei meinem Freund, meinen Freunden und meiner Lerngruppe, die ich mittlerweile zu meinen Freunden zähle, bedanken. Danke für die Unterstützung, die Zusprüche und das mich Aushalten während dieser Arbeit und des gesamten Studiums.

Mein besonderer Dank geht an meine Geschwister, die mich zu diesem Studium ermutigt und es mir ermöglicht haben. Vielen Dank für die finanzielle Unterstützung, das Dau-mendrücken, ein offenes Ohr und noch soviel mehr.

# Inhaltsverzeichnis

<b>Tabellenverzeichnis</b>	<b>VIII</b>
<b>Abbildungsverzeichnis</b>	<b>IX</b>
<b>Abkürzungsverzeichnis</b>	<b>XI</b>
<b>1. Einführung</b>	<b>1</b>
1.1. Einführung und Motivation . . . . .	1
1.2. Ziele der Arbeit . . . . .	3
1.3. Gliederung der Arbeit . . . . .	3
<b>2. Grundlagen</b>	<b>4</b>
2.1. Produktionslogistischer Hintergrund . . . . .	4
2.1.1. Manufacturing Execution Systems . . . . .	4
2.1.2. Erfassung von betrieblichen Rückmeldedaten . . . . .	5
2.1.3. Visualisierung betrieblicher Rückmeldedaten . . . . .	6
2.2. OPC UA . . . . .	11
2.3. Scriptsprachen und Entwicklungstools . . . . .	13
2.3.1. Python . . . . .	13
2.3.2. Webseiten . . . . .	14
2.3.3. Datenbanken . . . . .	15
2.3.4. Entwicklungstools . . . . .	16
<b>3. Anlagenbeschreibung</b>	<b>17</b>
3.1. Gesamtübersicht . . . . .	17
3.2. Prozessablauf . . . . .	19
3.3. Beschreibung der Stationen . . . . .	21
3.4. Aufbau der Kommunikation . . . . .	26
<b>4. Konzeption</b>	<b>27</b>
4.1. Graphische Oberfläche zur Visualisierung der Rückmeldedaten . . . . .	27
4.1.1. Anforderungen . . . . .	27
4.1.2. Aufteilung der Anlage und Definition der relevanten Daten . . . . .	28
4.1.3. Darzustellender Inhalt . . . . .	34

---

4.1.4. Layout . . . . .	36
4.2. Datenbereitstellung und -verarbeitung . . . . .	43
4.2.1. Anforderungen . . . . .	43
4.2.2. Teil-Alternativen . . . . .	45
4.2.3. Gesamt-Alternativen . . . . .	51
4.2.4. Vergleich und Entscheidung . . . . .	54
4.2.5. Weitere zu beachtende Überlegungen . . . . .	57
4.3. Hardware . . . . .	60
4.3.1. Server . . . . .	60
4.3.2. Anzeigesystem . . . . .	60
4.3.3. WLAN-Bridge . . . . .	63
4.3.4. Mobiles Anzeigesystem . . . . .	64
4.4. Zusammenfassung des Gesamtkonzeptes . . . . .	64
<b>5. Realisierung</b>	<b>66</b>
5.1. Annahmen zum Betrieb der Modellfabrik . . . . .	66
5.2. Hardware . . . . .	67
5.2.1. Raspberry Pi einrichten . . . . .	67
5.2.2. Einrichten der WLAN-Bridge . . . . .	68
5.2.3. Anzeigesysteme einrichten . . . . .	68
5.3. Datenbereitstellung und -verarbeitung . . . . .	69
5.3.1. Datengewinnung . . . . .	69
5.3.2. Empfangen und Verarbeiten der Daten . . . . .	71
5.3.3. Speicherstruktur der Datenbank . . . . .	75
5.4. Graphische Oberfläche zur Visualisierung der Rückmeldedaten . . . . .	75
5.4.1. Verwendete Bibliotheken . . . . .	75
5.4.2. Struktur und Design der Webseite . . . . .	76
5.4.3. Steuerung der Webseiten . . . . .	83
5.4.4. Auslesen der Datenbank . . . . .	89
5.4.5. Darstellung der Durchlaufdiagramme . . . . .	89
5.4.6. Darstellung der Tabellen . . . . .	90
5.4.7. Hinweise zur Auswertung der Daten auf den historischen Webseiten . . . . .	90
5.5. Zusammenfassung des realisierten Gesamtsystems . . . . .	90
<b>6. Verifikation</b>	<b>92</b>
6.1. Überprüfung der Anforderungen . . . . .	92
6.1.1. Hardware . . . . .	92
6.1.2. Visualisierung der Rückmeldedaten . . . . .	92
6.1.3. Datenbereitstellung und -verarbeitung . . . . .	93
6.2. Test des Gesamtsystems . . . . .	94
6.2.1. Normalbetrieb . . . . .	94

---

6.2.2. Elektrischer Fehler . . . . .	95
6.2.3. Optischer Fehler . . . . .	96
6.2.4. Fazit . . . . .	98
<b>7. Zusammenfassung und Ausblick</b>	<b>99</b>
7.1. Zusammenfassung . . . . .	99
7.2. Ausblick . . . . .	100
<b>Literaturverzeichnis</b>	<b>102</b>
<b>A. Anhang</b>	<b>105</b>
A.1. Bildschirmpräsentation Mobil und Leitstand (DVD) . . . . .	105
A.2. Anleitung Inbetriebnahme der Modellfabrik (DVD) . . . . .	105
A.3. Passwörter (DVD) . . . . .	105
A.4. Konfiguration des Anzeigesystems (DVD) . . . . .	105
A.5. TIA-Projekt der Modellfabrik (DVD) . . . . .	105
A.6. Quellcode Webseiten und Python (DVD) . . . . .	106
A.7. Tabellen und Listen (DVD) . . . . .	106
A.7.1. Steuerungsprogramm . . . . .	106
A.7.2. OPC UA . . . . .	106
A.7.3. Datenbank . . . . .	106
A.8. Video Normalbetrieb (DVD) . . . . .	106

# Tabellenverzeichnis

4.1. Betrachtung des Gesamtprozesses in verschiedenen Detailtiefen . . . . .	32
4.2. Übersicht der gewichteten Anforderungen an die Datenbereitstellung und - verarbeitung . . . . .	45
4.3. Morphologischer Kasten . . . . .	51
4.4. Entscheidungstabelle Datenbereitstellung und -verarbeitung . . . . .	56
4.5. Entscheidungstabelle Anzeigesystem . . . . .	62
5.1. Status <i>stop.txt</i> . . . . .	77
5.2. Verhalten der Button auf Status des Pythonprogramms . . . . .	77
5.3. Zuordnung der Schnitte zu der Variable <code>anzeige</code> bei vier und sechs Schnitten	84
6.1. Auflistung der Gesamtkosten der Hardware [33] . . . . .	92

# Abbildungsverzeichnis

1.1. Gesamtansicht der Modellfabrik . . . . .	2
2.1. Einordnung eines MES in ein Unternehmen . . . . .	5
2.2. Durchlaufzeit (nach [29]) . . . . .	6
2.3. Trichtermodell [29] . . . . .	7
2.4. Verschiedene Arten der Vernetzung von Trichtern (nach [31]) . . . . .	8
2.5. Durchlaufdiagramm (nach [29]) . . . . .	9
2.6. Durchlaufdiagramm mit Bestandskurve (nach [29]) . . . . .	10
2.7. Mögliche Betriebszustände eines Arbeitsvorganges (nach [29]) . . . . .	10
2.8. Schichtenmodell von OPC UA (nach [20]) . . . . .	12
2.9. Schichten einer Webseite (nach [27]) . . . . .	14
2.10. Beispiel einer Datenbankabfrage mittels Ajax und PHP . . . . .	15
3.1. Bauteile und Fertigprodukt . . . . .	17
3.2. Schematische Darstellung der Gesamtanlage . . . . .	18
3.3. Ablaufdiagramm bei Wiedereinlastung . . . . .	20
3.4. Trichtermodell des Gesamtprozesses . . . . .	21
3.5. Schematische Darstellung der Station 10 . . . . .	22
3.6. Schematische Darstellung der Station 20 . . . . .	23
3.7. Schematische Darstellung der Station 30 . . . . .	24
3.8. Schematische Darstellung der Station 60 . . . . .	25
4.1. Anlagenschnitte durch Arbeitsvorgänge ohne Ausgangslager . . . . .	29
4.2. Anlagenschnitte durch Arbeitsvorgänge mit Ausgangslager . . . . .	30
4.3. Anlagenschnitte durch Zusammenfassung von Arbeitsschritten mit Ausgangslager . . . . .	31
4.4. Darzustellende Inhalte . . . . .	35
4.5. Design der Startoberfläche . . . . .	37
4.6. Navigationsleiste für die Oberfläche . . . . .	38
4.7. Layout Mobil für aktuelle Daten am Beispiel Bestückung . . . . .	39
4.8. Layout Mobil für historische Daten am Beispiel Bestückung . . . . .	40
4.9. Layout Leitstand für aktuelle Daten am Beispiel Bestückung . . . . .	41
4.10. Layout Leitstand für historische Daten am Beispiel Bestückung . . . . .	42

---

4.11. Kombinationen 1 bis 4 der Teil-Alternativen des morphologischen Kastens . . .	51
4.12. Kombinationen 5 bis 12 der Teil-Alternativen des morphologischen Kastens . .	52
4.13. Ablaufdiagramm seriell Pythonprogramm . . . . .	57
4.14. Ablaufdiagramm Pythonprogramm mit Subprozess (SP) . . . . .	58
4.15. Alternative 1: Jede Steuerung erhält ihre eigene WLAN-Schnittstelle . . . . .	58
4.16. Alternative 2: Es existiert eine WLAN-Schnittstelle zur Datenübertragung . . .	58
4.17. Schematische Darstellung des Gesamtaufbaus . . . . .	65
5.1. Zähler des Zugangs Funktionsprüfung . . . . .	70
5.2. Ablaufdiagramm des Subprozesses (Job) . . . . .	72
5.3. Ablaufdiagramm Python Hauptprogramm . . . . .	73
5.4. Sequenzdiagramm zum Starten und Stoppen des Pythonprogramms . . . . .	78
5.5. Aufteilung der Leitstand Webseite aktuell in Bereiche . . . . .	78
5.6. Unterteilung des Bereiches Navigation . . . . .	79
5.7. Unterteilung des Bereiches Übersicht . . . . .	80
5.8. Unterteilung des Bereiches Gesamtanlage . . . . .	81
5.9. Unterteilung des Bereiches Detailschnitt . . . . .	81
5.10. Unterteilung der mobilen Webseite für aktuelle Daten . . . . .	82
5.11. Unterteilung der Navigationsliste für die historische Webseite . . . . .	83
5.12. Ablaufdiagramm zur Berechnung der Variable <code>anzeige</code> . . . . .	85
5.13. Ablaufdiagramm der JS-Funktion <code>auswahl_l</code> . . . . .	86
5.14. Ablaufdiagramm der JS-Funktion <code>auswahl_m</code> . . . . .	87
5.15. Bereich Warnung bei nicht eingeschaltetem Pythonprogramm . . . . .	88
5.16. Realisiertes Anzeigesystem . . . . .	91
6.1. Tabellen der relevanten Schnitte nach Wiedereinlastung eines Unterbaus . . .	95
6.2. Tabellen der relevanten Schnitte nach vier Wiedereinlastungen und einem Ausschuss . . . . .	96
6.3. Tabellen der relevanten Schnitte nach Entnahme der Fertigprodukte . . . . .	97

# Abkürzungsverzeichnis

AC	Alarms and Conditions
Ajax	Asynchronous JavaScript and XML
AVG	Arbeitsvorgänge
AWL	Anwendungsliste
BDE	Betriebsdatenerfassung
CSS	Cascading Stylesheets
DA	Data Access
DataLog	Datenprotokolierung
div	Bereich
el. NOK	elektrisch nicht in Ordnung
el. OK	elektrisch in Ordnung
ERP	Enterprise-Resource-Planning
FUP	Funktionsplan
HA	Historical Access
Handarbeit	Handarbeitsplatz
HAW	Hochschule für Angewandte Wissenschaften Hamburg
HTML	Hypertext Markup Language
I/O	Input/Output
IoT	Internet of Things
JS	JavaScript
MDE	Maschinendatenerfassung
MES	Manufacturing Execution Systems
OPC UA	Open Platform Communications Unified Architecture
opt. NOK	optisch nicht in Ordnung
opt. OK	optisch in Ordnung
PHP	Personal Home Page
Raspberry	Raspberry Pi
RFID	radio-frequency identification

---

Roboter .....	6-Achsroboter
Schnitt .....	Arbeitsabschnitt
Seite .....	Webseite
SP .....	Subprozess
SPS .....	Speicherprogrammierbaren Steuerung
Station .....	Arbeitsstationen
TIA .....	Totally Integrated Automation
Tx .....	Transportband x
WLAN .....	Wireless Local Area Network
WYSIWYG .....	What You See Is What You Get

# 1. Einführung

Die Einführung in die Masterarbeit erfolgt in diesem Kapitel. Zunächst wird eine Übersicht zur aktuellen Situation und der Motivation gegeben. Anschließend werden die Ziele der Arbeit beschrieben und abschließend auf die Gliederung der Masterarbeit eingegangen.

## 1.1. Einführung und Motivation

Um im Wettbewerb mit anderen Firmen mithalten zu können, müssen Produktionsstätten immer flexibler werden. Die Kunden verlangen kurze Lieferzeiten. Das Einlagern produzierter Ware erhöht allerdings den Preis. Um dies zu vermeiden und trotzdem keine langen Lieferzeiten zu erzeugen, müssen die Firmen schnell auf Aufträge reagieren können und die vereinbarten Termine einhalten. Hierfür wird es immer wichtiger, die Fertigungsebene, in der die Steuerung der Anlage überwacht wird, mit der Unternehmensleitebene zu verknüpfen. Durch die Verknüpfung können betriebliche Daten, wie die Durchlaufzeit eines Auftrages oder der Bestand in der Anlage, bzw. in verschiedenen Abschnitten der Anlage, aus den Betriebs- und den Maschinendaten erfasst und ausgewertet werden. Hier spricht man von Betriebsdatenerfassung (BDE) und Maschinendatenerfassung (MDE). Mit den in der Fertigungsebene erhaltenen Daten kann die Unternehmensebene mit Hilfe von Enterprise-Resource-Planning (ERP) Systemen ihre zur Verfügung stehenden Ressourcen wie z. B. Personal, Kapital und Material besser zuteilen und so die Planung von Aufträgen effizienter vornehmen. Zur Verbindung dieser zwei Ebenen werden Manufacturing Execution Systeme (MES) verwendet. Diese bilden das Bindeglied zwischen der Produktion und der Planung. Durch die Prozessnähe des MES kann eine Kontrolle wie auch ein aktiver Eingriff in die Produktion in Echtzeit gewährleistet werden. [23] [24] [28]

Im Fachbereich Automatisierungstechnik der Hochschule für Angewandte Wissenschaften Hamburg (HAW) wurde von der Firma Köster im Jahr 2016 eine flexible Fertigung (Modellfabrik) installiert. Diese produziert Relais für eine Windkraftanlage, welche in einem Gehäuse mit Oberbau montiert werden.

Die Gesamtansicht der Modellfabrik ist in Abbildung 1.1 zu sehen.



Abbildung 1.1.: Gesamtansicht der Modellfabrik

Die Montage ist steuerungstechnisch in sechs Abschnitte unterteilt. Die Aufgaben der Teilabschnitte sind:

- die Koordination eines Teil- oder des Gesamtsystems
- die Verwaltung des Unterbau-Lagers und der Einlagerung der Fertigprodukte
- die Bestückung der Unterbauten mit den Platinen
- die elektrische Prüfung der Platinen
- das Aufsetzen eines Oberbaus, wenn die elektrische Prüfung in Ordnung war
- die optische Endkontrolle, in der geprüft wird, ob der Oberbau korrekt aufgesetzt wurde

Zurzeit werden die betrieblichen Rückmeldedaten der Modellfabrik nicht betrachtet. Die Relevanz in der Wirtschaft wird, wie oben beschrieben, jedoch immer größer. Aus diesem Grund soll für die Modellfabrik in dieser Arbeit eine Aufnahme und Auswertung der Rückmeldedaten in Anlehnung an ein MES realisiert werden.

## 1.2. Ziele der Arbeit

Ziel dieser Arbeit ist die Realisierung eines Systems, das in Anlehnung an ein MES die Rückmeldedaten der Modellfabrik visuell darstellt. Dafür ist im ersten Schritt zu untersuchen, welche Prozessdaten wie z. B. die Durchlaufzeit oder der Bestand für die Unternehmensleitung von Bedeutung sind. Des Weiteren gilt es zu entscheiden, in welchen Unterteilungen der Anlage diese Betrachtung sinnvoll ist.

Die relevanten Rückmeldedaten sollen anschließend erfasst und mittels geeigneter Kommunikation an einen Server übertragen werden. Hierzu sind verschiedene Konzepte wie z. B. die Übertragung mittels vorhandenem Bussystems oder WLAN zu betrachten und umzusetzen.

Abschließend sind die an den Server übertragenen Rückmeldedaten auszuwerten, zu speichern und zu visualisieren. Hierfür sind verschiedene Konzepte zu betrachten und sich für das am besten geeignete zu entscheiden. Eine Mindestanzeige der dargestellten Betriebsdaten besteht aus der Darstellung der betrieblichen Rückmeldedaten in einem Durchlaufdiagramm. Zur Darstellung ist ein geeigneter Zeitraum zu wählen. Außerdem ist die Auflistung der durchschnittlichen Durchlaufzeit, des durchschnittlichen Bestandes und der Anzahl von gefertigten Produkten und Fehlteilen, z.B. in einer Tabelle, erforderlich. Für die Visualisierung ist zu beachten, dass diese übersichtlich dargestellt und von möglichst vielen Endgeräten nutzbar sein soll.

## 1.3. Gliederung der Arbeit

Diese Arbeit unterteilt sich in 7 Kapitel. Nachdem in diesem Kapitel eine Einführung in die Arbeit beschrieben ist, werden in Kapitel 2 die benötigten Grundlagen vorgestellt. Hier wird der produktionslogistische Hintergrund beschrieben, bevor das Kommunikationsmodell OPC UA näher erläutert wird und die verwendeten Scriptsprachen und Entwicklungstools genannt werden. Nachdem die Grundlagen erläutert sind, wird in Kapitel 3 ein Überblick über die Modellfabrik gegeben. Die Vorstellung verschiedener Konzepte zur Umsetzung wird in Kapitel 4 vorgenommen. Außerdem wird sich dort für ein Konzept entschieden. Die Realisierung dieses Konzeptes ist in Kapitel 5 beschrieben. Kapitel 6 umfasst die Überprüfung der Anforderungen an das realisierte System, bevor dieses in verschiedenen Testszenarien auf Funktion geprüft wird. Das Kapitel 7 bietet eine Zusammenfassung dieser Arbeit sowie einen Ausblick.

## 2. Grundlagen

Dieses Kapitel stellt die benötigten Grundlagen vor. Dafür wird zunächst auf die produktionslogistischen Hintergründe eingegangen. Anschließend wird das Kommunikationsprotokoll OPC UA beschrieben, bevor zuletzt die verwendeten Entwicklungstools und Programmiersprachen eingefügt werden.

### 2.1. Produktionslogistischer Hintergrund

Dieses Unterkapitel beschreibt zunächst die Funktion von Manufacturing Execution Systems (MES). Anschließend wird auf die Betriebsdatenerfassung eingegangen und zum Schluss die Theorie logistischer Kennlinien zur Visualisierung von Rückmeldedaten nach Wiendahl erläutert.

#### 2.1.1. Manufacturing Execution Systems

Durch die steigenden Anforderungen der heutigen Kunden, nach schnelleren Lieferterminen bei hoher Qualität und sinkenden Preisen der Produkte, wird der Druck auf Produktionsunternehmen in den letzten Jahren immer weiter erhöht. Es wird nach einer Lösung gesucht, die die Unternehmensleitebene und die Fertigungsebene verbindet. Die Unternehmensebene nutzt Systeme wie Enterprise-Resource-Planning (ERP) und arbeitet in einem mittel- bis langfristigen Zeithorizont (Wochen bis Monate). Sie plant die Aufträge sowie die Ressourcen des Unternehmens und sagt Liefertermine zu. In der Fertigungsebene werden die Produkte hergestellt. Der Zeithorizont ist hier zeitnah oder sogar online (Sekunden bis Minuten). Durch die unterschiedlichen Zeithorizonte ist es für diese Ebenen nicht möglich, direkt aufeinander Einfluss zuzunehmen. Durch die Verbindung der Ebenen und damit einer Verbindung der Zeithorizonte, könnte flexibler auf sich ändernde Umstände, wie z.B. einen neuen wichtigen Auftrag, eine Abweichung des Liefertermins oder Probleme während der Produktion, reagiert werden. Diese Aufgabe der Verbindung übernehmen Manufacturing Execution Systems (MES). Sie bilden die Verbindungsebene zwischen ERP-Systemen und der Fertigung und stellen jeder Ebene die benötigten Daten rechtzeitig, also zum richtigen Zeitpunkt, zur Verfügung. MES weisen einen zeitnahen bis mittelfristigen Zeithorizont auf. [24] [25]

Eine Übersicht der unterschiedlichen Ebenen und deren Zeithorizonte ist in Abbildung 2.1 gezeigt.

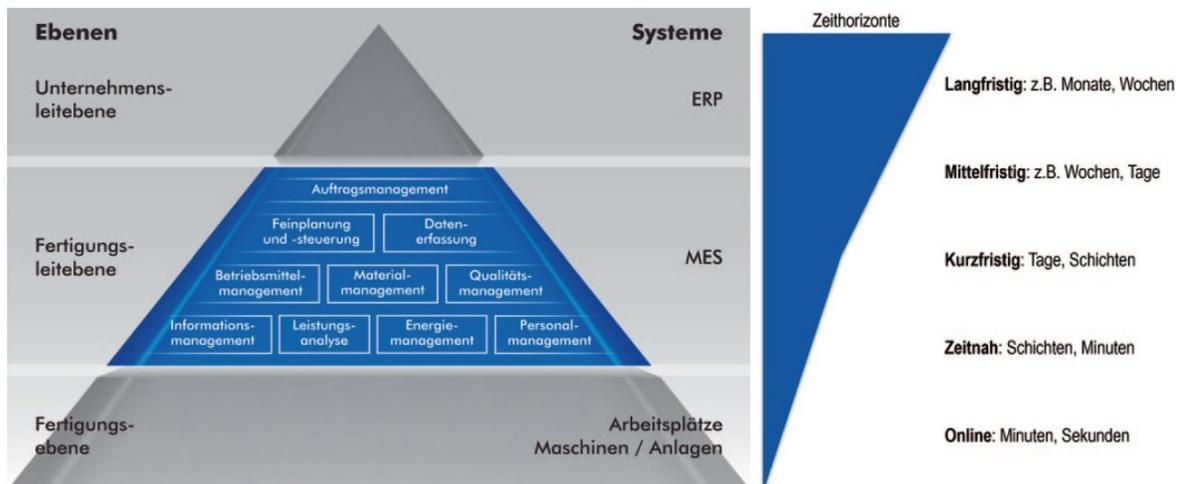


Abbildung 2.1.: Aufgaben eines MES und deren Einordnung in die Unternehmensebenen mit Zeithorizonten (nach [24])

Des Weiteren zeigt die Abbildung 2.1 die nach der VDI 5600 Richtlinie beschriebenen Aufgaben eines MES. Laut dieser Richtlinie übernehmen MES diese Aufgaben komplett oder auch Teile davon, je nach Anforderung des Kunden. [24]

In dieser Arbeit wird die Aufgabe der Datenerfassung von der Produktionsebene in das MES als Hauptaufgabe des zu entwickelnden Systems angesehen. Die Erfassung betrieblicher Rückmeldedaten wird nachfolgend genauer beschrieben.

### 2.1.2. Erfassung von betrieblichen Rückmeldedaten

Wichtige Daten eines MES sind z.B. Zählimpulse, Betriebssignale, Maschinenstatus und Messwerte. Mit diesen können Rückschlüsse auf betriebswirtschaftliche Größen wie z.B. produzierte Gutteile und Ausschüsse sowie verbrauchtes Material und Durchlaufzeiten erfolgen. Die Erfassung dieser Daten ermöglicht eine hohe Transparenz der Produktion. Somit ist es zum einen möglich, Störungen direkt zu erkennen und sie zu beheben, zum anderen Informationen über die Produktion zur Auftragsplanung zu verwenden. Erfasst werden die erforderlichen Maschinendaten entweder durch Sensoren oder die direkte Anbindung an die Steuerung der Maschine. [24] Im Verlauf der Arbeit werden die Signale einer Steuerung, also die Ereignisse in den Schrittketten, als Events bezeichnet.

### 2.1.3. Visualisierung betrieblicher Rückmeldedaten

Dieses Unterkapitel beschreibt die Grundlagen zur Visualisierung von Rückmeldedaten mittels Trichtermodell und dem daraus abgeleiteten Durchlaufdiagramm in Anlehnung an [29] und [31]. Hierfür wird zunächst die Durchlaufzeit vorgestellt, die für das Verständnis des Modells und dieser Arbeit relevant ist und anschließend das Modell und die Ableitung auf das Durchlaufdiagramm erläutert.

#### Durchlaufzeit

Eine Voraussetzung zur Nutzung des Trichtermodelles ist die Definition der Durchlaufzeit. Die Fertigung eines Auftrages kann in mehrere Arbeitsvorgänge (AVG) unterteilt werden. Nach der Fertigstellung des Auftrages an einem Arbeitsvorgang wird der Auftrag nach einer eventuellen Wartezeit (Ausgangslager) zum nächsten Arbeitsvorgang transportiert. Nach einer eventuellen Wartezeit, die sich durch vorausgehende Aufträge ergibt (Eingangslager), wird der Auftrag bearbeitet. Die Bearbeitung des Auftrages kann sich unterteilen in Umrüstung des Arbeitsvorganges und die tatsächliche Bearbeitungszeit des Auftrages. Die Abbildung 2.2 zeigt den beschriebenen Ablauf beispielhaft für einen Arbeitsvorgang innerhalb eines Auftrages. [31]

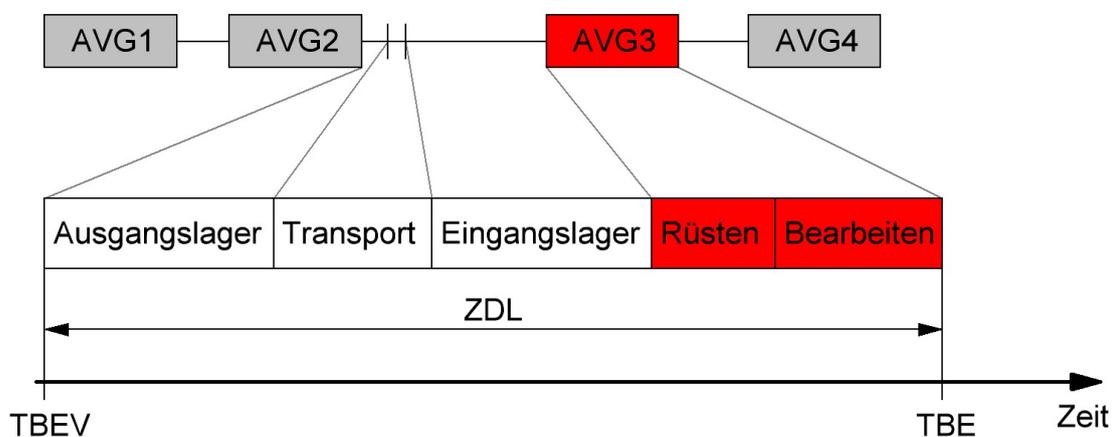


Abbildung 2.2.: Durchlaufzeit (nach [29])

Aus der Abbildung 2.2 wird die Definition der Durchlaufzeit für einen Arbeitsvorgang ersichtlich. Sie umfasst den Zeitraum zwischen der Liegezeit nach dem vorherigen Arbeitsvorgang bzw. dem Einstoßzeitpunkt des Auftrages (TBEV), wenn es sich um den ersten Arbeitsvorgang handelt, bis zur Beendigung des aktuellen Arbeitsvorganges (TBE). [29]

Die Berechnung der Durchlaufzeit ergibt sich nach [29] zu:

$$ZDL = TBE - TBEV \quad (2.1)$$

mit

ZDL	Durchlaufzeit eines Arbeitsvorganges
TBE	Termin Bearbeitungsende eines Arbeitsvorganges
TBEV	Termin Bearbeitungsende des Vorgänger-Arbeitsvorganges

### Trichtermodell

Das Trichtermodell bildet einen Produktionsprozess mit den Größen Zugang, Abgang und Bestand ab. Dafür wird der Prozess in Arbeitsvorgänge unterteilt und jeder Arbeitsvorgang als ein Trichter dargestellt. Kommt ein Auftrag hinzu, so bildet dieser mit den auf ihre Bearbeitung wartenden Aufträgen den Bestand. Der Bestand wird in Vorgabestunden gemessen. Vorgabestunden sind der Arbeitsinhalt eines Auftrages und beschreiben die Zeitspanne, die für die Durchführung des Auftrages vorgesehen ist. Nach der Bearbeitung eines Auftrages verlässt dieser den Trichter durch die untere Öffnung. [31]

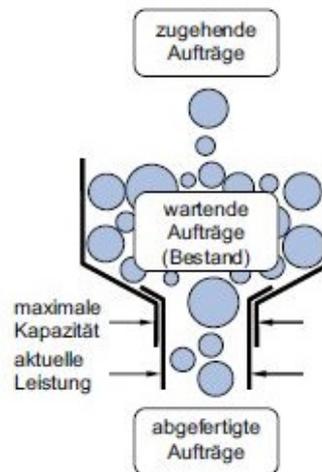


Abbildung 2.3.: Trichtermodell [29]

Die Abbildung 2.3 zeigt einen solchen Trichter. Die Kreise stellen die Aufträge dar. Die Größe des Arbeitsinhaltes eines Auftrages wird anhand der Größe des Kreises dargestellt. Die untere Trichteröffnung visualisiert die Leistung des Arbeitsvorganges. Die aktuelle Leistung kann in den Kapazitätsgrenzen verändert werden. Nimmt die Zahl der zugehenden Aufträge nicht ab, ist aber die maximale Kapazität der Leistung erreicht, so erhöht sich der Bestand. [31]

Zur Abbildung eines Prozesses mithilfe des Trichtermodells wird jeder Arbeitsvorgang als Trichter dargestellt und als Netz miteinander verbunden. Der Materialfluss zwischen den Trichtern entspricht den Aufträgen und den Gegebenheiten des Prozessablaufes. Durch dieses Modell wird der Prozessablauf visuell dargestellt. Zusätzlich bietet es die Möglichkeit, Arbeitssysteme ihrem Aufbau nach zu verdichten. Dadurch werden Kennzahlen und Durchlaufdiagramme für beliebige Detailtiefen erhältlich. Die Verbindung der Trichter untereinander können wie die Arbeitsvorgänge parallel, in Reihe oder mittels Rückkopplung miteinander verbunden sein. Die Abbildung 2.4 bildet die drei Arten zur Vernetzung der Trichter ab. [31]

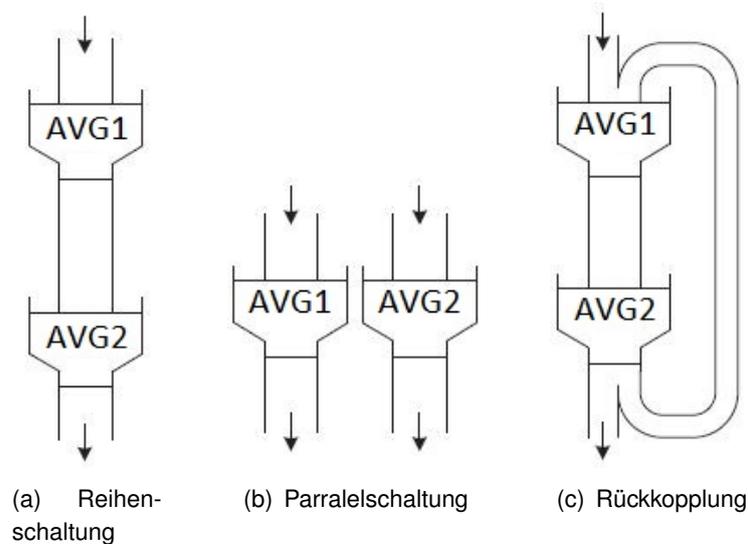


Abbildung 2.4.: Verschiedene Arten der Vernetzung von Trichtern (nach [31])

### Durchlaufdiagramm

Betrachtet man die Änderung im Trichtermodell über einen längeren Zeitraum, so kann das Ergebnis als Durchlaufdiagramm dargestellt werden. Ein beispielhaftes Durchlaufdiagramm ist in der Abbildung 2.5 gezeigt.

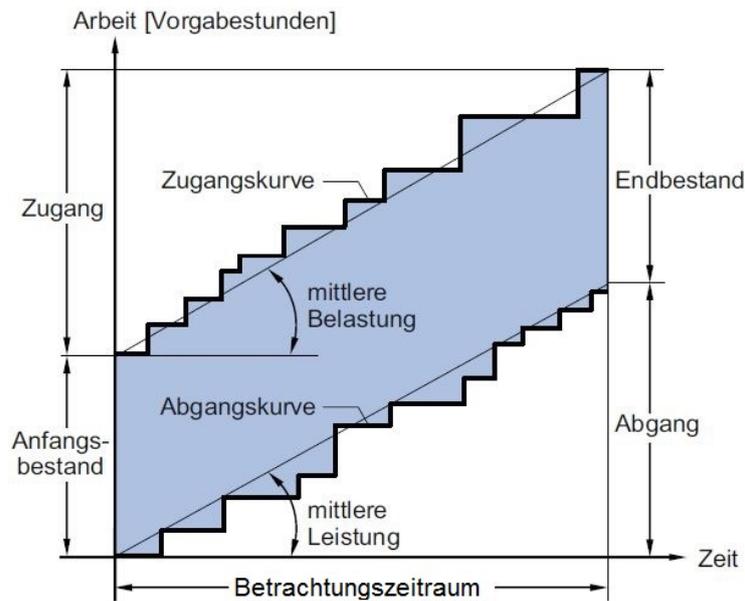


Abbildung 2.5.: Durchlaufdiagramm (nach [29])

Die fertiggestellten Aufträge werden mit ihrem Arbeitsinhalt in Vorgabestunden über den betrachteten Zeitraum abgebildet. Die daraus resultierende Kurve ist die Abgangskurve. Analog lässt sich die Zugangskurve mit den hinzukommenden Aufträgen und deren Arbeitsinhalten darstellen. Der Schnittpunkt der Zugangskurve mit der y-Achse entspricht dem aktuellen Bestand am Anfang des Betrachtungszeitraumes (Anfangsbestand). Der Endbestand lässt sich am Ende des Betrachtungszeitraumes ablesen und ist die Differenz zwischen dem Zugang und dem Abgang zum Zeitpunkt des Betrachtungsendes. Die mittlere Leistung des Durchlaufdiagramms und damit des Arbeitsvorganges lässt sich durch die mittlere Steigung der Abgangskurve und die mittlere Belastung durch die Steigung der Zugangskurve ermitteln. [29]

Zusätzlich zu den Zu- und Abgangskurven lässt sich die Bestandskurve im Durchlaufdiagramm darstellen. Sie ergibt sich zu jedem Zeitpunkt aus der Subtraktion von Zu- und Abgangskurve. [31] Der mittlere Bestand  $B_m$  ergibt sich nach [31] aus der Fläche zwischen der Zu- und Abgangskurve ( $FB$ ) dividiert durch den Betrachtungszeitraum ( $T$ ):

$$B_m = \frac{FB}{T} \quad (2.2)$$

Das Durchlaufdiagramm mit der eingezeichneten Bestandskurve ist in Abbildung 2.6 dargestellt.

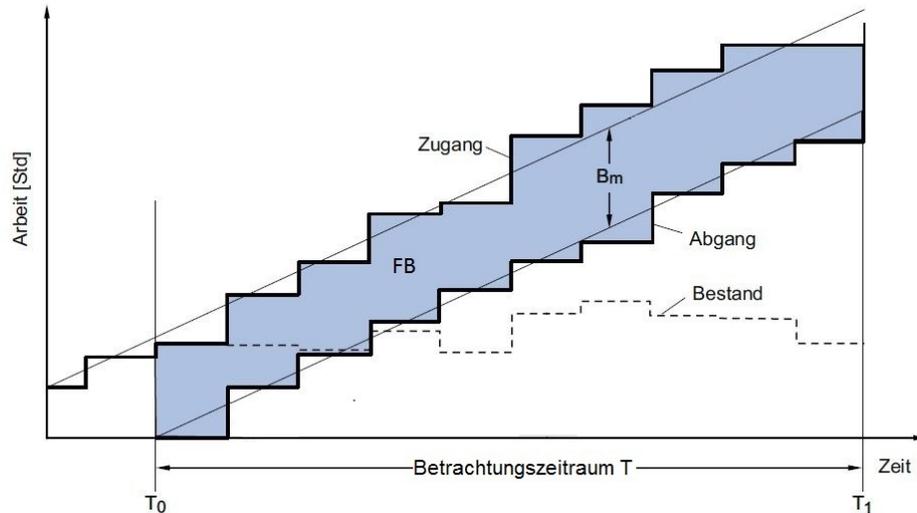


Abbildung 2.6.: Durchlaufdiagramm mit Bestandskurve (nach [29])

Ein Durchlaufdiagramm kann drei verschiedene Betriebszustände eines Arbeitsvorgangs beschreiben. Die verschiedenen Zustände sind in Abbildung 2.7 dargestellt.

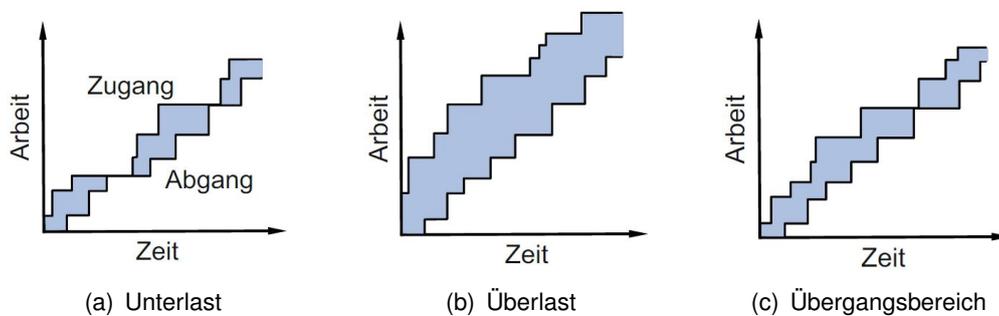


Abbildung 2.7.: Mögliche Betriebszustände eines Arbeitsvorganges (nach [29])

Wird der Bestand eines Arbeitsvorganges reduziert, so sinkt die Durchlaufzeit aufgrund verkürzter Wartezeiten. Sie wird allerdings nie kürzer als die Summe der eventuellen Transportzeit und der Bearbeitungszeit. Wird der Bestand erhöht, so erhöht der Arbeitsvorgang seine Leistung. Dies hat allerdings eine längere Durchlaufzeit zur Folge. Ist die maximale Leistung erreicht, so erhöht sich die Durchlaufzeit weiter und bei weiterer Erhöhung des Bestandes sinkt die Leistung, da es zu einem Blockieren des Arbeitsvorganges kommt. [31]

Wie in Abbildung 2.7 gezeigt, gibt es zum einen die Unterlast (Abbildung 2.7 (a)), hier kommt es durch fehlende Aufträge zu Leistungseinbußen, zum anderen die Überlast (Abbildung 2.7 (b)), bei der die maximale Leistung des Arbeitsvorganges erreicht ist. Der letzte und erstrebenswerte Zustand ist der Übergangsbereich (Abbildung 2.7 (c)). Hier liegt kontinuierlich Arbeit vor, die maximale Leistung ist allerdings noch nicht erreicht. [31]

## 2.2. OPC UA

In diesem Unterkapitel wird das Kommunikationsprotokoll Open Platform Communications Unified Architecture (OPC UA) beschrieben.

### Einführung

OPC UA ist die Weiterentwicklung und Ergänzung des klassischen OPC (OPC Classic). Es ist ein serviceorientiertes, industrielles Kommunikationsprotokoll, das einen wichtigen Standard für die Industrie 4.0 bietet. OPC UA kann ohne Leistungseinbußen in ein bestehendes Industrial Ethernet-Netzwerk integriert werden und es ist ein uneingeschränkter Parallelbetrieb zu Profinet möglich. Im Gegensatz zum klassischen OPC ist OPC UA nicht weiter nur auf Windowssysteme begrenzt. Dies bietet den Vorteil, dass ein OPC UA Server z.B. in Embedded-Feldgeräten oder speicherprogrammierbaren Steuerungen implementiert werden kann. So ist es möglich, eine direkte Verbindung mit verschiedenen Betriebssystemen wie z.B. embedded Linux herzustellen. Beim klassischen OPC wurde für das Auslesen der Daten mittels einer Nicht-Windows-Plattform ein Windows Rechner als OPC Server benötigt, welcher nun unter OPC UA entfällt. Die Kernaufgabe von OPC UA ist die plattformunabhängige Übertragung von Daten und Informationen sowohl horizontal in der Feldebene als auch vertikal zwischen Feldebene und Betriebsleitebene. Durch OPC UA lassen sich mittels des Server-Client-Prinzips die Steuerungsdaten direkt an MES und ERP-Systeme übertragen, was dazu führt, dass der MES D.A.CH Verband e.V. dieses als Transportschicht für den Datenaustausch vorschlägt. [14] [26] [30]

### Technische Beschreibung

Der Aufbau und die Funktionen von OPC UA lassen sich in einem Schichtenmodell beschreiben (siehe Abbildung 2.8). Die hier dargestellte Architektur beschreibt spezielle Aufgaben zur Integration und Nutzung von OPC UA. [20]

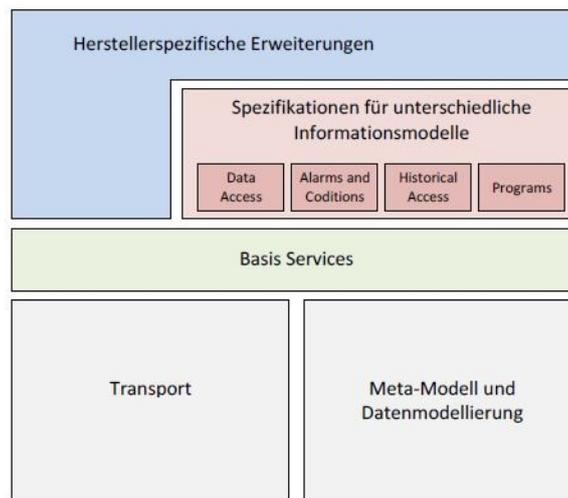


Abbildung 2.8.: Schichtenmodell von OPC UA (nach [20])

Im Folgenden werden die fünf verschiedenen Komponenten, wie der Transport und die Basis Services, aus der Abbildung 2.8 erläutert.

**Transport** Zum Transport der Daten stehen zwei verschiedene Kommunikationswege und eine Kombination aus beiden zur Verfügung. Dabei setzen alle auf das TCP/IP-Protokoll auf. Der erste Kommunikationsweg *Binary* ist optimiert für Geschwindigkeit und Durchsatz. Der zweite Kommunikationsweg über *Webservices* macht sich das bestehende Protokoll HTTP zunutze. Diese hat den Vorteil, dass der Aufwand und die Anfälligkeit bezüglich Firewalls reduziert werden. Die Kombination aus den zwei Wegen ist der *Hybrid* und nutzt Protokolle aus beiden zuvor beschriebenen Kommunikationswegen. [20] [30]

**Meta-Modell und Datenmodellierung** Diese Komponente beschreibt die Regeln und Grundbausteine zur Veröffentlichung eines Informationsmodells über OPC UA. [20]

**Basis Services** Jeder OPC UA Server bietet verschiedene Services (Dienste) an, die der Client nutzen kann. Die Dienste werden zusammengefasst in sogenannten Service Sets dargestellt. Es stehen insgesamt neun Basis Service Sets zur Verfügung, von denen hier exemplarisch zwei genannt und beschrieben werden. Für weiterführende Informationen siehe [20].

**View Service Set** beinhaltet die Funktion des Clients, durch den Adressraum oder durch Teile dieses browsen zu können. Der Server kann dem Client die Sicht und damit den Zugriff

auf Knoten einschränken. Der Client kann also die freigegebenen Verbindungen zwischen den Knoten erkunden und nachvollziehen. [20]

Durch das **Attribute Service Set** wird der Zugriff auf die Variablen (Attribute) und damit das Lesen und Schreiben dieser durch den Client ermöglicht. [20]

**Spezifikation für unterschiedliche Informationsmodelle** Das Informationsmodell beschreibt die Zugriffsart vom Client auf die vom Server zur Verfügung gestellten Informationen, wie z.B. die Variablen. Unterschieden werden die Informationsmodelle Data Access (DA), Alarms and Conditions (AC), Historical Access (HA) und Programms. Das Informationsmodell DA beschreibt das Lesen und Schreiben einzelner Variablen im Adressraum. Mit dem Informationsmodell AC kann der Client sich auf Zustände anmelden. Bei einer Änderung wird vom Server ein Event für den Client ausgelöst. Mittels HA kann der Client auf historische Daten aus Datenbanken zugreifen. Mit Programms lassen sich komplexere Aufgaben in Zustandsautomaten darstellen. Beim Wechsel eines Zustandes wird eine Meldung an den Client ausgelöst. [26]

**Herstellerspezifische Erweiterungen** Die oberste Schicht des OPC UA Schichtenmodell beinhaltet die herstellerspezifischen Erweiterungen. Die OPC Foundation bietet damit den Nutzern eine Schnittstelle, um ihre Erweiterungen und nutzerspezifische Anpassungen vorzunehmen. [20]

## 2.3. Scriptsprachen und Entwicklungstools

In diesem Unterkapitel werden die benötigten Grundlagen der Programmiersprachen allgemein vorgestellt. Zunächst wird die Scriptsprache Python beschrieben. Danach wird auf die Struktur einer Webseite und anschließend auf Datenbanken eingegangen. Abschließend werden die verwendeten Entwicklungstools eingeführt.

### 2.3.1. Python

Python ist eine höhere Programmiersprache. Das Ziel der Entwickler war, eine möglichst leicht erlernbare und gut lesbare Programmiersprache zu entwickeln. Es ist eine serverseitige Programmiersprache, die auf dem Server ausgeführt wird. [19] In dieser Arbeit wird Python 3.5 verwendet. Wird im Folgenden von Python gesprochen, so ist, wenn nicht anders beschrieben, Python 3.5 gemeint.

### 2.3.2. Webseiten

Eine Webseite (Seite) lässt sich in drei Schichten (Abbildung 2.9) beschreiben. Die erste, innerste Schicht und damit der Kern in dem Konstrukt Webseite, ist die Struktur, die durch HTML5 vorgegeben wird. **HTML** steht für Hypertext Markup Language und beinhaltet den Inhalt, wie z.B. das geschriebene Wort und Bilder. Die Inhalte können in verschiedenen Bereichen (divs) gespeichert werden. Die zweite Schicht beschreibt die Designschicht mittels **CSS** (Cascading Stylesheets). Durch CSS wird der Inhalt des HTML-Dokumentes, wie z.B. die divs gestaltet. Hier werden Stilanweisungen für die Webseite festgelegt und die Positionierung der in HTML erstellten divs vorgenommen. Die dritte (optionale) Schicht ist eine Scriptsprache, die das Verhalten der Webseite beeinflusst. In dieser Arbeit wird für diese Aufgabe **JavaScript** (JS) gewählt. Durch die Nutzung von JS wird das Verhalten der Seite gesteuert. Somit werden Interaktionen des Nutzers, wie das Klicken eines Buttons, ermöglicht. Es ermöglicht die Darstellung verschiedener Inhalte und eine Aktualisierung der Daten einer Seite, ohne dass diese neu geladen wird. [19] [27] [35]



Abbildung 2.9.: Schichten einer Webseite (nach [27])

Alle beschriebenen Ebenen werden clientseitig, sprich im Webbrowser ausgeführt. Beispielsweise zur Nutzung von Datenbanken wird die verbreitete serverseitige Scriptsprache **PHP** 7.1 (Personal Home Page) genutzt. Der Aufruf eines PHP-Scripts kann clientseitig durch einen JS-Aufruf mittels Ajax erfolgen. **Ajax** steht für Asynchronous JavaScript and XML und ermöglicht es, auch ohne das Klicken eines Buttons asynchron mit dem Webserver zu kommunizieren. Die Nutzung von Ajax kann über ein Framework erfolgen. **Frameworks** unterstützen den Programmierer und erleichtern die Verwendung verschiedener Funktionen durch Bibliotheken. Durch Ajax kann der Inhalt einer Webseite durch Abfrage und Empfangen serverseitiger Daten verändert werden, ohne dass der Nutzer die Kommunikation aktiv auslöst. Durch Ajax wird serverseitig ein PHP-Interpreter aktiviert, der das angeforderte PHP Script ausführt. [19] [27] [35]

Hierzu ein Beispiel: Auf einer Webseite soll der aktuelle Kontostand dargestellt werden. Zusätzlich zum Inhalt und dem Design der Webseite wird eine dauerhafte Aktualisierung des Kontostandes benötigt. Zur Umsetzung wird in JS eine Funktion geschrieben, die sich sekundlich wieder aufruft und mittels Ajax ein PHP-Script öffnet, das die Datenbank öffnet, den Kontostand ausliest und anschließend den ausgelesenen Wert an JS zurück gibt. Die Funktion schreibt den empfangenen Kontostand in die HTML-Datei. Der Ablauf ist in Abbildung 2.10 graphisch dargestellt.

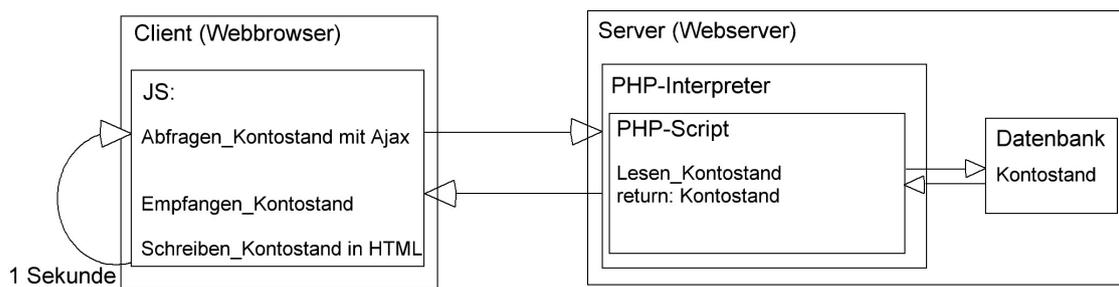


Abbildung 2.10.: Beispiel einer Datenbankanfrage mittels Ajax und PHP

### 2.3.3. Datenbanken

Die Einführung in Datenbanken ist an [32] angelehnt.

Datenbanken speichern große Datensätze zur übersichtlichen Darstellung. Eine Datenbank kann aus mehreren Tabellen bestehen. In einer Tabelle werden die Datenfelder (Namen der Daten) in der ersten Zeile gespeichert. Darunter befinden sich die Datensätze. Eine Datenbank wird in drei Schritten erzeugt:

1. Datenbank anlegen
2. Tabellen in der Datenbank erstellen und eine Struktur (Datenfelder) vorgeben
3. Datensätze in die Tabellen eingeben.

**MySQL** Zur Nutzung einer Datenbank mittels PHP ist die Datenbank *MySQL* weit verbreitet. Diese ist Open-Source und nutzt die Datenbanksprache SQL. Mittels SQL können Tabellen erzeugt, Datensätze gelesen und geschrieben sowie gelöscht werden.

**phpMyAdmin** Die Bedienoberfläche phpMyAdmin bietet eine visuelle Darstellung der Datenbank mit deren Tabellen und Datensätzen. Hier können Datenbanken angelegt und verwaltet werden. Außerdem können die Tabellen und Datensätze bearbeitet werden.

### 2.3.4. Entwicklungstools

**Totally Integrated Automation (TIA) Portal** Das TIA Portal der Firma Siemens wird in dieser Arbeit in der Version 14 genutzt. Es bietet eine Arbeitsumgebung zur Konfiguration und Programmierung der S7-1500 Steuerungen von Siemens. Als Programmiersprachen werden unter anderem Anwendungsliste (AWL) und Funktionsplan (FUP) verwendet. Den Einstieg in diese Arbeitsumgebung bietet das Dokument [15].

**Thonny** Thonny ist eine Entwicklungsumgebung, in der Pythonprogramme programmiert werden können und ist in der Version 2.1.16 auf dem Raspberry vorinstalliert. Der Vorteil besteht darin, dass das Pythonprogramm direkt in der Entwicklungsumgebung ausgeführt und gedebuggt werden kann. Zusätzlich steht ein Ausgabefenster (Shell) zur Verfügung. Sämtliche in dieser Arbeit entwickelten Pythonprogramme wurden in dieser Umgebung entwickelt. [12]

**Microsoft Expression Web 4** Microsoft Expression Web V4.0.1460.0 ist eine Umgebung zur Entwicklung von Webseiten. Es bietet zum einen eine WYSIWYG (What You See Is What You Get) Funktion und zum anderen einen Quelltext Editor, der unter anderen Sprachen wie HTML und PHP unterstützt. Durch die WYSIWYG-Funktion wird das Tool in dieser Arbeit auf einem Windows Computer genutzt, um sich in die Webentwicklung einzuarbeiten. [7]

**Geany** Geany ist eine Entwicklungsumgebung für verschiedene Programmiersprachen wie z.B. HTML und PHP. Es ist auf dem Raspberry in der Version 1.29 vorinstalliert und wird in dieser Arbeit zur Entwicklung und Bearbeitung von HTML-, CSS-, JS- und PHP-Dateien auf dem Raspberry genutzt. [3]

**Microsoft PowerPoint** In dieser Arbeit wird PowerPoint 2016 aus dem Office 365 Paket für die Konzeption der Webseite genutzt. Mit PowerPoint lassen sich Bildschirmpräsentationen erstellen. In den Einstellungen der Bildschirmpräsentation lässt sich die Art der Präsentation auf `Ansicht an einem Kiosk` auswählen. Dies bewirkt, dass die Folien nach Start der Präsentation nicht mehr verändert werden können. Werden Hyperlinks auf den Folien eingefügt, so ist es nur mit diesen möglich, sich in der Präsentation zu bewegen. Diese Einstellung wird bei den in dieser Arbeit erstellten Bildschirmpräsentationen gewählt. [9]

## 3. Anlagenbeschreibung

In diesem Kapitel wird die Modellfabrik in Anlehnung an [22] und [34] beschrieben. Hierbei wird zunächst die Anlage allgemein und anschließend der Prozessablauf dargestellt. Des Weiteren werden die Arbeitsstationen im Detail beschrieben und schließlich die Kommunikation der einzelnen Komponenten miteinander erläutert. Der Fokus der Beschreibung liegt hierbei nicht auf dem Steuerungsprogramm, sondern auf dem Gesamtprozess. Außerdem werden spezielle Sensorik sowie einzelne Events aus den Schrittketten der bestehenden Steuerungen betrachtet. Dies dient dazu, die Anlage im Verlauf dieser Arbeit in verschiedene Arbeitsabschnitte (Schnitte) zu unterteilen.

### 3.1. Gesamtübersicht

Die Modellfabrik wurde im Jahr 2016 von der Firma KÖSTER Systemtechnik GmbH an die HAW Hamburg ausgeliefert. Sie stellt Relais für eine Windkraftanlage her. Die Fertigprodukte bestehen aus einem Unterbau, der mit einer Platine bestückt wird und einem Oberbau, der auf den Unterbau gepresst wird. Ein Fertigprodukt kann entweder als ein- oder zweipoliges Relais ausgeführt sein. Hierzu wird entweder eine ein- oder zweipolige Platine verwendet. Die einzelnen Bauteile sowie das fertige Produkt sind in Abbildung 3.1 zu sehen.



Abbildung 3.1.: Bauteile und Fertigprodukt, von links nach rechts:  
Unterbau, Platine (hier einpolig), Oberbau, Fertigprodukt

Eine schematische Darstellung der Anlage ist in Abbildung 3.2 zu sehen. Um die Übersicht zu wahren, werden in der Darstellung nur für diese Arbeit relevante Komponenten gezeigt.

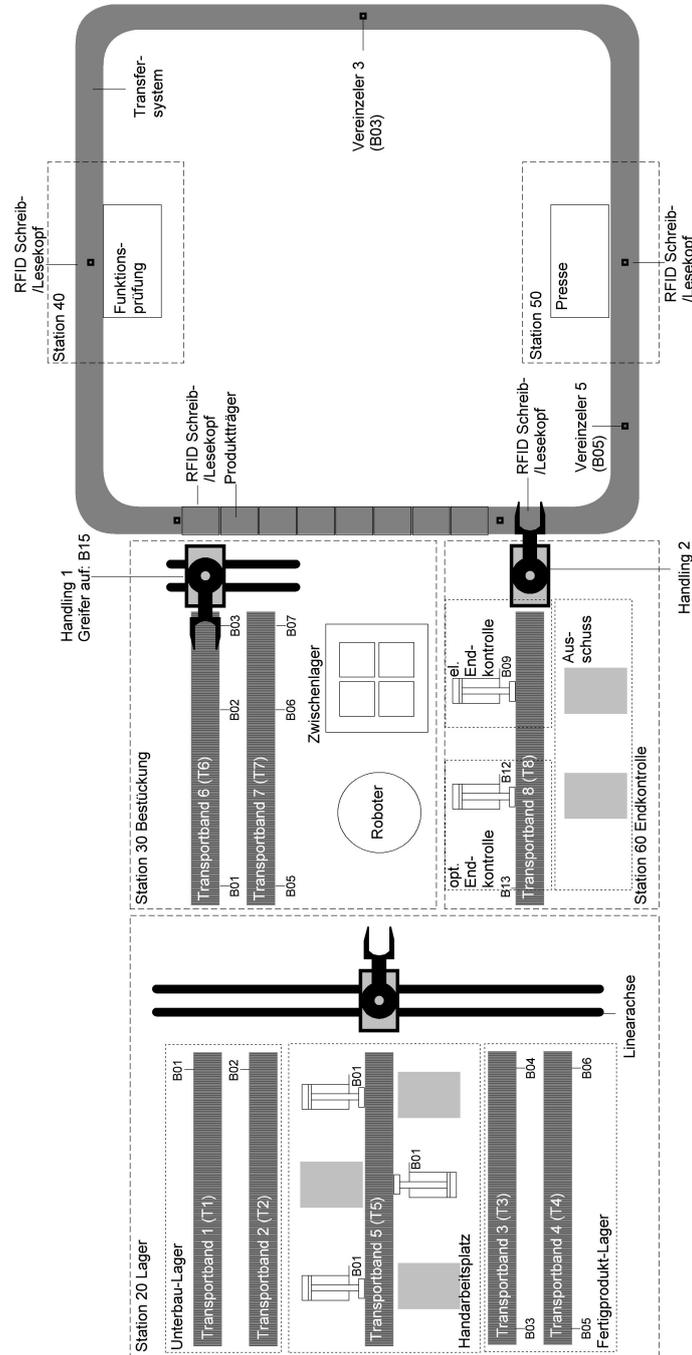


Abbildung 3.2.: Schematische Darstellung der Gesamtanlage

Die Anlage ist in sechs autarke Arbeitsstationen (Station) unterteilt. Jede Station ist mit einer Speicherprogrammierbaren Steuerung (SPS), welche den Arbeitsprozess steuert, ausgestattet. Die verwendete Steuerung an allen Stationen ist die Siemens S7 1512C-1PN, mit der Firmware 1.8.

Die Station 20 beinhaltet die Auftragseingabe, die Lagerverwaltung und das Achsenhandling. Die Station 30 umfasst die Bestückung der Unterbauten mit einer ein- oder zweipoligen Platine durch einen 6-Achsroboter (Roboter) der Firma Denso. An Station 40 findet die elektrische Funktionsprüfung der Platine statt. Die Station 50 besteht aus der Pressung der Oberbauten. Zu der Station 60 gehört die Endkontrolle der Produkte. Das Transfersystem wird durch ein übergeordnetes I/O-System gesteuert. Die Produktträger auf dem Transfersystem sind mit einem radio-frequency identification (RFID)-Tag versehen. Auf diesem werden die Zustände *Produkt ein- oder zweipolig* und *Platine elektrisch in Ordnung (el. OK)* oder *elektrisch nicht in Ordnung (el. NOK)* für das sich darauf befindende Produkt gespeichert. Zum Lesen und Schreiben der RFID-Tags stehen an relevanten Positionen RFID Schreib- und Leseköpfe zur Verfügung. Die Station 10 ist der Leitstand und übernimmt die Aufgabe der Kommunikation der Stationen untereinander sowie mit dem übergeordneten I/O-System und den RFID Schreib-/Leseköpfen.

## 3.2. Prozessablauf

Hier wird der Ablauf der Produktion anhand eines Produktes beschrieben und anschließend der Prozess in Abbildung 3.4 als Trichtermodell dargestellt. Die Arbeitsvorgänge eines ein- und zweipoligen Produktes unterscheiden sich nur an der Station 30. Aus diesem Grund wird der Prozess allgemein beschrieben und nur an relevanten Stellen auf die unterschiedliche Behandlung von ein- und zweipoligen Produkten hingewiesen.

Durch ein Touchpanel kann die Anzahl der zu fertigenden Produkte eingegeben werden. Jedes Produkt stellt einen eigenen Auftrag dar. Sollen also z.B. zwei einpolige Produkte gefertigt werden, so werden diese im Folgenden als zwei voneinander unabhängige Aufträge betrachtet. Die Unterbauten werden aus dem Unterbau-Lager (Station 20) entnommen und durch die Linearachse der Station 30 zugeführt. Für einpolige Produkte werden die Unterbauten von Transportband 1 (T1), für zweipolige Produkte Transportband 2 (T2) entnommen und dem Transportband 6 (T6) (einpolig) oder dem Transportband 7 (T7) (zweipolig) zugeführt. Dort werden sie mit einer Platine bestückt und dann durch Handling 1 auf einem Produktträger des Transfersystems abgelegt. Auf den RFID-Tag, der sich am Produktträger befindet, wird die Information geschrieben, ob es sich um ein ein- oder zweipoliges Zwischenprodukt handelt. Über das Transfersystem wird das Zwischenprodukt zunächst zur Funktionsprüfung (Station 40) transportiert. Hier wird die Platine auf ihre elektrische Funktion geprüft. Das Ergebnis der Prüfung wird auf den RFID-Tag des Produktträgers geschrieben.

Hat das Produkt die Funktionsprüfung nicht bestanden, ist die Platine also defekt (*el. NOK*), so wird automatisch ein neues Produkt in Auftrag gegeben. Anschließend wird an der Station 50 bei bestandener Funktionsprüfung (*el. OK*) ein Oberbau auf den Unterbau gepresst. Aus der Station 50 kommt so ein Endprodukt, bestehend aus dem Unterbau, einer Platine und dem Oberbau heraus. War die Funktionsprüfung nicht in Ordnung (*el. NOK*), wird kein Oberbau auf das Zwischenprodukt gepresst, sondern dieses in der Station 50 durchgeschleust. In diesem Fall verlässt das Produkt die Station 50 als Zwischenprodukt bestehend aus Unterbau und Platine. An der Endkontrolle (Station 60) wird der RFID-Tag des Produktträgers ausgelesen und das Produkt dem Transfersystem durch das Handling 2 entnommen. Ist das Produkt *el. NOK*, so wird entweder die Platine demontiert und der Unterbau wieder in den Prozess eingelastet, oder der Unterbau wird samt Platine ausgeschoben. Wiedereinlastung bedeutet hier, der Unterbau wird dem Prozess erneut zugeführt. Dazu steht an der Station 30 zusätzlich zu T6 und T7, auf denen die Unterbauten bestückt werden, ein Zwischenlager bereit, in dem vier Unterbauten gelagert werden können. Sind bei Wiedereinlastung noch nicht alle Unterbauten bestückt und wird somit noch ein Unterbau benötigt, so wird der Unterbau direkt auf T6 oder T7 abgelegt. Anderenfalls wird dieser wenn möglich gelagert. Wenn die Kapazität des Zwischenlagers ausgeschöpft ist, ist ein Lagern nicht mehr möglich. Soll kein Unterbau mehr bestückt werden und eine Lagerung ist nicht möglich, wird das Zwischenprodukt ausgeschoben. Die ausgeschobenen Produkte werden nachfolgend als elektrischer Ausschuss betrachtet, da der Unterbau nicht mehr verwendet wird, obwohl dieser nicht beschädigt ist. Das Ablaufdiagramm zur Wiedereinlastung ist in Abbildung 3.3 gezeigt.

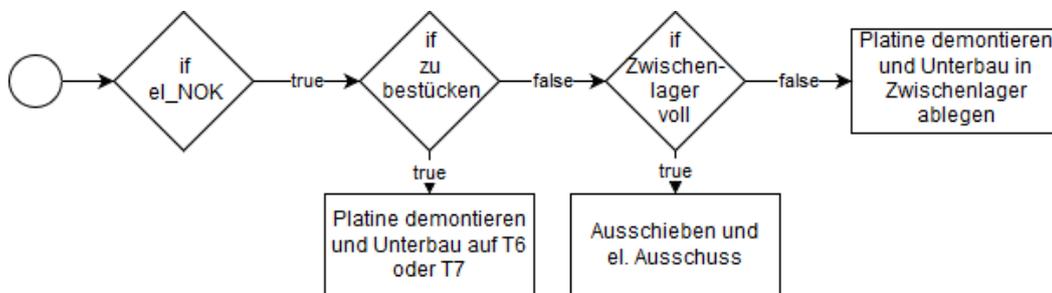


Abbildung 3.3.: Ablaufdiagramm bei Wiedereinlastung

Ist das Produkt ein Endprodukt, wird es in der Station 60 einer optischen Endkontrolle unterzogen. Ist der Oberbau korrekt aufgepresst, wie in Abbildung 3.1 gezeigt, wird dieses als *Endprodukt-opt-OK* in das Fertigprodukt-Lager transportiert. Bei der Ankunft im Fertigprodukt-Lager wird es als *Fertigprodukt-gut* angesehen. Nicht korrekt aufgepresste Oberbauten werden als *Endprodukt-opt-NOK* entweder in den Handarbeitsplatz transportiert, oder, wenn der Puffer des erkannten Fehlers am Handarbeitsplatz voll ist, ausgeschoben. Wird das *Endprodukt-opt-NOK* ausgeschoben, wird es nachfolgend als optischer Ausschuss betrachtet. Wenn es zum Handarbeitsplatz transportiert wird, wird es dort durch

den Mitarbeiter nachbearbeitet und anschließend als *Fertigprodukt-opt-nachbearbeitet* im Fertigprodukt-Lager abgelegt.

Bei Betrachtung des in Abbildung 3.4 gezeigten Trichtermodells wird klar, dass der Prozess seriell abläuft und keine parallele Bearbeitung von Produkten an einer Station zulässt. Ein Überholen der Produkte ist nur durch den Handarbeitsplatz möglich.

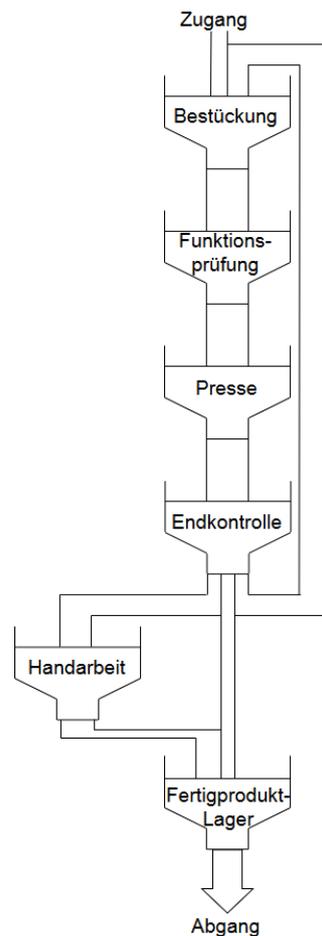


Abbildung 3.4.: Trichtermodell des Gesamtprozesses

### 3.3. Beschreibung der Stationen

Dieses Unterkapitel befasst sich mit der Beschreibung der einzelnen Stationen. Beschrieben werden die für diese Arbeit relevanten Merkmale. Jedem Arbeitsvorgang ist ein Eingangslager vor- und ein Ausgangslager nachgestellt. Die Ein- und Ausgangslager eines Arbeitsvorganges werden in der Beschreibung der Stationen definiert.

**Station 10** Die Station 10 ist der Leitstand der Anlage. Alle Informationen, die die Stationen 20-60 miteinander teilen, werden zunächst an die Station 10 und von dieser an die Empfangsstation geschickt. Eine direkte Kommunikation zwischen den Stationen 20-60 ist nicht möglich. Außerdem kommuniziert die Station 10 mit den RFID Lese- und Schreibköpfen sowie mit dem übergeordneten I/O-System, das für die Steuerung des Transfersystems zuständig ist.

Der Ausschnitt des Transfersystems aus Abbildung 3.2 ist in Abbildung 3.5 noch mal vergrößert dargestellt.

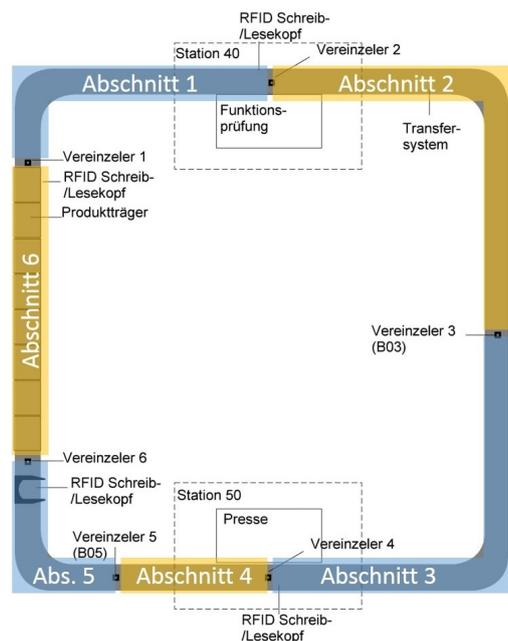


Abbildung 3.5.: Schematische Darstellung der Station 10

Wie hier zu sehen ist, besitzt das Transfersystem sechs Vereinzeler, welche bei Ansteuerung jeweils einen Produktträger passieren lassen und das Transportband so in sechs Abschnitte unterteilen. Abschnitt 1 ist der Transportweg zur Funktionsprüfung (Station 40) und damit dessen Eingangslager. Abschnitt 2 liegt zwischen der Funktionsprüfung und dem Vereinzeler 3 und kann so als Ausgangslager betrachtet werden. Abschnitt 3 ist der Transportweg zwischen dem Vereinzeler 3 und der Presse (Station 50) und damit deren Eingangslager. Abschnitt 4 liegt zwischen der Presse und dem Vereinzeler 5 und kann als Ausgangslager angesehen werden. Der Abschnitt 5 zwischen dem Vereinzeler 5 und dem Handling 2 ist der Transportweg zur Endkontrolle, weshalb dieser als Eingangslager betrachtet werden kann. Der Abschnitt 6 ist das Pufferlager für die Produktträger und wird nicht weiter betrachtet.

**Station 20** Die Station 20 umfasst die Lagerverwaltung sowie die Auftragseingabe und den Transport der Produkte zwischen den Lagern und den Stationen 30 und 60. Der Ausschnitt der Station 20 aus Abbildung 3.2 ist in Abbildung 3.6 noch mal dargestellt.

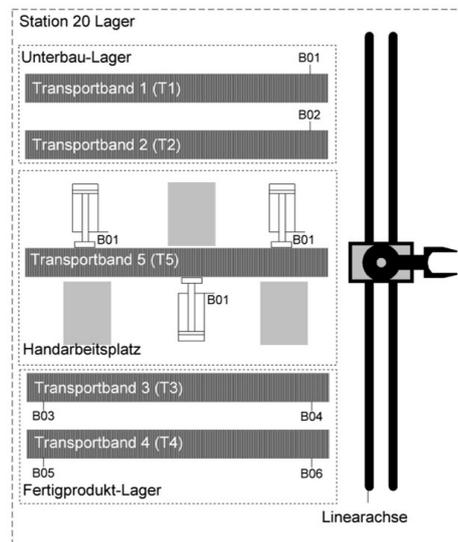


Abbildung 3.6.: Schematische Darstellung der Station 20

Die Lager unterteilen sich in das Unterbau-Lager mit den Transportbändern 1 (T1) und 2 (T2), das Fertigprodukt-Lager mit Transportband 3 und 4 (T3 und T4) und den Handarbeitsplatz mit Transportband 5 (T5). Das Unterbau-Lager ist das Eingangslager der Gesamtanlage. Es wird angenommen, dass hier immer genug Unterbauten vorhanden sind. Verlässt ein Unterbau das Lager (B01 oder B02), so wird er als Produkt in der Anlage und somit als Bestand angesehen. Das Fertigprodukt-Lager beschreibt das Ausgangslager der Anlage. Alle gefertigten Produkte werden hier gelagert. Werden die Fertigprodukte aus dem Lager entnommen (B03 oder B05), sind sie nicht weiter im Bestand der Anlage geführt. Der Handarbeitsplatz kann als Zwischenlager angesehen werden. Hier werden optische Fehlteile bis zur Nachbearbeitung eines Mitarbeiters gelagert. Er beinhaltet drei Puffer für verschiedene Fehlerbilder der Oberbauten mit je einer Kapazität von 2. Die *Endprodukte-opt-NOK* werden durch Zylinder in die entsprechenden Puffer ausgeschoben.

Durch die Eingabe von Aufträgen an einem Touchpanel der Station, kann die Anzahl der zu fertigenden ein- und zweipoligen Produkte gewählt werden. Die Steuerung gibt vor, dass die Unterbauten für einpolige Produkte von T1, die für zweipolige von T2 entnommen werden. Fertigprodukte können entweder direkt von der Endkontrolle oder durch den Mitarbeiter am Handarbeitsplatz in das Fertigprodukt-Lager gelangen. Die Unterscheidung zwischen ein- und zweipoligen Endprodukten findet beim Ausschieben der *Endprodukte-opt-NOK* in den jeweiligen Puffer des Handarbeitsplatzes oder am Eingang des Fertigprodukt-Lagers (B04

oder B06) statt. End- bzw. Fertigprodukte, die im Handarbeitsplatz oder Fertigprodukt-Lager liegen, sind von der Anlage nicht mehr in ihrer Art unterscheidbar.

**Station 30** In Station 30 werden die Unterbauten mit der Platine bestückt. Der Ausschnitt Station 30 aus der Abbildung 3.2 ist in Abbildung 3.7 dargestellt.

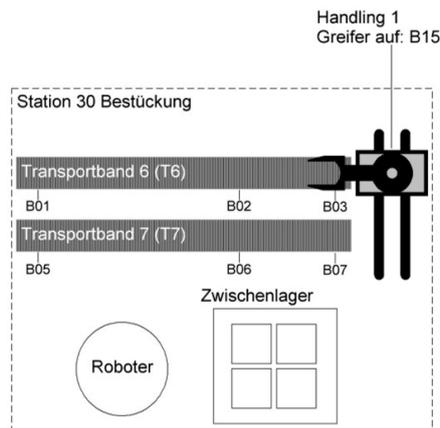


Abbildung 3.7.: Schematische Darstellung der Station 30

Zur Bestückung der Unterbauten stehen zwei Transportbänder zur Verfügung. Produkte, die mit einer einpoligen Platine bestückt werden, werden auf T6 und die, die mit einer zweipoligen Platine bestückt werden, auf T7 abgelegt. Auf jedem Transportband kann sich nur ein Unterbau zur Zeit befinden. Erst wenn das Handling 1 das Zwischenprodukt auf dem Produktträger abgelegt hat und sich wieder in Grundstellung befindet, lässt die Steuerung ein neues Produkt auf dem Transportband zu. T6 und T7 haben somit jeweils eine Kapazität von 1. Die Transportbänder lassen sich jeweils in ein Eingangslager, den Fertigungsprozess Bestückung und ein Ausgangslager unterteilen. Das Eingangslager der Transportbänder T6/T7 liegt zwischen B01/B05 und B02/B06. Die Bestückung findet an B02/B06 statt. Das Ausgangslager ist der Bereich zwischen B02/B06 und dem Produktträger am Handling 1. Das Produkt verlässt das Ausgangslager, wenn das Handling 1 nach dem Ablegen wieder auf seiner Ausgangsposition steht (fallende Flanke von dem Event `HAND_W`). Zusätzlich zu den zwei Transportbändern beinhaltet die Station ein Zwischenlager mit einer Kapazität von 4, in dem die Unterbauten gelagert werden, welche aus der Endkontrolle wieder eingelastet wurden und nicht gleich bestückt werden.

Ebenso wird der Roboter durch die Station 30 gesteuert. Dieser bestückt zum einen die Unterbauten in Station 30. Zum anderen lastet er Unterbauten der *el. NOK* Zwischenprodukte aus der Station 60 wieder ein. Das bedeutet, zusätzlich zur Bestückung der Unterbauten in Station 30 demontiert der Roboter die *el. NOK* Platine und transportiert den Unterbau entweder zu T6 oder T7 oder lagert ihn im Zwischenlager ein. Ist ein Unterbau zu bestücken und es befindet sich einer im Zwischenlager, transportiert der Roboter diesen auf T6 bzw. T7.

**Station 40** Station 40 steuert die Funktionsprüfung der Platinen. Hier wird zunächst vom RFID-Tag gelesen, ob es sich um ein ein- oder zweipoliges Produkt handelt. Anschließend wird die entsprechende Funktionsprüfung durchgeführt. Das Ergebnis der Prüfung wird auf den RFID-Tag gespeichert.

**Station 50** Station 50 presst bei *el. OK* Produkten einen Oberbau mit HAW Hamburg Logo auf das Zwischenprodukt. Ist das Produkt *el. NOK*, so wird dieses ohne Oberbau durchgeschleust.

**Station 60** Station 60 beinhaltet die Endkontrolle. Der Ausschnitt Station 60 aus der Abbildung 3.2 wird in Abbildung 3.8 noch mal gesondert betrachtet.

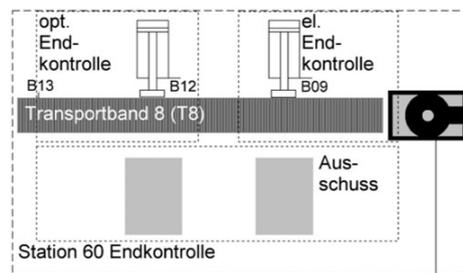


Abbildung 3.8.: Schematische Darstellung der Station 60

Erreicht ein Produkt die Station 60 als Zwischenprodukt, hat es die Funktionsprüfung nicht bestanden. Es wurde kein Oberbau aufgepresst und das Zwischenprodukt wird, wenn möglich, demontiert und wieder eingelastet. Muss kein Unterbau bestückt werden und ist auch kein Platz mehr im Zwischenlager, so wird das Zwischenprodukt ausgeschoben und als elektrischer Ausschuss (*el. Ausschuss*) gewertet. Ist das Produkt ein Endprodukt, so wird es zur optischen Endkontrolle transportiert. Der Vision Sensor zur optischen Kontrolle erkennt drei verschiedene Fehlerfälle:

- Fehler A: Das Logo auf dem Oberbau fehlt
- Fehler B: Der Oberbau und das Logo sind um  $180^\circ$  verdreht
- Fehler C: Der Oberbau und das Logo sind um  $\pm 90^\circ$  verdreht

Erkennt der Vision Sensor keinen der drei Fehler, hat das Endprodukt die optische Endkontrolle bestanden (*opt. OK*) und wird als Gutteil ins Fertigprodukt-Lager transportiert. Besteht das Endprodukt die optische Kontrolle nicht (*opt. NOK*), so wird das Endprodukt entweder zum Handarbeitsplatz transportiert oder ausgeschoben und als optischer Ausschuss (*opt. Ausschuss*) angesehen. Ausschüsse sind Produkte, welche nicht mehr weiter bearbeitet werden können. Sie werden ausgeschoben, da eine Wiedereinlastung oder händische Reparatur aus Kapazitätsmangel zu diesem Zeitpunkt nicht möglich ist. Aus diesem Grund gehören Produkte im Ausschuss nicht mehr zum Bestand der Anlage.

Je nach Art des Produktes lassen sich Ein- bzw. Ausgangslager und Arbeitsvorgänge definieren.

Ist das Produkt ein Zwischenprodukt, so ist das Eingangslager der Transport des Produktes vom Produktträger bis zum ersten Abschieber. Die Demontage bzw. das Ausschieben des Zwischenproduktes wird als Arbeitsvorgang angesehen. Ein Ausgangslager existiert in diesem Fall nicht.

Für Endprodukte beschreibt das Eingangslager den Transport des Endproduktes vom Produktträger zur optischen Prüfung des Produktes (Sensor B10). Wird das Endprodukt zur Station 20 weiter transportiert, so lässt sich der Transport von der optischen Prüfung (Sensor B10) bis zum Ende des T8 (Sensor B13) als Ausgangslager ansehen.

Ausgeschobene Produkte sind der Ausschuss der Produktion und damit ein Abgang der Gesamtanlage und werden als Abgang Station 60 betrachtet.

### 3.4. Aufbau der Kommunikation

Die Kommunikation der Steuerungen erfolgt über Profinet. Die Stationen 20 bis 60 senden ihre Daten an die Station 10. Diese sendet die Informationen dann über virtuelle I/O's an die anderen Stationen. Durch die Vernetzung der Anlage in einem Ring mit Profinet sind alle Informationen der gesamten Anlage von jeder Station abrufbar. Durch eine Verbindung mit einem PC an einer Profinet-Schnittstelle kann also jede SPS im Netzwerk angesprochen werden. Somit ist es möglich, die Informationen einer jeden SPS an einer einzigen Profinet-Schnittstelle auszulesen und zu sammeln.

## 4. Konzeption

Wie in den Zielen der Arbeit (Unterkapitel [1.2](#)) beschrieben, soll ein System zur Beobachtung und Auswertung der betrieblichen Rückmeldedaten der Modellfabrik realisiert werden. Dieses System soll an ein MES angelehnt sein und relevante Daten in Durchlaufdiagrammen und in tabellarischer Form darstellen. Diese Aufgabe lässt sich in die drei Teilaufgaben aufteilen: Graphische Oberfläche erstellen, Daten bereitstellen und verarbeiten sowie die Auswahl zu verwendender Hardware. In den folgenden Unterkapiteln werden für jede Teilaufgabe die geltenden Anforderungen definiert und Konzepte zur Realisierung vorgestellt. Abschließend werden die Teilkonzepte zu einem Gesamtkonzept zusammengefasst.

### 4.1. Graphische Oberfläche zur Visualisierung der Rückmeldedaten

In diesem Unterkapitel werden der Inhalt und das Layout der Oberfläche zur Visualisierung der Daten dargestellt. Hierbei werden zunächst die Anforderungen an die Benutzeroberfläche beschrieben. Anschließend wird die Anlage in verschiedene Abschnitte (Schnitte) aufgeteilt, bevor die Inhalte der Benutzeroberfläche beschrieben werden. Zum Schluss wird das Layout der Benutzeroberfläche konzeptioniert. Sämtliche innerhalb der folgenden Abschnitte dieses Unterkapitels ([4.1](#)) referenzierte Anforderungen beziehen sich auf jene, die in Abschnitt [4.1.1](#) definiert sind.

#### 4.1.1. Anforderungen

Die Benutzeroberfläche soll relevante Rückmeldedaten in Durchlaufdiagrammen und in tabellarischer Form darstellen. Die darzustellenden Daten sind zum einen Live-Daten, um die Produktion zu verfolgen und zum anderen historische Daten, die bis zu einem Jahr gespeichert werden. Es muss eine graphische Oberfläche entwickelt werden, die die Durchlaufdiagramme und Tabellen übersichtlich darstellt und intuitiv in der Benutzung ist. Die Anforderungen an diese Oberfläche sind im Folgenden aufgezählt.

1. Relevante Daten sind übersichtlich in Durchlaufdiagrammen und in tabellarischer Form darzustellen.
2. Die Anlage ist in unterschiedlichen Schnitten zu betrachten, um die Informationsebene zu variieren.
3. Alle Durchlaufdiagramme der Schnitte einer Informationsebene sind gemeinsam darzustellen, um den Produktdurchlauf in der Anlage beobachten zu können. Außerdem ist jeder Schnitt einzeln als Durchlaufdiagramm und in tabellarischer Form darzustellen.
4. Die Nutzung der Oberfläche soll intuitiv und übersichtlich sein.
5. Es sollen die aktuellen sowie die historischen Daten dargestellt werden.
6. Die Oberfläche soll auf einer Darstellungsfläche von ca. 1m x 2m dauerhaft in den Räumlichkeiten der Modellfabrik dargestellt werden.
7. Die Oberfläche ist zusätzlich auf verschiedene Endgeräte wie Tablets oder Laptops auszulegen.
8. Es soll eine Startoberfläche erstellt werden, auf der Informationen zur Modellfabrik und zur Nutzung der Oberfläche zur Verfügung stehen. Außerdem soll die Startoberfläche zur Darstellung der Rückmeldedaten führen und eine Steuerung der Rückmeldedatenerfassung beinhalten.

#### **4.1.2. Aufteilung der Anlage und Definition der relevanten Daten**

Die Rückmeldedaten sollen als Durchlaufdiagramm dargestellt werden, um den Bestand sowie die Durchlaufzeiten des Gesamtprozesses zu erfassen. Dazu wird, wie in den Grundlagen (Abschnitt 2.1.3) erläutert, der Zu- und Abgang benötigt. Der Zugang ist in diesem Fall der Zeitpunkt, an dem das Produkt in die Anlage gelangt und der Abgang der Zeitpunkt, an dem das Produkt die Anlage verlässt. Um mehr Informationen über eventuelle Bottlenecks oder Produktivitätseinbußen zu erhalten, werden zusätzlich zu dem Durchlaufdiagramm der Gesamtanlage auch Durchlaufdiagramme zu verschiedenen Schnitten der Anlage betrachtet. Somit wird die Anforderung 2 erfüllt. Hierbei kann die Anlage auf verschiedene Arten unterteilt werden, die im Folgenden betrachtet werden.

### Alternative 1: Arbeitsvorgang ohne Ausgangslager

Als ein Schnitt wird der Transport zu einer Arbeitsstation sowie der Arbeitsvorgang an sich angesehen. Sobald das Produkt fertig bearbeitet ist, verlässt es den aktuellen Schnitt. In Abbildung 4.1 ist die resultierende Unterteilung durch die Schnitte (im Wechsel blau und gelb gefärbt) für diese Alternative dargestellt.

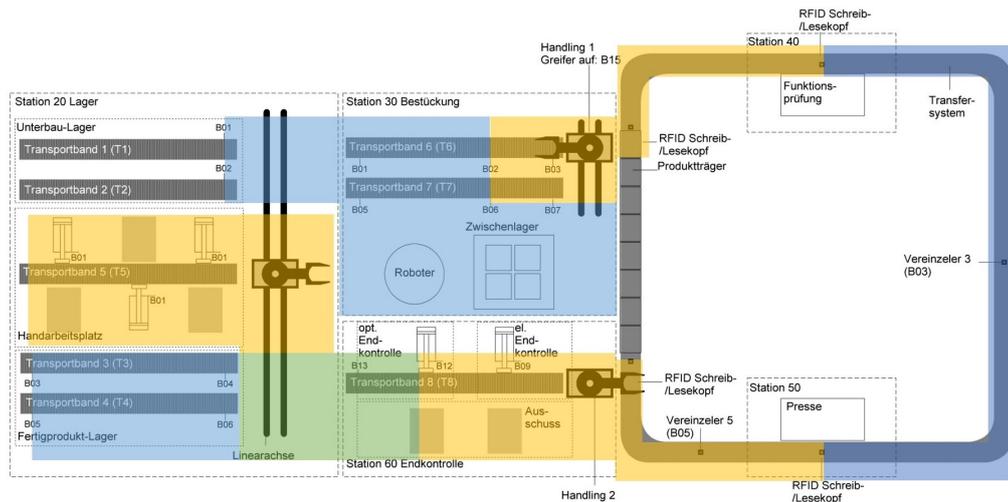


Abbildung 4.1.: Anlagenschnitte durch Arbeitsvorgänge ohne Ausgangslager

Die Anlage unterteilt sich in dieser Alternative in die sechs Arbeitsvorgänge: Bestückung, Funktionsprüfung, Presse, Endkontrolle, Handarbeitsplatz (Handarbeit) sowie Fertigprodukt-Lager mit dem jeweiligen Transport zu diesem Arbeitsvorgang. Das Zwischenlager in Station 30 gehört zu dem Eingangslager bzw. dem Transportweg des Produktes zum Arbeitsvorgang Bestückung. Der Transportweg von der optischen Endkontrolle, entweder zum Handarbeitsplatz oder zum Fertigprodukt-Lager, überschneidet sich in der Abholung des Produktes an Station 60. Diese Überschneidung ist in grün dargestellt.

Der Vorteil dieser Alternative zeigt sich in der Nähe zur Theorie der Durchlaufzeit. Diese wird, wie in den Grundlagen (Abschnitt 2.1.3) beschrieben, durch den Transport zum Arbeitsvorgang und den Arbeitsvorgang selbst berechnet.

Der Nachteil ergibt sich durch die vorliegende Steuerung. Durch diese wird in Station 30 erst ein neuer Unterbau auf das Transportband T6 oder T7 transportiert, wenn dieses wieder leer ist. Das bedeutet, dass das vorherige Zwischenprodukt auf dem Produktträger abgelegt wurde und das Handling 1 sich wieder in Grundposition befindet. Mit der Betrachtung der Schnitte ohne Ausgangslager wäre in dem Schnitt Bestückung für die Zeitspanne, die das Zwischenprodukt zum Produktträger benötigt, die Kapazität des Transportbandes 0. Das liegt daran, dass sich auf dem Transportband noch ein Unterbau befindet. Die Kapazität des

Schnittes Bestückung wechselt also für ein Transportband zwischen 0 und 1. Eine wechselnde Kapazität ist für den Nutzer der Oberfläche nur schwer nachvollziehbar, weshalb die Betrachtung der Schnitte ohne Ausgangslager in diesem Fall nicht optimal ist.

### Alternative 2: Arbeitsvorgang mit Ausgangslager

So wie in der Alternative 1 orientieren sich auch hier die Schnitte an den Arbeitsvorgängen der Stationen. Der Unterschied besteht darin, dass zu einem Schnitt nach der Bearbeitung das dazugehörige Ausgangslager, welches in der Anlagenbeschreibung (Unterkapitel 3.3) bereits definiert wurde, dazu zählt. Die Unterteilung der Anlage ist in Abbildung 4.2 zu sehen.

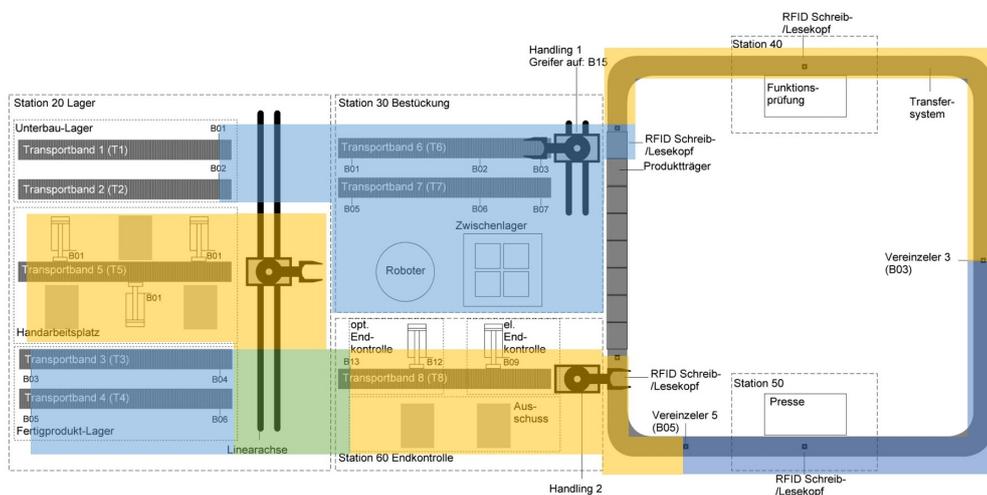


Abbildung 4.2.: Anlagenschnitte durch Arbeitsvorgänge mit Ausgangslager

Wie schon in der Alternative 1 erläutert, unterteilt sich die Anlage auch hier in sechs Schnitte. Die Abbildung 4.2 zeigt die Unterteilung der Anlage mit dem Transportweg, der Bearbeitung und dem in der Anlagenbeschreibung (Unterkapitel 3.3) definierten Ausgangslager. Es ist zu erkennen, dass für jeden Arbeitsvorgang ein Ausgangslager definiert werden kann, ohne dass der Transportweg zur nächsten Arbeitsstation wegfällt.

Die Darstellung der Schnitte mit Ausgangslager bietet den Vorteil, dass die Schnitte flexibler an die Steuerung angepasst werden können. Dadurch kommt es zu keinen Kapazitätsänderungen während der Produktion.

Der Nachteil ist, dass die Definition der Durchlaufzeit verändert werden muss. Für diese Alternative müsste die Liegezeit nach der Bearbeitung, sprich das Ausgangslager, noch

zu dem vorherigen Arbeitsschritt gehören. Dies widerspricht der Definition in den Grundlagen (Abschnitt 2.1.3), wo das Ausgangslager schon zum nachfolgenden Arbeitsvorgang gehört.

### Alternative 3: Arbeitsvorgänge zusammenfassen

Die Darstellung des Gesamtprozesses in sechs Schnitten stellt eine genaue Betrachtung eines jeden Arbeitsschrittes dar. Das Trichtermodell aus Abbildung 3.4 der Anlagenbeschreibung zeigt einen seriellen Ablauf der Arbeitsvorgänge Funktionsprüfung und Presse sowie den potenziellen seriellen Ablauf der Schritte Handarbeit und Fertigprodukt-Lager. Diese seriellen Vorgänge können, wie in den Grundlagen (Abschnitt 2.1.3) beschrieben, zusammengefasst werden. Die Arbeitsschritte Funktionsprüfung und Presse können durch das Transfersystem, welches von Station 10 gesteuert wird, verdichtet werden und als ein Schnitt betrachtet werden. Dieser gibt darüber Aufschluss, wie viele Produkte sich auf dem Transfersystem befinden und wie lange ein Produkt auf dem Transfersystem verweilt. Ebenso kann man den Handarbeitsplatz sowie das Fertigprodukt-Lager (Station 20) zusammengefasst als Endlager betrachten. Der Handarbeitsplatz ist nur eine optionale Vorverarbeitung des Fertigproduktes. Die Schritte Bestückung und Endkontrolle sind an der Rückkopplung beteiligt, was die Verdichtung weiterer Arbeitsvorgänge nicht sinnvoll macht. Die Betrachtung des Prozesses in vier Schnitten ermöglicht die Darstellung in einer weiteren Informationsebene. Die Verdichtung der Schritte ist sowohl mit als auch ohne Ausgangslager möglich. Für die Alternative mit Ausgangslager ist die Unterteilung der Anlage in Abbildung 4.3 dargestellt.

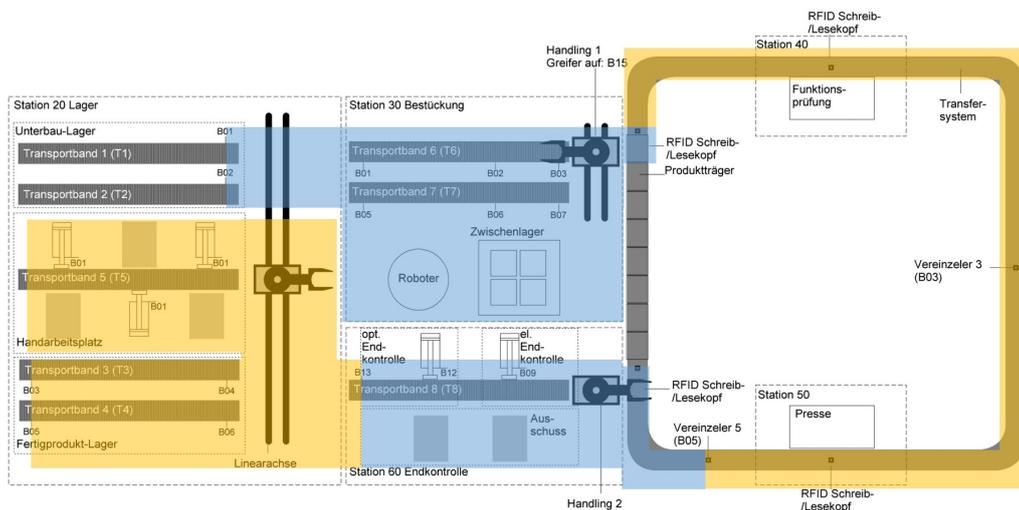


Abbildung 4.3.: Anlagenschnitte durch Zusammenfassung von Arbeitsschritten mit Ausgangslager

Der Vorteil einer weiteren Informationsebene ist, dass sie ein Bindeglied zwischen der genauen Betrachtung eines jeden Arbeitsschrittes und der Betrachtung der Gesamtanlage bietet.

Wird nur diese Alternative umgesetzt, ist der Nachteil, dass die Informationsebene einer jeden zu betrachtenden Arbeitsstation verloren geht. Aus diesem Grund ist die Alternative 3 als Ergänzung zu den Alternativen 1 und 2 zu betrachten. Je nachdem, ob mit oder ohne Ausgangslager, kann die Anlage dann in vier oder sechs Schnitte unterteilt dargestellt werden.

### Vergleich und Entscheidung

Betrachtet werden zunächst die Alternativen 1 und 2. Die Alternative 3 wird, unabhängig davon, welche Alternative zum Einsatz kommt, zusätzlich realisiert. Dadurch wird eine weitere Informationsebene erhalten.

Vergleicht man nun die Alternativen 1 und 2 miteinander, so wird aus den genannten Vor- und Nachteilen deutlich, dass die Alternative 2 besser geeignet ist. Zwar muss die Definition der Durchlaufzeit verändert werden, jedoch kommt es während des Ablaufes so zu keinen Kapazitätsänderungen, wie es in Alternative 1 der Fall ist. Diese Änderung wäre ein großer Nachteil, da sie für den Betrachter schwer verständlich ist. Aus diesem Grund wird die Alternative 2, die Betrachtung der Arbeitsvorgänge mit Ausgangslager, umgesetzt. Die Alternative 3 wird zusätzlich zu der Alternative 2 wie in Abbildung 4.3 dargestellt realisiert.

Für die weitere Betrachtung der Rückmeldedaten und deren Darstellung in Durchlaufdiagrammen wird die Anlage in drei unterschiedlichen Detailtiefen betrachtet. Diese sind in Tabelle 4.1 zusammengefasst dargestellt.

Tabelle 4.1.: Betrachtung des Gesamtprozesses in verschiedenen Detailtiefen

Schnittnummer	Detailtiefe 1 1 Schnitt	Detailtiefe 2 4 Schnitte	Detailtiefe 3 6 Schnitte
1	Gesamtanlage	Bestückung	Bestückung
2		Transfersystem	Funktionsprüfung
3		Endkontrolle	Presse
4		Handarbeit und Fertigprodukt-Lager	Endkontrolle
5			Handarbeit
6			Fertigprodukt-Lager

### Weitere relevante Daten

Nachdem die verschiedenen Schnitte für die Durchlaufdiagramme feststehen, kann über die Inhalte der Tabellen entschieden werden. Eine Tabelle zeigt immer die mögliche Kapazität sowie den Bestand in dem jeweiligen Schnitt an. Außerdem wird immer die Planzeit sowie die Durchlaufzeit dargestellt. Bei der Darstellung der aktuellen Daten ist mit *Bestand* der aktuelle Bestand und mit *letzte Durchlaufzeit* die Durchlaufzeit des letzten Produktes gemeint. Bei der Darstellung der historischen Daten wird der mittlere Bestand sowie die mittlere Durchlaufzeit berechnet und angegeben. Zusätzlich wird im Folgenden für jeden einzelnen Schnitt entschieden, ob relevante Zusatzinformationen existieren. Dazu werden die neun verschiedenen Schnitte einzeln betrachtet und die relevanten Zusatzinformationen benannt.

**Bestückung** Der Schnitt Bestückung lässt sich in drei Abschnitte aufteilen. Zum einen die zwei Transportbänder (T6 und T7) auf denen jeweils ein einpoliges oder ein zweipoliges Produkt bearbeitet werden kann. Die Kapazität eines Transportbandes ist somit  $Kapa_T = 1$ . Da auf T6 nur einpolige und auf T7 nur zweipolige Produkte bearbeitet werden, ergibt sich für ein- und zweipolige Produkte jeweils eine Kapazität von  $Kapa_{1Pol} = Kapa_{2Pol} = 1$ . Zum anderen beinhaltet der Schnitt das Zwischenlager, in dem vier Unterbauten gelagert werden können ( $Kapa_{Zw.Lager} = 4$ ). Somit können sich ein ein- und ein zweipoliges Produkt sowie vier Unterbauten gleichzeitig in diesem Schnitt befinden. Die Gesamtkapazität des Schnittes Bestücken ist also  $Kapa_{Best.} = 6$ . Um dem Nutzer zu verdeutlichen, wieso kein Produkt in den Schnitt gelangt, obwohl die Gesamtkapazität von 6 noch nicht erreicht ist, werden zusätzlich zur Gesamtkapazität die Teilkapazitäten und deren Bestände erfasst und dargestellt. Des Weiteren wird in der Tabelle angegeben, wie viele Unterbauten insgesamt verwendet wurden und wie viele davon aus dem Unterbau-Lager oder dem Prozess kommen.

**Funktionsprüfung** In der Tabelle Funktionsprüfung wird die Anzahl der geprüften Zwischenprodukte sowie die Anzahl der *el. OK* und *el. NOK* Produkte dargestellt.

**Presse** Der Schnitt Presse gibt keine weiteren Informationen über den Produktionsstatus aus. Hier werden keine Zusatzinformationen dargestellt.

**Endkontrolle** Die Endkontrolle gibt die Anzahl der elektrisch und optisch *NOK* Produkte sowie die gesamte Anzahl von *NOK* Produkten an. Außerdem werden die optischen Fehlertypen einzeln gezählt und dargestellt. Zusätzlich werden die wieder eingelasteten Unterbauten, welche vom Roboter von T8 entnommen wurden, aufgelistet. Zuletzt wird der

Ausschuss in elektrischen und optischen Ausschuss unterteilt. Diese werden zusammen mit dem gesamten Ausschuss aufgelistet.

**Handarbeit** In der Ansicht der aktuellen Daten werden die Anzahl der nachbearbeiteten Fehlertypen sowie die gesamt nachbearbeiteten Produkte angezeigt. Zusätzlich werden in der Ansicht der historischen Daten die Anzahl der Fehlertypen gezählt.

**Fertigprodukt-Lager** In dem Schnitt Fertigprodukt-Lager werden die gefertigten Gutteile und die nachbearbeiteten Endprodukte sowie die Gesamtanzahl der gefertigten Produkte angezeigt. Außerdem wird die Anzahl der abgegangenen Produkte insgesamt und aufgeteilt in die entnommenen Fertigprodukte, den el. Ausschuss und den opt. Ausschuss dargestellt.

**Transfersystem** Das Transfersystem fasst die Schnitte Funktionsprüfung und Presse zusammen. Wie in dem Schnitt Funktionsprüfung wird auch hier die Anzahl der geprüften Produkte sowie die Anzahl der *OK* und *NOK* Produkte dargestellt.

**Handarbeit und Fertigprodukt-Lager** Dieser Schnitt fasst den Handarbeitsplatz sowie das Fertigprodukt-Lager zusammen. Zusätzlich zu dem Gesamtbestand wird der Bestand des Handarbeitsplatzes und des Fertigprodukt-Lagers angegeben. Außerdem werden die Informationen über die gefertigten Produkte und entnommene Fertigprodukte dargestellt. In der Ansicht der historischen Werte werden zusätzlich die Anzahl der fehlerhaften Produkte, unterteilt nach Fehlertyp, aufgelistet.

**Gesamtanlage** Die Gesamtanlage zeigt zusätzlich zum Gesamtbestand den Bestand von ein- und zweipoligen Produkten an. Außerdem wird angezeigt, wie viele Unterbauten noch bestückt werden müssen und wie viele Produkte schon gefertigt wurden. Diese unterteilen sich in ein- und zweipolige Gutteile und die nachbearbeiteten Produkte. Zusätzlich werden der gesamte Abgang sowie entnommene Fertigprodukte und elektrischer und optischer Ausschuss aufgelistet.

### 4.1.3. Darzustellender Inhalt

Darzustellen sind die in Abschnitt 4.1.2 ausgewählten Schnitte in Durchlaufdiagrammen sowie deren relevante Daten in tabellarischer Form (Anforderung 1). Der Betrachtungszeitraum für die Live-Durchlaufdiagramme und Tabellen wird auf 30 Minuten festgelegt. Ein Produkt

benötigt, wenn dieses ein Gutteil ist, sprich weder *el.* noch *opt.* NOK ist, ca. 3:10 Minuten vom Zugang in die Anlage, welcher dem Unterbaulager entspricht, bis zum Abgang aus dem Fertigprodukt-Lager. In 30 Minuten kann die Produktion mehrerer Produkte erfolgen und trotzdem bleibt das Durchlaufdiagramm detailliert. Bei der Auswertung der Rückmelde-daten, sprich der Betrachtung der historischen Daten, werden alle gespeicherten Daten im Durchlaufdiagramm dargestellt. Durch eine Navigation ist ein Wechsel des Betrachtungszeit-raumes zu ermöglichen. Die Durchlaufdiagramme sollen für jede in Tabelle 4.1 beschriebene Detailtiefe einmal in einer Übersicht gemeinsam dargestellt werden (Anforderung 3). Diese Übersicht bildet dann jeweils die Gesamtanlage ab. Die Durchlaufdiagramme Bestückung und Endkontrolle werden in der Betrachtung von vier und von sechs Schnitten dargestellt, um den Bezug der Aufteilung in vier und sechs Schnitte und damit den verschiedenen De-tailtiefen nicht zu verlieren. Zur genauen Betrachtung eines einzelnen Schnittes und dessen relevanten Rückmeldedaten soll jeder Schnitt einzeln mit Durchlaufdiagramm und Tabelle dargestellt werden. In der jeweiligen Detailtiefe soll ein Wechsel zwischen den Detaildia-grammen möglich sein. Aus diesem Grund sind in der Detailansicht, links und rechts neben den Durchlaufdiagrammen, Navigationsbuttons zu verwenden. Die Übersicht für die Gesamtanlage entfällt, da hier nur ein Durchlaufdiagramm dargestellt werden würde. Stattdessen wird die Gesamtanlage mit der Tabelle zusammen dargestellt. Es ergeben sich vier verschiedene Inhalte, die die Anforderungen 1 und 3 erfüllen. Diese sind in Abbildung 4.4 dargestellt.



Abbildung 4.4.: Darzustellende Inhalte

#### 4.1.4. Layout

Gefordert ist einerseits ein Leitstand ähnliches Layout, welches auf einer Darstellungsfläche von ca. 1m x 2m dauerhaft in den Räumlichkeiten der Modellfabrik dargestellt werden kann (Anforderung 6). Zum anderen ist ein Layout zu entwickeln, welches über möglichst verschiedene Endgeräte wie Tablets oder Laptops betrachtet werden kann (Anforderung 7). Um beiden Anforderungen gerecht zu werden, werden zwei Layouts konzeptioniert. Im Folgenden ist mit „Leitstand“ das Layout für die dauerhafte Darstellung in den Räumlichkeiten der Modellfabrik und mit „Mobil“ die mobile Alternative der Oberfläche für Laptops und Tablets gemeint.

Beide Oberflächen sollen die aktuellen Live-Daten sowie die historischen Daten, wie in Abschnitt 4.1.3 erläutert, darstellen (Anforderung 5). Die Oberflächen sollen übersichtlich und intuitiv in der Handhabung sein (Anforderung 4). Für die Darstellung der Live-Daten sowie die Auswertung der historischen Daten wird jeweils ein eigenes Layout benötigt, zwischen denen es aber möglich sein soll, zu wechseln. Es müssen demzufolge 4 Layouts entwickelt werden. Zusätzlich zu den darzustellenden Inhalten soll eine Startoberfläche konzipiert werden, auf der generelle Informationen über die Modellfabrik und eine Beschreibung der Oberfläche zu finden sind. Außerdem soll der Nutzer über diese Startoberfläche direkt zu den aktuellen und historischen Rückmeldedaten gelangen (Anforderung 8).

Um das Design, das Layout und die Handhabung testen zu können und so ein Konzept zu entwickeln, wurde eine Beispieloberfläche mit Hilfe von PowerPoint erstellt. Dieses ermöglicht es mit einfachen Mitteln, verschiedene Layouts anschaulich darzustellen und deren Handhabung zu testen. Die PowerPoint Präsentation ist auf der beiliegenden DVD einsehbar. Die Umsetzung erfolgt dann nach dem Vorbild dieser Präsentation. Allerdings können geringe Abweichungen in der Darstellung nicht ausgeschlossen werden. Dies liegt daran, dass die verwendete Plattform evtl. nicht dieselben Darstellungsmöglichkeiten wie PowerPoint bietet.

#### Startoberfläche

Laut Anforderung 8 soll eine Startoberfläche realisiert werden. Über diese ist ein Video über die Modellfabrik und der EForum Beitrag zur generellen Information über die Modellfabrik zu erreichen. Des Weiteren können die Oberflächen der aktuellen und historischen Daten über die Startseite erreicht werden. Zum Schluss soll zusätzlich die Steuerung der Rückmeldedatenerfassung über diese Oberfläche erfolgen. Für das Design wurde das HAW-Logo als Navigationsbuttons genutzt. Außerdem wird ein Bild der Modellfabrik auf der Startseite angezeigt. Die Startoberfläche wird sowohl für die mobile als auch für die Leitstandoberfläche benötigt. Hier ändert sich allerdings nichts im Design, sondern die Oberflächen verweisen

auf die geforderten Oberflächen der aktuellen und historischen Daten. Das Design der Startoberfläche ist in Abbildung 4.5 gezeigt.



Abbildung 4.5.: Design der Startoberfläche

### Mobile Oberfläche

Um die verschiedenen Inhalte aus Abbildung 4.4 darzustellen, stellt sich zunächst die Frage, ob der gesamte Inhalt ohne Navigation dargestellt wird, oder sich der Inhalt durch die Navigation über Buttons ändern soll. Da die Darstellung aller Inhalte auf einen Blick nicht möglich ist, müsste bei einer Realisierung ohne Navigation der Scrollbalken verwendet werden. Eine komfortablere Lösung ist die Navigation durch Buttons. Aus diesem Grund wird sich für diese entschieden. Es ist jeweils ein Inhalt (siehe Abbildung 4.4) gleichzeitig darzustellen, der sich durch die Navigation auswählen und ändern lässt.

**Navigationsleiste** Über eine Navigationsleiste soll sowohl die Startoberfläche dargestellt werden können, als auch zwischen den einzelnen Inhalten aus Abbildung 4.4 und den aktuellen und historischen Werten gewechselt werden. Die Navigationsleiste soll im oberen Bereich der Visualisierung realisiert werden und zusätzlich zu den Navigationsbuttons das HAW-Logo darstellen. Für das Erreichen der Startoberfläche wird ein Button verwendet. Um den Platzbedarf der Navigation gering zu halten, wird sich zur Auswahl der verschiedenen Inhalte aus Abbildung 4.4 zu einem Drop-down-Menü entschieden, welches die Auswahlmöglichkeiten bei Betätigung anzeigt. Bei Betätigung kann zur Übersicht der Gesamtanlage mit Tabelle wie auch zu den Übersichten der vier und sechs Schnitte gewechselt werden. Zusätzlich ist jede Detailansicht der Schnitte erreichbar. Zum Wechsel zwischen den aktuellen und historischen Daten wird ebenfalls ein Drop-down-Menü genutzt. Hier besteht die

Auswahl zwischen der aktuellen und historischen Ansicht. Die Navigationsleiste mit ihren Auswahlmöglichkeiten ist in Abbildung 4.6 dargestellt.

	Home	6 Schnitte Bestückung ▾	Aktuell ▾
		Gesamtanlage	Aktuell
		4 Schnitte Übersicht	Historisch
		6 Schnitte Übersicht	
		4 Schnitte Bestückung	
		4 Schnitte Transfersystem	
		4 Schnitte Endkontrolle	
		4 Schnitte Handarbeit & Fertigprodukt-Lager	
		6 Schnitte Bestückung	
		6 Schnitte Funktionsprüfung	
		6 Schnitte Presse	
		6 Schnitte Endkontrolle	
		6 Schnitte Handarbeit	
		6 Schnitte Fertigprodukt-Lager	

Abbildung 4.6.: Navigationsleiste für die Oberfläche

**Anlagenbild** Zusätzlich zu der Navigationsleiste soll die Anlage mit eingetragenen Schnitten, z.B. wie in Abbildung 4.2, dargestellt werden, um dem Nutzer den Bezug zwischen Anlage und Schnitten zu verdeutlichen. Wird die Anlage gerade aus der Sicht von vier Schnitten beschrieben, ist also ein Detailschnitt oder die Übersicht der vier Schnitte aktiv, so soll das Anlagenbild in die vier Schnitte unterteilt werden. Wird die Anlage aus der Sicht von sechs Schnitten betrachtet, sollen die sechs Schnitte in dem Bild dargestellt sein. Ist ein Detailschnitt ausgewählt, ist dieser in dem Anlagenbild farblich zu markieren, um eine Verbindung zwischen betrachtetem Schnitt und Anlage herzustellen. Ist eine der Übersichten aktiv, so ist kein Schnitt farblich markiert. Bei Auswahl der Gesamtanlage soll jeweils die zuletzt betrachtete Unterteilung als Anlagenbild ohne markierte Schnitte dargestellt werden.

Das Anlagenbild kann entweder mit in der Navigationsleiste angezeigt werden oder muss in das Hauptfenster integriert werden. Da die Navigationsleiste recht schmal ist und das Anlagenbild gut erkennbar sein soll, wird dieses an den linken Rand des Hauptfensters unter dem HAW-Logo positioniert.

Bei der Entwicklung der Beispieloberfläche mit PowerPoint wurde festgestellt, dass die Nutzung der Oberfläche intuitiver wird, wenn die Navigation zwischen den Detailschnitten in einer Detailtiefe zusätzlich zu dem Drop-down-Menu auch über das Anlagenbild möglich ist. Somit soll das Anlagenbild zusätzlich als Navigation genutzt werden können.

**Wechsel zwischen 4 und 6 Schnitten** Der Wechsel zwischen verschiedenen Schnitten in einer Detailtiefe ist nun durch das Anlagenbild möglich. Der Wechsel zwischen den Detailtiefen 4 und 6 Schnitte allerdings nur über das Drop-down-Menü. Um die aktuelle Betrachtung von vier oder sechs Schnitten sichtbar zu machen und zusätzlich den Wechsel der Detailtiefe zu vereinfachen, werden unter das Anlagenbild noch zwei Buttons positioniert. Der eine soll den Detailschnitt Bestückung für vier Schnitte, der andere den Detailschnitt Bestückung für sechs Schnitte aufrufen. Je nachdem, welche Detailtiefe gerade aktiv ist, ist der entsprechende Button farbig zu markieren.

**Besonderheiten bei der Darstellung der Übersichten** Wie in Abschnitt 4.1.3 beschrieben, sollen zusätzlich zu den einzelnen Detailschnitten und der Gesamtanlage eine Übersicht der in der Detailtiefe vorhandenen Schnitte mit einer Vorschau auf die Durchlaufdiagramme dargestellt werden. Werden diese in dem Hauptfenster gezeigt, kann man durch Auswahl einer Vorschau direkt zu dem Detailschnitt gelangen. Das bedeutet, wird gerade die Übersicht der Schnitte dargestellt, existieren drei verschiedene Möglichkeiten zu einem Detailschnitt zu gelangen. Zum einen über das Drop-down-Menü, des Weiteren über das Anlagenbild und in der Übersicht zusätzlich über die Vorschauen.

**Gesamtlayout Mobil aktuell** Der Inhalt aus der Abbildung 4.4 kann im verbleibenden Hauptfenster dargestellt werden. Die Abbildung 4.7 zeigt das Grundgerüst der Oberfläche unterteilt in Navigation, Anlagenbild und Inhaltsfenster. Beispielhaft ist im Inhaltsfenster der Detailschnitt Bestückung dargestellt. Das Inhaltsfenster ist variabel austauschbar.



Abbildung 4.7.: Layout Mobil für aktuelle Daten am Beispiel Bestückung

**Änderungen des Layouts für historische Werte** Das Layout für die historischen Daten der mobilen Oberfläche unterscheidet sich nur durch zusätzliche Navigationsbuttons zur Auswahl des Betrachtungszeitraumes. Hier werden unter das Durchlaufdiagramm Buttons eingefügt, um zwischen 5 Minuten (ein Auftrag dauert in der Regel ca. 3:10 Minuten), 30 Minuten (der Betrachtungszeitraum der aktuellen Live-Daten), einem Tag, einer Woche, einem Monat und einem Jahr wählen zu können. Außerdem wird die Menüleiste um einen Button ergänzt, um die Daten zu aktualisieren. Das Grundgerüst der historischen Oberfläche ist in Abbildung 4.8 dargestellt.



Abbildung 4.8.: Layout Mobil für historische Daten am Beispiel Bestückung

### Leitstandoberfläche

Für die Leitstandoberfläche steht eine Gesamtfläche von ca. 1m x 2m zur Verfügung. Es ist also möglich, auf dieser Fläche mehrere Inhalte aus Abbildung 4.4 darzustellen. Genaugenommen könnten alle vier Inhalte gut sichtbar dargestellt werden. Hier wäre eine Lösung zu finden, wie man dem Nutzer deutlich macht, in welcher Detailtiefe sich die Detailansicht gerade befindet. Außerdem soll, wie in der mobilen Version, das Anlagenbild dargestellt und als zusätzliche Navigation genutzt werden. Unter Beachtung dieser Bedingung kann die zur Verfügung stehende Gesamtfläche in vier gleichgroße Bereiche unterteilt werden und die Inhalte Übersicht, Gesamtanlage und Detailschnitt gemeinsam mit dem Anlagenbild dargestellt werden. Ein Wechsel zwischen der Detailtiefe vier und sechs Schnitte wird, wie in der mobilen Version, durch eine Navigation mit zwei Buttons ermöglicht.

Die Leitstandoberfläche ist in Abbildung 4.9 dargestellt und wird im Folgenden näher betrachtet.

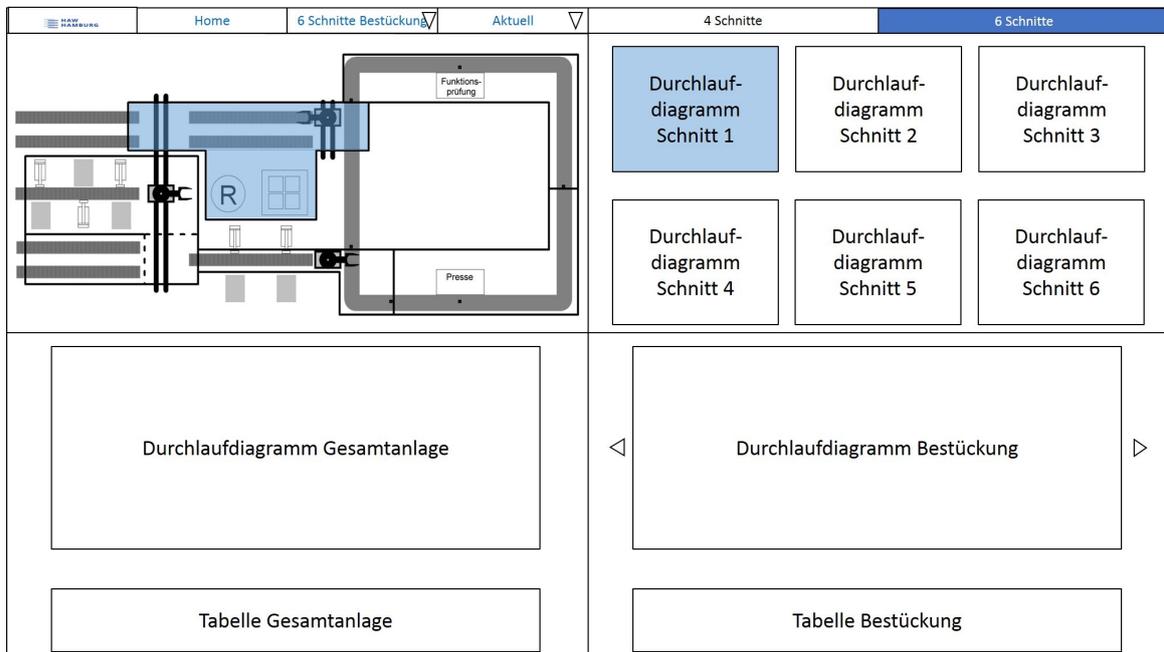


Abbildung 4.9.: Layout Leitstand für aktuelle Daten am Beispiel Bestückung

**Bereich Navigation** Das Anlagenbild samt Navigationsbuttons zu den einzelnen Detailschnitten wird im oberen linken Quadranten positioniert. Zusätzlich zu dem Anlagenbild und dessen Navigation wird die in der mobilen Version beschriebene Navigationsleiste in diesem Bereich oben positioniert. Wie auch in der mobilen Version soll das Anlagenbild sich je nach Detailtiefe und -schnitt einfärben. So ergibt sich im oberen linken Quadranten ein Navigationsfeld.

**Bereich Übersicht** Im oberen rechten Quadranten wird die Übersicht über die vier oder sechs Schritte dargestellt. Je nachdem, welche Detailtiefe ausgewählt ist, wird die jeweilige Übersicht gezeigt. Im Gegensatz zu der mobilen Version der Oberfläche ist die Übersicht somit unabhängig von der Auswahl im Detailbereich dauerhaft sichtbar. Aus diesem Grund wird die Vorschau auf den momentan dargestellten Detailschnitt eingefärbt, um auch in der Übersicht zu erkennen, welcher Schnitt gerade betrachtet wird. Ebenso wie in der mobilen Version kann durch Auswahl einer Vorschau in der Übersicht der gewählte Detailschnitt dargestellt werden. Zusätzlich zu der Übersicht werden die Buttons zum Wechsel der Detail-

tiefe oben in diesem Bereich eingepflegt. Der rechte obere Quadrant kann also sowohl als Übersichts- wie auch als zusätzlicher Navigationsbereich angesehen werden.

**Bereich Gesamtanlage** Im Bereich Gesamtanlage, der sich im unteren linken Quadranten befindet, wird der Inhalt Gesamtanlage, wie in Abschnitt 4.1.1 beschrieben, dargestellt. Das Durchlaufdiagramm sowie die Tabelle der Gesamtanlage ist also dauerhaft sichtbar.

**Bereich Detailschnitt** Der Inhalt des Detailschnittes aus Abschnitt 4.1.1 wird im unteren rechten Quadranten dargestellt. Bei Wechsel des Inhaltes in diesem Bereich ändert sich zusätzlich die Färbung im Bereich Navigation und Übersicht.

**Änderungen des Layouts für historische Werte** Wie auch in der mobilen Version sind ein Button zum neuen Laden der Daten und die Navigationsbuttons des zu betrachtenden Zeitraumes in das Layout zu integrieren. Die Navigationsleiste im Bereich Navigation sowie die Detailansicht werden wie in der mobilen Oberfläche angepasst. Das Grundgerüst der historischen Oberfläche ist in Abbildung 4.10 dargestellt.

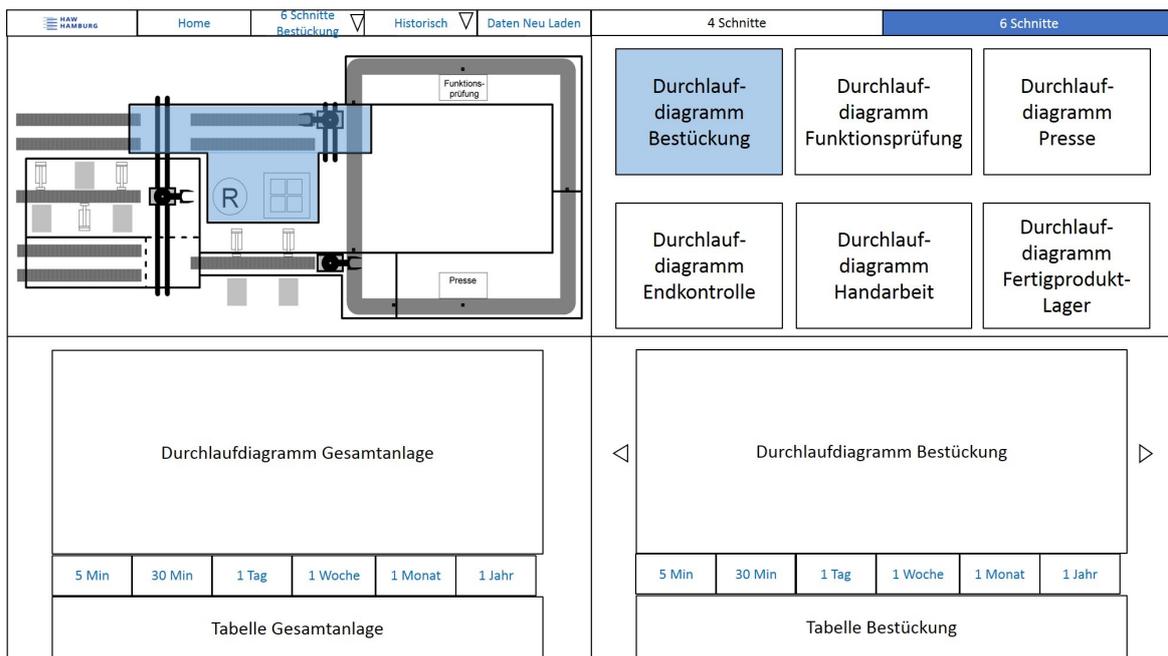


Abbildung 4.10.: Layout Leitstand für historische Daten am Beispiel Bestückung

## 4.2. Datenbereitstellung und -verarbeitung

In diesem Unterkapitel werden verschiedene Alternativen zur Datengewinnung, Übertragung, Verarbeitung, Speicherung und Darstellung vorgestellt. Dazu werden zunächst die Anforderungen festgelegt. Anschließend werden verschiedene Alternativen zur Umsetzung vorgestellt und die resultierenden Alternativen für das Gesamtkonzept durch einen morphologischen Kasten ermittelt. Zum Schluss werden diese durch eine Entscheidungsmatrix verglichen und sich für die beste Alternative entschieden. Sämtliche innerhalb der folgenden Abschnitte dieses Unterkapitels (4.2) referenzierte Anforderungen beziehen sich auf jene, die in Abschnitt 4.2.1 definiert sind.

### 4.2.1. Anforderungen

Die Datenbereitstellung und -verarbeitung soll die für die Visualisierung benötigten Daten aus der Anlage auslesen und zentral sammeln. Welche Rückmeldedaten zusätzlich zum Zu- und Abgang sowie dem Zeitstempel relevant sind, ist in Abschnitt 4.1.2 beschrieben. Anschließend sollen die Daten verarbeitet und gespeichert werden und schließlich auf einer geeigneten Plattform, wie in Unterkapitel 4.1 beschrieben, dargestellt werden.

Die Anforderungen an die Bereitstellung und Verarbeitung der Daten werden in diesem Abschnitt beschrieben und ihnen ein Gewichtungsfaktor zugewiesen. Die hier beschriebenen Anforderungen werden an die Gesamtlösung gestellt, um die Alternativen miteinander vergleichen zu können.

#### **Anforderungen an die Gesamtlösung:**

1. Die Anzahl der Softwareschnittstellen zur Bereitstellung und Verarbeitung der Rückmeldedaten soll gering gehalten werden. Das bedeutet, dass möglichst mehrere Schritte, von der Gewinnung bis zur Darstellung, im selben Programmcode abgearbeitet werden sollten, um möglichst wenig verschiedene Programme zu nutzen. Dies soll den Aufwand, der durch die Übergabe der Daten zwischen unterschiedlichen Programmen entsteht, verringern.
2. Der Implementierungsaufwand ist gering zu halten.
3. Der Übertragungsweg der Rückmeldedaten und der Daten für die Steuerung soll sich unterscheiden, um eine Trennung der Rückmeldedaten und der für die Steuerung relevanten Daten zu ermöglichen.
4. Der Prozessablauf und damit die Steuerungsprogramme sollen nicht verändert werden.

5. Es dürfen keine relevanten Daten verpasst werden.
6. Die Gesamtkosten sind gering zu halten. Aus diesem Grund sind Alternativen, die mit den vorhandenen Mitteln realisierbar sind, zu bevorzugen.
7. Die Übertragung der Daten soll sowohl mittels vorhandenem Bussystem wie auch über WLAN möglich sein.
8. Die gewonnenen Daten sollen ein Jahr lang gespeichert werden.
9. Die Rückmeldedaten sollen übersichtlich und aussagekräftig gespeichert werden.
10. Es sind Industrielösungen zu bevorzugen.
11. Die Rückmeldedaten sind bei der Darstellung der Live-Daten so schnell wie möglich, spätestens jedoch nach 5 Sekunden, in der Visualisierung darzustellen.
12. Die graphische Oberfläche muss auf möglichst vielen Endgeräten nutzbar sein.
13. Eine benötigte Zusatzsoftware zur Darstellung der graphischen Oberfläche ist möglichst zu vermeiden.

**Gewichtung der Anforderung:**

Die Anforderungen werden nachfolgend mit einem Gewichtungsfaktor versehen, welcher die Wichtigkeit von 5 „sehr wichtig“ bis 1 „unwichtig“ beschreibt.

Die Anforderungen 5, 7 und 12 werden als am wichtigsten (Gewichtungsfaktor 5) betrachtet, um die Datenkonsistenz zu sichern und die Nutzung der Daten komfortabel zu gestalten.

Anforderung 3, die Unterscheidung des Übertragungsweges und Anforderung 11, die Darstellung der Rückmeldedaten nach spätestens 5 Sekunden sind etwas weniger wichtig als die Konsistenz der Daten und die komfortable Nutzung und werden deshalb mit 4 gewichtet.

Die Anforderungen 4, 8 und 10 werden als wichtig, aber weniger wichtig als die zuvor genannten Anforderungen angesehen und erhalten damit eine Gewichtung von 3.

Die Anforderung 9 ist in diesem Vergleich eher unwichtig und geht mit der Gewichtung 2 in die Wertung. Die Anforderungen 1, 2, 6 und 13 werden einfach gewertet, da diese am unwichtigsten sind.

Die Gewichtungsfaktoren der Anforderungen sind in Tabelle 4.2 zusammengefasst.

Tabelle 4.2.: Übersicht der gewichteten Anforderungen an die Datenbereitstellung und -verarbeitung

<b>Anforderung</b>	1	2	3	4	5	6	7	8	9	10	11	12	13
<b>Gewichtung</b>	1	1	4	3	5	1	5	3	2	3	4	5	1

## 4.2.2. Teil-Alternativen

### Gewinnung der Rohdaten

**WLAN Input/Output (I/O)** Eine Alternative zur Gewinnung der Daten sind WLAN I/O's. Hierzu wird das Produkt „WISE-4000 - IoT WLAN Wireless I/O-Modul“ betrachtet. Dieses speichert lokal die anliegenden Signale (1 oder 0) mit aktuellem Zeitstempel in einer JSON-Datei zwischen. Die Datei kann über einen integrierten Webserver ausgelesen und weiterverarbeitet werden. Bei der Weiterverarbeitung müssen dann die registrierten Flanken der Sensoren gezählt werden, um den Zu- und Abgang zu bestimmen. Die Durchlaufzeit ergibt sich aus der Differenz der Zeitstempel vom Abgang und vom Zugang bei einem Flankenwechsel. Eine Konvertierung in eine Exceldatei ist in diesem Dateiformat möglich. Das bedeutet, dass die Rohdaten nach Abholung in einer Exceldatei vorliegen und so weiterverarbeitet werden können. Für weitere Informationen zu diesem Produkt siehe [13].

Der Vorteil dieser Alternative ist die Trennung der Steuerung und der Gewinnung der Rückmeldedaten. Das Steuerungsprogramm wird nicht verändert, sondern die Daten direkt durch entsprechende Sensorik an das I/O-Modul übertragen. Außerdem ist eine Datensammlung stationsunabhängig möglich.

Der Nachteil ist, dass nur Sensoren und keine Events aus den SPS-Steuerungsprogrammen verwendet werden können. Um ein Event wie z.B. die Demontage des Zwischenproduktes erfassen zu können, müssten zusätzliche Sensoren verbaut werden. Die zurzeit verbaute Sensorik ist für diesen Anwendungsfall nicht zuverlässig genug. Aus diesem Grund müssten zusätzlich zu den ergänzenden Sensoren auch die verwendeten Sensoren erneuert werden. Dies ist ein Kosten- und Aufwandspunkt. Des Weiteren muss sich im weiteren Verlauf Gedanken über die Verarbeitung der Daten hinsichtlich des Zählens des Zu- und Abganges gemacht werden. Die Sensorik schreibt nur eine 1 oder 0 in die Datei. Das Zählen des Auslösens des Sensors muss in der Verarbeitung passieren. Außerdem müsste das Auslesen der Datei automatisiert werden, um die Live-Daten aktuell zu halten.

**Datenprotokollierung (DataLog) direkt auf der SPS** Diese Alternative fasst die von Siemens gegebene Alternative zur Gewinnung und lokalen Speicherung der Daten auf. Hier werden die gewünschten Daten durch einen Baustein direkt in der Steuerung mit aktuellem

Zeitstempel in einer Exceldatei gespeichert, welche über den integrierten Webserver abgerufen und anschließend weiterverarbeitet werden kann. Es ist also möglich, den Zu- und Abgang in dem Steuerungsprogramm zu zählen und bei einer Änderung den aktuellen Zählerstand in eine Exceldatei zu schreiben. Die erste Verarbeitung der Sensordaten kann so auf der SPS vorgenommen werden. Für genauere Informationen zu dieser Alternative siehe [17].

Der Vorteil dieser Alternative ist die Nutzung einer implementierten Funktion von Siemens. Die Speicherung erfolgt direkt durch die Steuerung und es kann entschieden werden, wann die Speicherung stattfinden soll. Das heißt, die Daten werden nicht zyklisch geschrieben, sondern die Steuerung schreibt die Daten nur bei Änderung automatisch in die Datei.

Wie bei der Speicherung der Daten durch die WLAN I/O's müsste das Auslesen der Exceldatei durch ein Programm automatisiert werden, was einen Nachteil darstellt. Hier müsste sichergestellt werden, dass die Datei während des Auslesens nicht beschrieben wird. Des Weiteren müssen entweder die Daten aller Stationen auf einer gesammelt werden, um eine Exceldatei abzuspeichern, oder jede Station schreibt ihre eigene Exceldatei. Schreibt jede Station ihre eigene Exceldatei, kann nicht garantiert werden, dass die Daten alle den gleichen Zeitstempel erhalten, da jede SPS ihre eigene Systemuhrzeit nutzt und diese nicht ganz genau aneinander abgestimmt werden können.

**Erfassung der Sensoren und Steuerungsevents durch Abfrage der SPS-Daten** In dieser Alternative werden die auf der SPS vorhandenen Daten verwendet. So kann die vorhandene Sensorik ebenfalls als Event genutzt werden. Zusätzlich können jedoch auch die Events aus dem Steuerungsprogramm verwendet werden. So muss für das Beispiel der Demontage keine neue Sensorik verbaut werden, sondern es kann das Event, das diesen Vorgang auslöst, aus dem Steuerungsprogramm der Station 30 ausgewertet werden. Wie in der zuvor beschriebenen Alternative (DataLog) ist es möglich, den Zu- und Abgang im Steuerungsprogramm zu zählen und diese Zähler zyklisch auf Änderung abzufragen.

Der Vorteil dieser Alternative ist, dass zusätzlich zu den Sensoren auch steuerungsrelevante Events zur Auswertung genutzt werden können. Außerdem kann die Steuerung zur ersten Verarbeitung, dem Zählen des Zu- und Abganges, verwendet werden.

Der Nachteil zeigt sich darin, dass die Gewinnung der Rückmeldedaten nicht von dem Steuerungsprogramm entkoppelt ist. Außerdem sind die Informationen an die einzelnen Stationen gebunden und müssten durch einen anschließenden Schritt zentral gesammelt und ausgewertet werden.

## Übertragung

Zur Übertragung der Daten nach der Gewinnung werden drei verschiedene Alternativen vorgestellt.

**Abholen der lokal gespeicherten Dateien von entsprechenden Webservern** Diese Alternative wird eingesetzt, wenn sich bei der Gewinnung der Daten für WLAN I/O's oder Data-Log entschieden wird. Die lokal auf dem Gerät gespeicherten Dateien müssen zyklisch vom Webserver abgeholt und verarbeitet sowie gespeichert werden.

Der Nachteil dieser Alternative ist der Implementierungsaufwand eines solchen Programmes. Es muss sich zyklisch mit dem Webserver verbinden und die Datei abholen. Außerdem stellt sich die Frage, wie die Dateiverwaltung stattfindet. Ein zyklisches Löschen und Neuspeichern von Dateien benötigt Rechenzeit und beansprucht den Speicher des Servers.

Der Vorteil dieser Alternative ist, dass die lokal gespeicherten Daten direkt als Datei auf dem Server vorliegen und nicht erst abgespeichert werden müssen.

**Erstellung einer OPC UA Verbindung** In dieser Alternative wird der in der Steuerung vorhandene OPC UA Server verwendet, um die erforderlichen Daten mittels OPC UA an einen Client zu übertragen. Hierzu wäre die Firmware der Steuerung auf mindestens V2.0 hochzurüsten und anschließend der OPC UA Server zu aktivieren. Der Client zum Empfangen der Daten muss selbst programmiert werden.

Der Vorteil dieser Alternative ist zum einen der Industriestandard. In vielen Firmen wird OPC UA zur Übertragung derartiger Daten verwendet. Zum anderen ist der Server auf der SPS implementiert und es kann, nach Änderung der Einstellungen der Steuerung, direkt durch einen Client auf die Daten zugegriffen werden.

Ein Nachteil ist ein erhöhter Aufwand der Einarbeitung in die OPC UA Technologie zur Entwicklung des Clients. Durch die Verwendung in der Industrie gibt es allerdings in mehreren Programmiersprachen Bibliotheken oder Pakete, die den Aufbau eines OPC UA Clients erleichtern. Außerdem ist die Firmware der SPS auf mindestens V2.0 zu aktualisieren. Laut Siemens sollte dies allerdings keinen Einfluss auf die Steuerungsprogramme haben. Zum Schluss ist bei dieser Alternative zu beachten, dass der Zeitstempel nicht automatisch mit erfasst wird, sondern mit übertragen und abgespeichert werden muss.

**Node-Red und Raspberry Pi** Diese Alternative ist in Anlehnung an die Studienarbeit von Tim Geiger [21] entstanden. In dieser wird eine Verbindung zwischen dem Raspberry Pi und dessen Programm Node-Red und der Modellfabrik hergestellt und die erfassten Daten dargestellt. (Siehe [21] S.23ff). Zur Kommunikation zwischen Node-Red und der Modellfabrik werden die Nodes „node-red-contrib-s7comm“ verwendet. Mit diesen ist, nach einer Konfiguration des Steuerungsprogrammes, das Auslesen und auch das Schreiben von SPS-Variablen möglich.

Vorteil dieser Alternative ist, dass die Funktion an der Modellfabrik durch die Studienarbeit nachgewiesen ist, und ein guter Leitfaden zum Auslesen der Daten besteht. Zusätzlich können die gewonnenen Daten mit Node-Red direkt in eine Datenbank geschrieben werden.

Der Nachteil ist, dass sich in die ungewohnte visuelle Programmierung von Node-Red eingearbeitet werden muss.

## Speicherung

Zur Speicherung der Daten werden zwei mögliche Arten der Speicherung betrachtet.

**Datenspeichern in einer Exceldatei** Wenn die Daten durch die Übertragungsart in einer Exceldatei vorliegen, dann werden diese auch auf dem verarbeitenden System als solche gespeichert.

Der Vorteil dieser Alternative ist ein ausbleibendes Konvertieren der Exceldaten in ein anderes Speicherformat. So werden die Softwareschnittstellen und verwendeten Programme minimiert.

Der Nachteil ist die unübersichtliche Darstellung der Daten. Excel ist für eine Datenspeicherung und das Ausgeben bestimmter Datensätze, die über ein Jahr gesammelt werden, nicht optimiert.

**Datenspeicherung in einer Datenbank** Liegen die Daten nicht durch die Übertragung in einer Exceldatei vor, so sind sie in einer Datenbank zu speichern.

Dies hat den Vorteil, dass eine Datenbank dafür ausgelegt ist, große Mengen von Daten übersichtlich zu speichern und die Datensätze nach bestimmten Voraussetzungen zu sortieren und auszugeben.

Der Nachteil dieser Alternative ist der Implementierungsaufwand. Die Datenbank muss aufgesetzt und eingerichtet werden. Das Lesen und Schreiben der Datenbank muss durch ein Programm stattfinden, was die Softwareschnittstellen im Gesamtaufbau erhöht. Allerdings

gibt es in verschiedenen Programmiersprachen Bibliotheken oder Pakete, um die Verwaltung der Datenbank zu vereinfachen.

## **Verarbeitung**

Die Daten, die gespeichert werden, können entweder die gewonnenen Rohdaten oder die fertig verarbeiteten, darzustellenden Nutzdaten sein. Welche Alternative hier gewählt wird, hängt von der Übertragung der Daten ab. Im Folgenden werden drei Alternativen beschrieben.

**Verarbeitung der in Excel gespeicherten Rohdaten vor der Visualisierung** In dieser Alternative werden die gespeicherten Exceldateien vor der Visualisierung verarbeitet.

Der Vorteil besteht darin, dass die abgespeicherten Rohdaten direkt verwendet werden können. Es muss also keine zweite Exceldatei mit den Nutzdaten abgespeichert werden.

Der Nachteil ist, dass die Nutzdaten nach Beendigung des Programms verloren gehen, diese also bei jedem Start wieder neu berechnet werden müssen. Dies wird die Darstellung der Daten verlangsamen.

**Verarbeitung der in Excel gespeicherten Rohdaten und anschließende Speicherung der Nutzdaten in Excel** Die in einer Exceldatei gespeicherten Rohdaten können in Excel weiterverarbeitet werden und anschließend in eine Nutzdatendatei gespeichert werden. Die anschließende Visualisierung bedient sich dann der Nutzdaten.

Der Vorteil wäre hier, dass kein weiteres Programm zur Verarbeitung der Daten herangezogen wird. Außerdem können die Daten direkt visualisiert werden.

Allerdings liegen immer zwei Dateien, zum einen die Rohdaten und zum anderen die Nutzdaten, auf dem Server vor. Dies zieht doppelten Speicherbedarf nach sich, was als Nachteil angesehen wird.

**Verarbeitung der übertragenen Rohdaten vor der Speicherung in eine Datenbank** Werden die Daten durch OPC UA oder Node-Red übertragen, so werden sie vor der Speicherung verarbeitet und die Nutzdaten werden abgespeichert. Hierzu wird im Fall von OPC UA die Programmiersprache, in der auch der Client geschrieben wird, gewählt. In Node-Red besteht ebenfalls die Möglichkeit, die empfangenen Daten vorher zu verarbeiten und anschließend in eine Datenbank zu schreiben.

Dies hat den Vorteil, dass die Daten in der Datenbank einen Bezug zu den visualisierten Daten haben und damit leicht verständlich sind. Außerdem findet die Visualisierung unabhängig von der Verarbeitung statt. Dauert die Verarbeitung länger, können die zuvor gespeicherten Daten trotzdem dargestellt werden.

## Plattform

Zur Darstellung der verarbeiteten Nutzdaten in Durchlaufdiagramme und Tabellen soll eine Oberfläche entweder als eigenständiges Programm auf einem Endgerät oder als Webserver entwickelt werden.

**Eigenständiges Programm** In dieser Alternative wird mit einer beliebigen Programmiersprache eine graphische Oberfläche erstellt, die nach den Vorgaben aus Unterkapitel 4.1 die Nutzdaten visuell darstellt.

Der Vorteil dieser Alternative ist die freie Entscheidung der Programmiersprache. Dadurch kann auf vorhandenes Wissen aufgebaut werden, was den Implementierungsaufwand verringert.

Der Nachteil ist, dass die Benutzeroberfläche auf jedem Endgerät implementiert werden muss. Zusätzlich zu einer Windows Version müsste für Android eine App entwickelt werden, um die Nutzbarkeit auf verschiedenen Endgeräten zu ermöglichen.

Des Weiteren werden die gespeicherten Daten durch eigenständige Programme von jedem laufenden Programm angefragt. Das bedeutet, dass der Zugriff auf die Daten der Anzahl der Nutzer entspricht. Dies kann zu Problemen beim Auslesen führen.

**Webserver** In dieser Alternative wird ein Webserver implementiert und mittels HTML, CSS, JavaScript und PHP die Benutzeroberfläche nach den Vorgaben aus Unterkapitel 4.1 realisiert.

Der Vorteil liegt abgesehen davon, dass es von jedem Endgerät im Netzwerk nutzbar ist, auch darin, dass die Oberfläche nur auf einem Server läuft. Dies verringert den Zugriff auf die Daten von Anzahl der Nutzer auf einen Zugriff durch den Webserver.

Außerdem lässt sich das Konzept in der Zukunft über eine VPN-Verbindung erweitern, so dass der Webserver und damit die Visualisierung auch von außerhalb des Modellfabrik-Netzwerkes abrufbar wird.

Der Nachteil dieser Alternative ist der Implementierungsaufwand. Zur Darstellung der Daten wird nicht eine, sondern vier Scriptsprachen benötigt.

### 4.2.3. Gesamt-Alternativen

In diesem Abschnitt werden die Teil-Alternativen unter Zuhilfenahme eines morphologischen Kastens zusammengefasst und daraus die verschiedenen Gesamtkonzepte zur Datenbereitstellung und -verarbeitung ermittelt und vorgestellt. Hierbei werden nur die technisch realisierbaren Alternativen betrachtet. Der morphologische Kasten ist in Tabelle 4.3 dargestellt.

Tabelle 4.3.: Morphologischer Kasten

<b>Gewinnung der Rohdaten</b>	WLAN I/O's	DataLog	Abfrage der SPS-Daten
<b>Übertragung</b>	Abholen der gespeicherten Daten	OPC UA	Node-Red
<b>Speicherung</b>	Excel	Datenbank	
<b>Verarbeitung</b>	Excel Rohdaten Verarbeitung vor der Visualisierung	Excel Rohdaten in Nutzdaten speichern	Nutzdaten nach der Übertragung verarbeiten und speichern
<b>Plattform</b>	Eigenständiges Programm	Webserver	

Alle plausiblen Kombinationen der Teil-Alternativen sind in den Abbildungen 4.11 und 4.12 gezeigt.

Gewinnung der Rohdaten	WLAN I/O's	DataLog	Abfrage der SPS-Daten
Übertragung	Abholen der gespeicherten Daten	OPC UA	Node-Red
Speicherung	Excel	Datenbank	
Verarbeitung	Excel Rohdaten Verarbeitung vor der Visualisierung	Excel Rohdaten in Nutzdaten speichern	Nutzdaten nach Übertragung verarbeiten und speichern
Plattform	Eigenständiges Programm	Webserver	

(a) Kombination 1

Gewinnung der Rohdaten	WLAN I/O's	DataLog	Abfrage der SPS-Daten
Übertragung	Abholen der gespeicherten Daten	OPC UA	Node-Red
Speicherung	Excel	Datenbank	
Verarbeitung	Excel Rohdaten Verarbeitung vor der Visualisierung	Excel Rohdaten in Nutzdaten speichern	Nutzdaten nach Übertragung verarbeiten und speichern
Plattform	Eigenständiges Programm	Webserver	

(b) Kombination 2

Gewinnung der Rohdaten	WLAN I/O's	DataLog	Abfrage der SPS-Daten
Übertragung	Abholen der gespeicherten Daten	OPC UA	Node-Red
Speicherung	Excel	Datenbank	
Verarbeitung	Excel Rohdaten Verarbeitung vor der Visualisierung	Excel Rohdaten in Nutzdaten speichern	Nutzdaten nach Übertragung verarbeiten und speichern
Plattform	Eigenständiges Programm	Webserver	

(c) Kombination 3

Gewinnung der Rohdaten	WLAN I/O's	DataLog	Abfrage der SPS-Daten
Übertragung	Abholen der gespeicherten Daten	OPC UA	Node-Red
Speicherung	Excel	Datenbank	
Verarbeitung	Excel Rohdaten Verarbeitung vor der Visualisierung	Excel Rohdaten in Nutzdaten speichern	Nutzdaten nach Übertragung verarbeiten und speichern
Plattform	Eigenständiges Programm	Webserver	

(d) Kombination 4

Abbildung 4.11.: Kombinationen 1 bis 4 der Teil-Alternativen des morphologischen Kastens

Gewinnung der Rohdaten	WLAN I/O's	DataLog	Abfrage der SPS-Daten
Übertragung	Abholen der gespeicherten Daten	OPC UA	Node-Red
Speicherung	Excel	Datenbank	
Verarbeitung	Excel Rohdaten Verarbeitung vor der Visualisierung	Excel Rohdaten in Nutzdaten speichern	Nutzdaten nach Übertragung verarbeiten und speichern
Plattform	Eigenständiges Programm	Webserver	

(a) Kombination 5

Gewinnung der Rohdaten	WLAN I/O's	DataLog	Abfrage der SPS-Daten
Übertragung	Abholen der gespeicherten Daten	OPC UA	Node-Red
Speicherung	Excel	Datenbank	
Verarbeitung	Excel Rohdaten Verarbeitung vor der Visualisierung	Excel Rohdaten in Nutzdaten speichern	Nutzdaten nach Übertragung verarbeiten und speichern
Plattform	Eigenständiges Programm	Webserver	

(b) Kombination 6

Gewinnung der Rohdaten	WLAN I/O's	DataLog	Abfrage der SPS-Daten
Übertragung	Abholen der gespeicherten Daten	OPC UA	Node-Red
Speicherung	Excel	Datenbank	
Verarbeitung	Excel Rohdaten Verarbeitung vor der Visualisierung	Excel Rohdaten in Nutzdaten speichern	Nutzdaten nach Übertragung verarbeiten und speichern
Plattform	Eigenständiges Programm	Webserver	

(c) Kombination 7

Gewinnung der Rohdaten	WLAN I/O's	DataLog	Abfrage der SPS-Daten
Übertragung	Abholen der gespeicherten Daten	OPC UA	Node-Red
Speicherung	Excel	Datenbank	
Verarbeitung	Excel Rohdaten Verarbeitung vor der Visualisierung	Excel Rohdaten in Nutzdaten speichern	Nutzdaten nach Übertragung verarbeiten und speichern
Plattform	Eigenständiges Programm	Webserver	

(d) Kombination 8

Gewinnung der Rohdaten	WLAN I/O's	DataLog	Abfrage der SPS-Daten
Übertragung	Abholen der gespeicherten Daten	OPC UA	Node-Red
Speicherung	Excel	Datenbank	
Verarbeitung	Excel Rohdaten Verarbeitung vor der Visualisierung	Excel Rohdaten in Nutzdaten speichern	Nutzdaten nach Übertragung verarbeiten und speichern
Plattform	Eigenständiges Programm	Webserver	

(e) Kombination 9

Gewinnung der Rohdaten	WLAN I/O's	DataLog	Abfrage der SPS-Daten
Übertragung	Abholen der gespeicherten Daten	OPC UA	Node-Red
Speicherung	Excel	Datenbank	
Verarbeitung	Excel Rohdaten Verarbeitung vor der Visualisierung	Excel Rohdaten in Nutzdaten speichern	Nutzdaten nach Übertragung verarbeiten und speichern
Plattform	Eigenständiges Programm	Webserver	

(f) Kombination 10

Gewinnung der Rohdaten	WLAN I/O's	DataLog	Abfrage der SPS-Daten
Übertragung	Abholen der gespeicherten Daten	OPC UA	Node-Red
Speicherung	Excel	Datenbank	
Verarbeitung	Excel Rohdaten Verarbeitung vor der Visualisierung	Excel Rohdaten in Nutzdaten speichern	Nutzdaten nach Übertragung verarbeiten und speichern
Plattform	Eigenständiges Programm	Webserver	

(g) Kombination 11

Gewinnung der Rohdaten	WLAN I/O's	DataLog	Abfrage der SPS-Daten
Übertragung	Abholen der gespeicherten Daten	OPC UA	Node-Red
Speicherung	Excel	Datenbank	
Verarbeitung	Excel Rohdaten Verarbeitung vor der Visualisierung	Excel Rohdaten in Nutzdaten speichern	Nutzdaten nach Übertragung verarbeiten und speichern
Plattform	Eigenständiges Programm	Webserver	

(h) Kombination 12

Abbildung 4.12.: Kombinationen 5 bis 12 der Teil-Alternativen des morphologischen Kastens

Die Abbildungen 4.11 und 4.12 zeigen, dass die Kombinationen 1 und 2 sowie 5 und 6 alle die Teil-Alternative „Excel Rohdaten Verarbeitung vor der Visualisierung“ beinhalten. Wie in der Beschreibung dieser Teil-Alternative erwähnt, ist der Nachteil, dass die Nutzdaten bei jedem Aufruf der Visualisierung neu berechnet werden müssen, was einen zeitlichen Nachteil gegenüber der Alternative der Speicherung der Nutzdaten bringt. Die Anforderung 11 kann in dieser Alternative nicht oder nur zum Teil eingehalten werden. Außerdem sind die Rohdaten vor der Verarbeitung nicht aussagekräftig und erfüllen damit die Anforderung 9 nicht. Der Vorteil, nur eine Exceldatei abzuspeichern, überwiegt hier nicht den Nachteilen. Aus diesem Grund werden die Kombinationen 1 und 2 sowie 5 und 6 nicht weiter betrachtet.

Die zu vergleichenden Kombinationen lassen sich um die Hälfte verringern, wenn sich im Vorfeld für eine Plattform der Oberfläche entschieden wird. Hierfür sind die Anforderungen 12 und 13 relevant. Die Oberfläche soll auf möglichst vielen Endgeräten dargestellt werden können, und es soll auf Zusatzsoftware zur Darstellung verzichtet werden. Die Teil-Alternative „Eigenes Oberflächenprogramm“ hat, wie in Abschnitt 4.2.2 beschrieben, den Nachteil, dass das Programm überall installiert werden muss und für Android eine zusätzliche App entwickelt werden müsste. Viele Endgeräte wie zum Beispiel Tablet oder PC besitzen hingegen einen Webbrowser, auf dem die Oberfläche im Falle eines Webservers dargestellt werden kann. Dieser Vorteil überwiegt den Nachteil des Webservers, dass mehrere Scriptsprachen benötigt werden. Aus diesem Grund wird sich für den Webserver als Plattform entschieden. Die Kombinationen 3, 7, 9 und 11 werden somit nicht weiter ausgewertet.

Die Anforderung 7, dass die Alternativen sowohl über das vorhandene Bussystem wie auch über WLAN nutzbar sind, ist in allen Kombinationen möglich. Über den Unterschied der Geschwindigkeit der Übertragung und Speicherung und damit der Zeit, die bis zur Darstellung vergangen ist (Anforderung 11), lässt sich zurzeit keine Aussage machen. Die Anforderungen 12 und 13 wurden bereits in der Entscheidung zur Plattform ausgewertet. Aus diesem Grund werden die Anforderungen 7, 11, 12 und 13 nicht mit in die Entscheidung einbezogen.

### **Alternative 1**

In Alternative 1 (Kombination 4) werden zur Datengewinnung WLAN I/O's genutzt und die lokal gespeicherten Dateien zyklisch mittels eines Programms vom Webserver ausgelesen. Die Rohdaten werden auf dem verarbeitenden System in einer Exceldatei verarbeitet und als Nutzdaten in Excel gespeichert. Ein Webserver greift auf die Exceldatei zu und stellt die Rückmeldedaten wie gewünscht dar.

### **Alternative 2**

Die Alternative 2 (Kombination 8) nutzt zur Datengewinnung das von Siemens zur Datensicherung zur Verfügung gestellte DataLog. Die Daten müssen über virtuelle I/O's an die Station 10 gesendet werden. Diese speichert die Daten der Gesamtanlage in eine Exceldatei, die zyklisch von einem Programm, welches auf dem verarbeitenden System läuft, vom Webserver der SPS abgeholt wird und nach der Verarbeitung als Nutzdaten in einer Exceldatei zur Verfügung steht. Ein Webserver liest die Nutzdaten aus der Exceldatei aus und stellt sie dar.

### Alternative 3

Die Alternative 3 (Kombination 10) fragt die SPS-Daten zyklisch mittels einer OPC UA Verbindung zu jeder Steuerung ab. Die Daten werden bei einer Änderung der OPC UA Daten verarbeitet und anschließend in eine Datenbank geschrieben. Dargestellt werden die Rückmeldedaten mit Hilfe eines Webservers.

### Alternative 4

Die Alternative 4 (Kombination 12) nutzt zur Übertragung und Verarbeitung Node-Red. Die verarbeiteten Daten werden in eine Datenbank gespeichert und von einem Webserver wieder ausgelesen.

## 4.2.4. Vergleich und Entscheidung

Die noch möglichen Alternativen werden im Folgenden nach den verbleibenden, in Abschnitt [4.2.1](#) beschriebenen, Kriterien bewertet. Die Bewertung beschreibt die Erfüllung der Anforderung von 5(++), „Sehr gut“ bis 1(-) „überhaupt nicht“.

Die Anforderung 1, wenig Schnittstellen zu nutzen, ist in Alternative 1 am besten erfüllt und mit (++) bewertet. Die Schnittstellen beschränken sich auf das herunterladende Programm, die Speicherung der Nutzdaten und die Darstellung der Oberfläche. Die Alternative 2, die Nutzung von DataLog, hat durch das SPS-Programm eine Schnittstelle mehr. Diese Alternative wird deshalb mit (+) bewertet. Die Anforderungen 3 und 4 haben durch die Speicherung in einer Datenbank eine Schnittstelle und ein Programm mehr als die Alternative 2. Aus diesem Grund werden diese Alternativen mit (o) bewertet.

Den Implementierungsaufwand gering zu halten (Anforderung 2), wird von der Alternative 2, dem DataLog, am besten eingehalten und mit (+) bewertet. Das Steuerungsprogramm schreibt vorverarbeitete Daten bei einer Änderung in die Exceldatei, die zyklisch ausgelesen wird. Implementiert werden muss die Übertragung der Daten aus den anderen Stationen an Station 10, sowie ein Programm, das die Dateien ausliest. Zusätzlich müssen die ausgelesenen Daten verarbeitet werden. Die Implementierung der Oberfläche ist in allen Alternativen der gleiche Aufwand und wird nicht weiter betrachtet. Die Alternative 4 hat durch die Nutzung von Node-Red einen höheren Implementierungsaufwand, da die Einstellungen im Steuerungsprogramm an jeder SPS angepasst werden müssen. Jedoch ist das nur eine kleine Änderung, weshalb diese Alternative mit (o) bewertet wird. Im Vergleich zu der Alternative 4 muss bei der Alternative 3 die OPC UA Nutzung in den Einstellungen aller Steuerungen eingestellt werden. Dafür muss die Firmware von 1.8 auf 2.0 aktualisiert werden, was einen

höheren Aufwand bedeutet. Diese Alternative wird deshalb mit (-) bewertet. Die Alternative 1 wird aufgrund der Implementierung der zusätzlichen Sensorik am schlechtesten, mit (- -), bewertet.

Bei der Trennung der Übertragungswege der Rückmeldedaten und Steuerungsdaten (Anforderung 3) wird die Alternative 1 am besten bewertet (++), da diese keinerlei Bezug zu der Steuerung hat, sondern lediglich teilweise die gleichen Sensordaten verwendet. „Gut“ (+) wird diese Anforderung von Alternative 3 erfüllt. Die Steuerungs- und Rückmeldedaten werden nicht durch Hardware getrennt. Allerdings werden die Daten über OPC UA übertragen, was, wie in den Grundlagen (Unterkapitel 2.2) beschrieben, zu einer getrennten Übertragung durch unterschiedliche Protokolle führt. Die Alternativen 2 und 4 erfüllen die Anforderung „eher weniger“(-). Die Daten werden ohne Änderung der Übertragungsart oder ähnliches ausgelesen.

Die Anforderung 4, keine Änderungen am Steuerungsprogramm vorzunehmen, wird durch die komplette Trennung von Alternative 1 voll erfüllt (++). Zur Umsetzung der Alternative 2 und 4 müssen die Steuerungsprogramme um einen Funktionsbaustein, der die Zu- und Abgänge der Schnitte zählt, ergänzt werden. Für Alternative 4 muss zusätzlich eine kleine Änderung in den Einstellungen vorgenommen werden, für Alternative 2 in dem zusätzlichen Funktionsbaustein der Baustein DataLog eingebaut werden. Das Steuerungsprogramm an sich bleibt bei allen Alternativen von den Änderungen unberührt. Die Alternativen 2 und 4 erfüllen die Anforderung „gut“ (+). „Zum Teil“ (o) wird die Anforderung von der Alternative 3 erfüllt. Dieses setzt zusätzlich zu dem, wie auch bei Alternative 2 und 4, benötigten Funktionsbaustein eine Aktualisierung der Firmware voraus. Dies soll laut Siemens keine Auswirkungen auf das Programm haben, allerdings kann dies nicht garantiert werden.

Die Alternative 2 erfüllt die Anforderung 5, keine Daten zu verpassen, am besten (++). Die Zu- und Abgänge werden auf der Steuerung gezählt und bei einer Änderung in die Excel-datei geschrieben. Die Alternativen 3 und 4 erfüllen diese Anforderung „gut“ (+). Wenn die Abfrage der Daten zu lange dauert, können einzelne Zeitstempel und Datenänderungen verloren gehen. Hier muss dafür gesorgt werden, dass die Abfrage möglichst schnell stattfindet. Sollten doch Daten verloren gehen, ist durch die Vorverarbeitung in der Steuerung der Wert der Zähler allerdings immer korrekt. Es kann also nur der Zeitpunkt der Änderung, nicht aber die Änderung an sich verloren gehen. Die Alternative 1 erfüllt die Anforderung zurzeit „eher nicht“ (-). Durch die geringe Verlässlichkeit der Sensoren und das Fehlerpotenzial beim Auslesen der gespeicherten unverarbeiteten Daten können Zustandsänderungen und damit Änderungen der Daten verpasst werden.

Die Anforderung 6, möglichst vorhandene Mittel zu nutzen, wird von der Alternative 2 am besten eingehalten (++). Hier wird keine Zusatzhardware benötigt. Die Alternativen 3 und 4 benötigen jeweils ein System, mit dem eine Datenbank und ein Webserver eingerichtet werden können. Außerdem muss auf diesem für Alternative 4 Node-Red implementiert sein.

Aus diesem Grund werden die Alternativen mit (+) bewertet. Die Alternative 1 wird aufgrund ihrer zusätzlichen Sensorik am schlechtesten, mit (- -) bewertet.

Die Speicherung der Daten für ein Jahr (Anforderung 8) gelingt den Alternativen 3 und 4 mit der Datenbank am besten (++). Die Daten von einem Jahr in einer Exceldatei zu speichern, Alternative 1 und 2, wird als „gut“ (+) bewertet.

Zur Erfüllung der Anforderung 9, der übersichtlichen Speicherung, sind durch die Datenbanknutzung ebenfalls die Alternativen 3 und 4 am besten bewertet (++). Die Alternativen 1 und 2 erfüllen diese Anforderung „zum Teil“ (o).

Die beste Alternative zur Nutzung von Industrielösungen (Anforderung 10) ist die Alternative 3, welche mit (++) bewertet wird. OPC UA wird in der Industrie häufig angewendet. Mit (+) bewertet werden die Alternativen 2 und 4. Die Alternative 2 wird von Siemens vorgeschlagen, die Alternative 4 basiert auf dem Gedanken Internet of Things (IoT), was beides für einen gewissen Industriestandard spricht. Die Alternative 1 wird als „zum Teil“(o) bewertet, da die Implementierung der Sensoren, wenn nicht vorhanden, aufwendig ist und die Daten nicht von der Steuerung vorverarbeitet werden können.

Um sich für eine Alternative zu entscheiden, werden die Anforderungen mit dem Ergebnis der Bewertung sowie dem Gewichtungsfaktor in Tabelle 4.4 für jede Alternative aufgelistet.

Tabelle 4.4.: Entscheidungstabelle Datenbereitstellung und -verarbeitung

Anforderung	Gewichtungs- faktor	Alternative			
		1	2	3	4
1: Schnittstellen gering halten	1	++	+	o	o
2: Implementierungsaufwand gering halten	1	- -	+	-	o
3: Trennung der Übertragungswege	4	++	-	+	-
4: Änderung des Steuerungsprogramms gering	3	++	+	o	+
5: Keine Daten verpassen	5	-	++	+	+
6: Vorhandene Mittel nutzen	1	- -	++	+	+
8: Speicherung ein Jahr	3	+	+	++	++
9: Übersichtliche Speicherung	2	o	o	++	++
10: Industrielösungen	3	o	+	++	+
Ergebnis		79	88	94	87

Die Addition der gewichteten Bewertung ergibt das Gesamtergebnis der entsprechenden Alternative und ist in der untersten Zeile dargestellt. Die Ergebnisse liegen alle im selben Bereich und es hebt sich keine Alternative besonders hervor. Dies liegt daran, dass die Alternativen vorselektiert wurden und nur sinnvolle und technisch realisierbare Alternativen

beschrieben werden. Die Alternative 3 hebt sich mit 6 Punkten ab. Zusätzlich ist, wie in den Grundlagen (Unterkapitel 2.2) beschrieben, OPC UA in der Industrie der Standard zur Übertragung von Rückmeldedaten. Aus diesem Grund wird die Alternative 3 umgesetzt.

#### 4.2.5. Weitere zu beachtende Überlegungen

In diesem Abschnitt werden für die umzusetzende Alternative 3 zusätzliche Entscheidungen getroffen. Zunächst muss entschieden werden, mit welcher Programmiersprache das Empfangen, Verarbeiten und Abspeichern der Daten realisiert werden soll und wie dieses Programm aufzubauen ist. Außerdem muss entschieden werden, wie die Daten der Steuerungen über WLAN übertragen werden können. Zum Schluss wird sich noch für eine Dokumentenstruktur der Webseite entschieden.

#### Programm zum Empfangen, Verarbeiten und Speichern der Daten

Für das Empfangen der Daten sowie das Verarbeiten und Abspeichern fällt die Entscheidung auf Python. Python ist sowohl unter Windows, als auch unter Linux nutzbar und umfasst eine gute Dokumentation. Zusätzlich beinhaltet sie Module zur Erstellung eines OPC UA Clients und der Speicherung von Daten in einer Datenbank.

Das Empfangen, Verarbeiten und Speichern der Daten soll in einem Pythonprogramm stattfinden. Dieses kann entweder zyklisch die Daten abfragen und gegebenenfalls verarbeiten und speichern, wie in Abbildung 4.13 gezeigt, oder zyklisch die Daten abfragen und bei einer Änderung einen Subprozess erstellen, welcher die Daten verarbeitet und anschließend abspeichert, wie in Abbildung 4.14 dargestellt. Das Verarbeiten der Daten und das Schreiben dieser in die Datenbank verlangsamt das Programm und damit die Abfrage der OPC UA Daten. Um die Anforderung 5, keine Daten zu verpassen, einhalten zu können, sollte das Pythonprogramm zur Verarbeitung und Speicherung der Datensätze einen Subprozess erstellen, um die Abfragefrequenz zu erhöhen. Die erstellten Subprozesse werden dann, wie in Abbildung 4.14 dargestellt, immer dann gestartet, wenn kein anderer Subprozess aktiv ist.

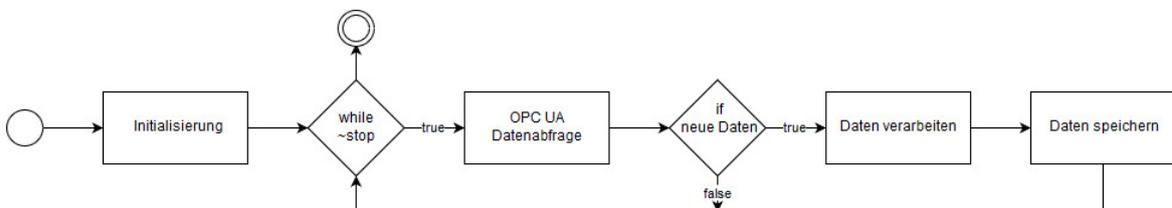


Abbildung 4.13.: Ablaufdiagramm serielles Pythonprogramm

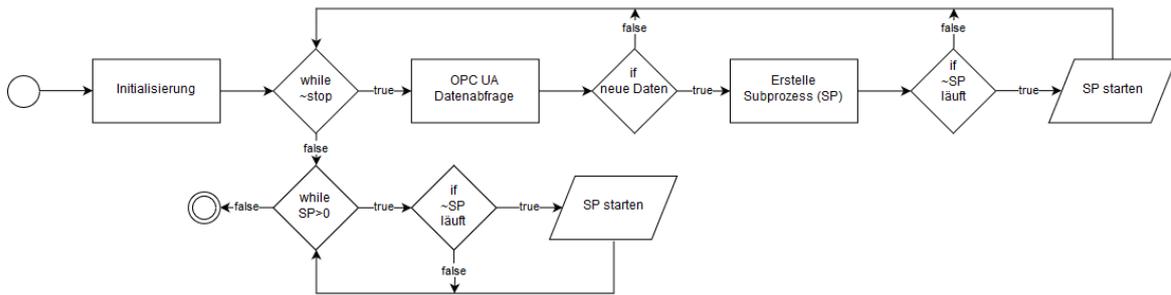


Abbildung 4.14.: Ablaufdiagramm Pythonprogramm mit Subprozess (SP)

## WLAN-Übertragung

Zur Datenübertragung mittels WLAN gibt es zwei Alternativen. Zum einen kann jede Steuerung mit einer WLAN-Schnittstelle ausgestattet werden und die eigenen Daten an den Server schicken (Alternative 1). Zum anderen können die Daten von nur einer Profinet Schnittstelle, z.B. von Steuerung 10 über eine WLAN-Schnittstelle übertragen werden (Alternative 2). Die Abbildungen 4.15 und 4.16 zeigen die zwei Alternativen.

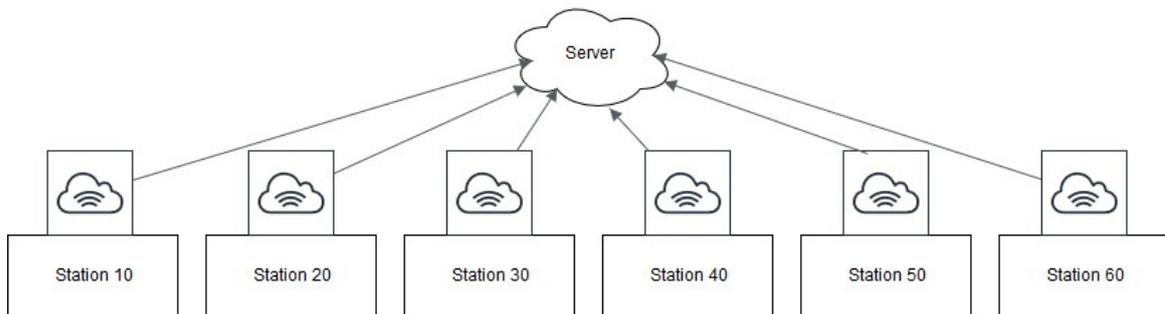


Abbildung 4.15.: Alternative 1: Jede Steuerung erhält ihre eigene WLAN-Schnittstelle

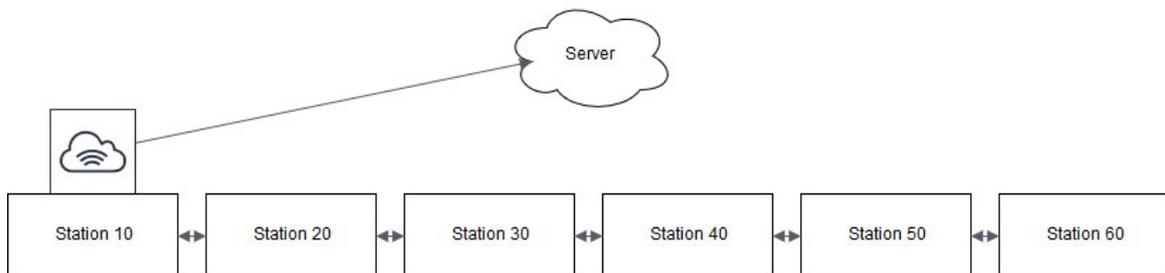


Abbildung 4.16.: Alternative 2: Es existiert eine WLAN-Schnittstelle zur Datenübertragung

Der Vorteil der Alternative 1 ist die Trennung der Übertragungswege der Rückmelde- und Steuerungsdaten. Durch die eigenen Schnittstellen kann jede Steuerung ihre eigenen Daten an den Server schicken und ist komplett unabhängig von der Profinet Verbindung unter den Stationen. Durch den Netzwerkaufbau der Anlage ist diese Alternative aber nur mit einer zusätzlichen Profinet Schnittstelle realisierbar, da sich die Schnittstellen zurzeit alle im selben Netzwerk befinden und damit nicht nur die Daten einer Steuerung, sondern alle Daten an jeder Schnittstelle anliegen. Dies führt zu Überschneidungen der Daten und verloren gehenden Paketen beim Empfangen der Daten. Eine zweite Schnittstelle einzubauen ist kostenintensiv und der Implementierungsaufwand ist groß. Die Trennung der Steuerungs- und Rückmeldedaten wird bei der Alternative 2 weniger gut eingehalten als bei der Alternative 1, ist aber durch die Übertragung mittels WLAN immer noch erfüllt. Ein Vorteil ist, dass eine WLAN-Bridge zur Übertragung genügt und keine weitere Hardware angeschafft und implementiert werden muss. Außerdem kann es so nicht zu Überschneidungen und verloren gehenden Paketen kommen. Aus diesem Grund wird sich für die Alternative 2 mit einer WLAN-Verbindung entschieden.

### **Dokumentenstruktur der Webseite**

Wie in Abschnitt 4.1.4 beschrieben werden sechs verschiedene Grundwebseiten benötigt. Zum einen werden die zwei Startwebseiten, die entweder zu den aktuellen oder historischen Werten der mobilen oder Leitstandwebseite führen, benötigt. Zum anderen sind jeweils zwei Webseiten für die aktuellen und historischen Werten der mobilen und Leitstandwebseite erforderlich. Zusätzlich ist zu entscheiden, wie die verschiedenen Inhalte der Grundwebseiten erstellt und eingebunden werden sollen. Der erste Ansatz, eine Webseite zu programmieren, ist, jeden darzustellenden Inhalt in einer eigenen HTML-Datei abzulegen und durch Verweise zwischen diesen zu wechseln. Der Nachteil in diesem Vorgehen ist, dass die gesamte Webseite bei jedem Inhaltswechsel neu geladen wird, obwohl sich nicht das Layout, sondern nur der darzustellende Inhalt ändert. Die HTML-Dateien unterscheiden sich also nur im Inhalt. Das Layout auf allen Seiten aktuell zu halten, ist sehr aufwendig. Um dies zu verhindern, kann mittels JavaScript (JS) bei Aktivierung eines Buttons ein anderer Inhalt dargestellt werden, ohne die komplette Seite neu zu laden. Für diese Alternative werden dann nur die sechs HTML-Dateien mit den Grundwebseiten benötigt und zwei JS-Dateien, die den Inhalt für die aktuellen und historischen Daten je nach Anforderung ändern. Wegen des Vorteils, die Anzahl der HTML-Dateien und damit die Pflege der Webseiten gering zu halten, wird sich für dieses Vorgehen entschieden.

## 4.3. Hardware

Für das Gesamtsystem wird zum einen ein Server zur Sammlung und Verarbeitung der Daten sowie zur Speicherung dieser und zum Betreiben eines Webservers benötigt. Zur Darstellung der Webseite werden ein Anzeigesystem mit einer Fläche von ca. 1m x 2m und ein Endgerät mit einer Bildschirmdiagonale von ca. 13 Zoll benötigt. Außerdem wird für das Konzept 2, die Übertragung der Daten mit einer WLAN-Verbindung, ein Router und eine WLAN-Bridge benötigt. In diesem Unterkapitel werden die Alternativen für die Hardware beschrieben und die Entscheidung für die jeweilige Alternative begründet. Die gesamte Hardware darf hierbei das Gesamtbudget von 3.500€ nicht überschreiten.

### 4.3.1. Server

Für das Empfangen, Verarbeiten und Speichern der Daten sowie das Implementieren eines Webservers wird ein Server benötigt, der diesen Anforderungen entspricht. Eine kostengünstige und leistungsstarke Alternative ist der Raspberry Pi 3B+, im Folgenden Raspberry genannt. Ein Raspberry Starterset ist bei Reichelt für 64,50€ erhältlich [1]. Auf diesem ist Python vorinstalliert, und Webserver und Datenbank können installiert werden. Durch die große Community stehen bei auftretenden Problemen im Umgang viele Lösungsansätze zur Verfügung. Zusätzlich kann der Raspberry als Router eingerichtet werden. Durch den WLAN-Standard AC kann ein 2,4-GHz-Netzwerk oder bei Bedarf ein 5-GHz-Netzwerk aufgespannt werden. Ein vergleichbarer Windowsserver ist bei gleichem Anforderungsprofil kostenintensiver. Aus diesem Grund wird sich für den Raspberry entschieden.

### 4.3.2. Anzeigesystem

Benötigt wird ein Anzeigesystem, welches die Leitstandwebseite darstellt. Im Folgenden werden die Anforderungen vorgestellt. Anschließend werden zwei alternative Systeme dargestellt und bewertet und zum Schluss mit Hilfe einer Entscheidungstabelle eine Entscheidung getroffen.

#### Anforderungen

1. Das System soll einem Leitstandsystem nachempfunden sein.
2. Eine Wandmontage muss möglich sein.
3. Der Implementierungsaufwand soll gering gehalten werden.

4. Fehleranfälligkeit soll minimiert werden.
5. Es soll eine Fläche von ca. 1 m x 2 m (88 Zoll) ausgefüllt werden.
6. Die Anzeige soll sowohl aus 50cm wie auch aus 8m Entfernung gut erkennbar sein.
7. Das System soll ein großes Vollbild und vier kleine Bilder darstellen können.

### **Alternative 1**

Eine Alternative, dieses System aufzubauen, besteht aus vier 40 Zoll Monitoren und einem PC mit einer Grafikkarte mit vier Ausgängen und einer Videowall Funktion.

Der Vorteil dieser Alternative ist, dass die zur Verfügung stehende Fläche ausgenutzt wird. Außerdem ähnelt dieses dem Aussehen eines Leitstandes.

Der Nachteil ist der Implementierungsaufwand sowie die Fehleranfälligkeit dieses Systems. Die vier Monitore müssen alle nebeneinander mit möglichst wenig Abstand an der Wand befestigt werden, was einen hohen Implementierungsaufwand bedeutet. Außerdem können alle vier Monitore kaputt gehen und wenn einer davon defekt ist und dieser nicht mehr erhältlich ist, müssen alle vier Monitore ausgetauscht werden, was die Instandhaltungskosten erhöht. Die Anschaffungskosten dieser Alternative liegen bei ca. 2700-3200€. In diesem Preissegment sind die Stege der Bildschirme allerdings recht breit, was die Betrachtung im Vollbild negativ beeinflusst.

### **Alternative 2**

Die zweite Alternative ist, einen 75 Zoll großen Monitor oder Fernseher zu nutzen und diesen in vier Bereiche zu unterteilen. Die Webseite für den Leitstand unterteilt die Oberfläche schon in vier Abschnitte und weitere Software kann durch die Windowsfunktion in vier Bereichen dargestellt werden. Ein PC mit einer 4K Grafikkarte wird als Anzeigequelle genutzt. An diesen kann zusätzlich ein zweiter kleinerer Monitor angeschlossen werden, damit die Bedienung der Webseite nicht über den großen Monitor vorgenommen werden muss.

Der Vorteil ist, dass keine Ränder das Vollbild stören und der Fernseher mit weniger Aufwand an die Wand montiert werden kann, da nur ein Gerät aufgehängt wird. Die Reduzierung der Komponenten verringert die Fehleranfälligkeit, da nicht so viele Komponenten einen Defekt haben können. Die Investitionskosten für einen 4K 75 Zoll Fernseher liegen bei ca. 1500€. Wird der PC wie in Alternative 1 mit 1000€ angenommen und ca. 40€ für eine Wandhalterung addiert, so ergibt sich eine Gesamtinvestition von 2540€. Der Nachteil dieser Alternative ist, dass die Gesamtfläche nicht komplett genutzt wird. Ein Fernseher mit einer Bildschirmdiagonale von 75 Zoll entspricht einer Fläche von 1 m x 1,70m.

### Vergleich und Entscheidung

Die in Abschnitt 4.3.2 definierten Anforderungen werden von beiden Alternativen erfüllt. Wird die Anforderung 1, das Nachempfinden eines Leitstandsystems, betrachtet, müssen in der Alternative 2 die verschiedenen Fenster von Hand auf die gewünschte Größe eingestellt werden, weshalb diese die Anforderung „gut“ (+) erfüllt, während die Alternative 1 die Anforderung „sehr gut“ (++) erfüllt. Die Montage eines großen Fernsehers (Alternative 2) ist weniger aufwendig, als vier Monitore zu montieren. Die Alternative 1 erfüllt die Anforderung 2 somit „gut“ (+), während die Alternative 2 diese „sehr gut“ (++) erfüllt. Der Implementierungsaufwand sowie die Fehleranfälligkeit (Anforderung 3 und 4) ist durch die Wandmontage, aber auch durch die vier Datenkabel und vier Spannungsversorgungen sowie daraus resultierend viele Komponenten in Alternative 1 größer und wird mit „gut“ (+) bewertet, als in der Alternative 2, welche mit „sehr gut“ (++) bewertet wird. Die Ausnutzung der zur Verfügung stehenden Fläche (Anforderung 5) wird von der Alternative 1 besser erfüllt und mit „sehr gut“ (++) bewertet. Die Alternative 2 erfüllt die Anforderung „gut“ (+). Die Anforderung 6 wird von beiden Alternativen „sehr gut“ (++) erfüllt. Zum Schluss wird die Anforderung 7, ein Vollbild und vier einzelne Anwendungen und Bilder darzustellen, bewertet. Die Alternative 2 erfüllt diese Anforderung „sehr gut“ (++)). Die Alternative 1 erfüllt sie durch die störenden Stege in der Mitte des Vollbildes „gut“ (+).

Zur Entscheidung für eine Alternative werden sie mit den bewerteten Anforderungen in einer Entscheidungstabelle 4.5 dargestellt und sich anschließend für eine Alternative entschieden. Die Anforderungen an das System werden als gleich wichtig angesehen und aus diesem Grund wird keine Gewichtung vorgenommen. Die Erfüllung der Anforderungen wird mit 5(++) „sehr gut“ und 4(+) „gut“ bewertet.

Tabelle 4.5.: Entscheidungstabelle Anzeigesystem

Anforderung	Alternative	
	1	2
1: Leitstand ähnliches Aussehen	++	+
2: Wandmontage möglich	+	++
3: Implementierungsaufwand gering	+	++
4: Fehleranfälligkeit minimiert	+	++
5: Fläche von 1m x 2m	++	+
6: Erkennbar aus 50cm und 8m	++	++
7: Ein Vollbild oder 4 einzelne Bilder	+	++
Ergebnis	31	33

Die Tabelle 4.5 zeigt, dass die Alternative 1 mit vier Bildschirmen weniger Vorteile bringt als die Alternative 2 mit einem großen Bildschirm. Die Alternative 2 wird also umgesetzt. Es wird sich für den Fernseher LG Electronics 75UK6200 entschieden, der mit ca. 1480€ im Budget liegt. Als Quelle dieses 4K Fernsehers wird sich für einen Fujitsu Q 957, von dem Rahmenvertragspartner der HAW, für ca. 1050€ entschieden. Die Grafikkarte dieses PC's (Intel HD Graphics 630) liefert die gewünschte 4K Auflösung. Weiter handelt es sich bei der Ausstattung um einen Office PC mit WLAN, um sich mit dem vom Raspberry aufgespannten WLAN verbinden zu können und damit die Webseite auf dem Fernseher darzustellen. Für Wandhalterung und HDMI Highspeed Kabel zur Übertragung der 4K Auflösung werden ca. 65€ investiert. Die Gesamtkosten für das Anzeigesystem beträgt somit ca. 2595€.

### 4.3.3. WLAN-Bridge

Wie in Abschnitt 4.2.5 beschrieben wird eine WLAN-Bridge zur Verbindung mit dem WLAN der Modellfabrik benötigt. Hier stehen zwei Modelle zur Auswahl. Zum einen das WLAN-Modul RP-AC51 von ASUS und zum anderen ein Accesspoint „SCALANCE W788-1 RJ45“ von Siemens. Im Folgenden werden die zwei Alternativen beschrieben und die Entscheidung begründet.

#### Alternative 1

Das WLAN-Modul von ASUS hat die Möglichkeit, dieses im WLAN-Bridge Modus zu nutzen und ist eine Plug and Play Alternative. Das Modul ist in eine Steckdose, welche im Schaltschrank der Station 10 installiert werden muss, einzustecken, über einen freien Ethernet Port im Schaltschrank mit dem Ethernet Netzwerk der Modellfabrik zu verbinden und durch den internen Webserver in kurzen Schritten zu konfigurieren. Das Modul ist mit ca. 45€ kostengünstig und die Lieferzeit entspricht zwei Tage. Es kann sowohl im 2,4GHz wie auch im 5GHz senden und ist damit gut für diese Anwendung geeignet.

#### Alternative 2

Der Accesspoint von Siemens benötigt zusätzliche Hardware wie eine omnidirektionale Stabantenne, Leitung und Hutschienenadapter. Der Implementierungsaufwand dieser Alternative ist durch die vermehrten Komponenten höher als der in Alternative 1. Da diese Alternative in der Industrie für größere Anwendungen konzipiert wurde, dauert die Konfiguration länger als bei der Alternative 1. Dass diese Alternative eher für größere Industrielösungen ausgelegt wurde, zeigt sich auch im Preis von ca. 670€.

## Fazit

Für die hier beschriebenen Bedürfnisse ist die Alternative 2 überdimensioniert. Deshalb wird sich für Alternative 1 entschieden.

### 4.3.4. Mobiles Anzeigesystem

Als mobiles Anzeigesystem wird ein Tablet mit einer Bildschirmdiagonalen von 13 Zoll benötigt. Für dieses stehen 400€ des Gesamtbudgets zur Verfügung. Die Auswahl an 13 Zoll Tablets ist sehr begrenzt. Es wird sich für das 2 in 1 Gerät "TrekStor® PrimeTab T13B-PO" mit einem Windowsbetriebssystem für ca. 400€ entschieden. Dieses entspricht der Vorgabe von 13 Zoll, liegt im Budget und kann durch das Windowsbetriebssystem wie ein normaler PC genutzt werden. Dies bietet den Vorteil, dass z.B. der VNC Viewer installiert werden kann, um sich auf den Raspberry zu schalten. So kann das Tablet neben der Darstellung der Rückmeldedaten auch als Wartungs- bzw Reparatur-Hilfsmittel genutzt werden.

## 4.4. Zusammenfassung des Gesamtkonzeptes

Zu realisieren ist ein System zur Visualisierung der Rückmeldedaten der Modellfabrik. Hierzu ist der LG Electronics 75UK6200 Fernseher mit einer Bildschirmdiagonale von 75 Zoll in der Modellfabrik an die Wand zu montieren. Dieser muss über einen PC mit dem über den Raspberry aufgespannten WLAN der Modellfabrik verbunden werden. Auf dem Raspberry sind die Datenbank und der Webserver aufzusetzen und die Rückmeldedaten mittels OPC UA Client zu empfangen. Für das Konzept 1, die Nutzung des vorhandenen Bussystems, soll die Station 10 über eine Ethernet-Verbindung mit dem Raspberry verbunden werden. Nach erfolgreichen Tests des Gesamtsystems soll das Konzept 2, die WLAN-Alternative, umgesetzt werden. Dafür wird die Übertragung mit dem WLAN-Modul RP-AC51 von ASUS, welches als WLAN-Bridge betrieben wird, verwendet. Durch OPC UA sollen die auf den Steuerungen gesammelten Rückmeldedaten an den Raspberry gesendet werden. Dieser fragt die OPC UA Daten periodisch mit einem Pythonprogramm (OPC UA Client) ab. Das Pythonprogramm soll die Daten empfangen und diese in einem Subprozess verarbeiten und in der Datenbank speichern. Es sind vier verschiedene Webseiten zu implementieren, die die Rückmeldedaten in Durchlaufdiagrammen und in tabellarischer Form darstellen. Gezeigt werden das Durchlaufdiagramm mit Tabelle der Gesamtanlage, die Detaildurchlaufdiagramme mit dazugehörigen Diagrammen und eine Übersicht der Gesamtanlage, aufgeteilt in vier und sechs Schnitte nur mit Durchlaufdiagrammen, um den Verlauf eines Produktes durch die

verschiedenen Schnitte auf einen Blick verfolgen zu können. Zusätzlich zu den vier Darstellungswebseiten sollen noch zwei Startwebseiten mit gleichem Design erstellt werden. Die eine führt zu den Daten der mobilen Seite und die andere zu den Daten der Leitstandwebseite. Zusätzlich ist die mobile Version der Webseite auf dem TrekStor PrimeTab darzustellen, um diese testen zu können. Der Gesamtaufbau ist in Abbildung 4.17 schematisch dargestellt.

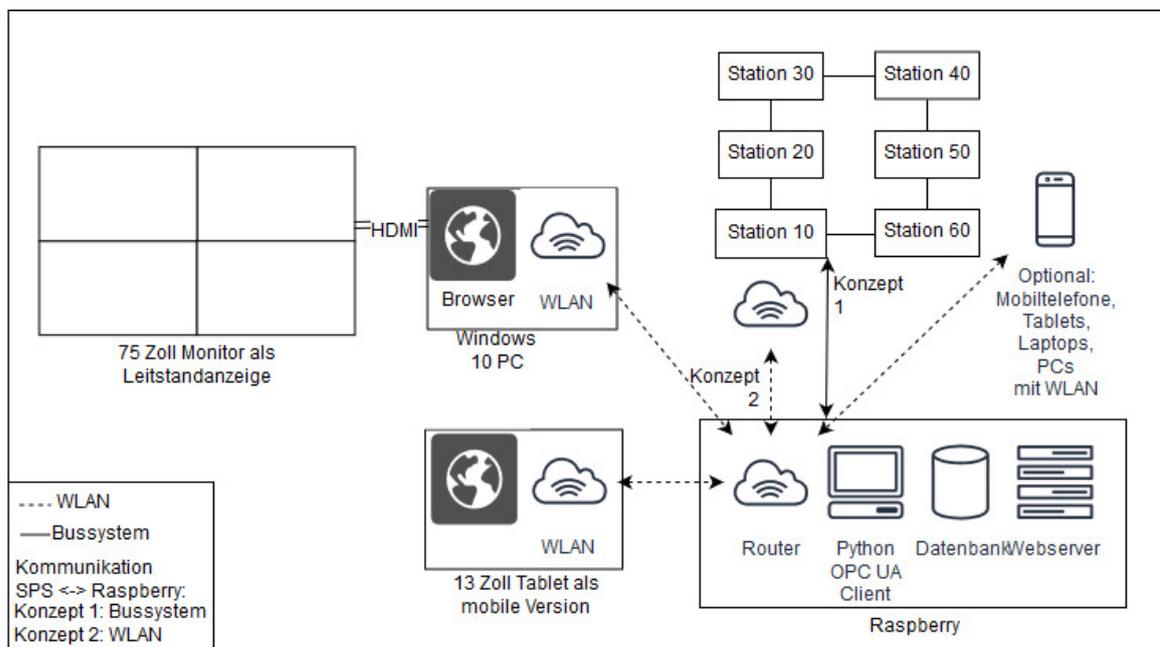


Abbildung 4.17.: Schematische Darstellung des Gesamtaufbaus

# 5. Realisierung

In diesem Kapitel wird das in der Konzeption (Kapitel 4) beschriebene Gesamtkonzept umgesetzt. Hierfür wird zunächst das Konzept 1 mit der Nutzung des vorhandenen Bussystems entwickelt und anschließend der Übertragungsweg zu WLAN geändert (Konzept 2). Zunächst werden die geltenden Annahmen der Modellfabrik zur Realisierung des Konzeptes dargestellt. Danach wird auf die Realisierung der Hardware eingegangen. Abschließend wird die Datenbereitstellung und -verarbeitung beschrieben und die Webseite realisiert.

## 5.1. Annahmen zum Betrieb der Modellfabrik

Zum einwandfreien Betrieb des in dieser Arbeit realisierten Konzeptes werden die folgenden Annahmen zum Betrieb der Modellfabrik vorausgesetzt:

- Alle Hardwarekomponenten arbeiten korrekt.
- Die vorhandene Sensorik wird nur durch (Teil-)Produkte ausgelöst. Ein manuelles Auslösen mittels anderer Gegenstände wird ausgeschlossen. Nur die Sensoren B3 und B5 in Station 20 müssen je Fertigprodukt einmal manuell ausgelöst werden, um es aus dem Prozess zu entnehmen.
- Die Inbetriebnahme erfolgt wie in der Anleitung beschrieben. Die Anleitung ist im Anhang A.2 einzusehen. Für die Steuerungsvariante wird die Variante 4 gewählt und der 6-Achsroboter auf eine Geschwindigkeit von 71% eingestellt.
- Alle Komponenten befinden sich nach dem Schritt 1 der Anleitung (Einschalten der Anlage durch den Hauptschalter) in ihrem Initialzustand.
- In allen Zufuhr-Lagern (Unterbaulager, Platinenlager, Oberbaulager) befinden sich stets ausreichend viele Bauteile zur Bearbeitung der Aufträge.
- Der manuelle Eingriff in die Anlage wird nur am Handarbeitsplatz und bei der Entnahme der Fertigprodukte zugelassen.
- Während der Produktion werden die Türen des Schutzkäfigs nicht geöffnet.

- Die Rückmeldedatenerfassung, sprich das Pythonprogramm, wird während der Produktion nicht gestoppt.
- Nach der Eingabe der Aufträge mittels Touchpad wird mit der erneuten Eingabe gewartet, bis alle geforderten Aufträge abgearbeitet sind.

## 5.2. Hardware

Dieses Unterkapitel beschreibt die Einrichtung und Implementierung der Hardware. Dafür wird zunächst auf die Einrichtung des Raspberrys eingegangen, bevor die Konfiguration des WLAN-Moduls und abschließend die Einrichtung des Anzeigesystems beschrieben wird. Im Laufe dieses Unterkapitels werden Anmeldenamen und Passwörter vergeben, die gesammelt im Anhang [A.3](#) zu finden sind.

### 5.2.1. Raspberry Pi einrichten

Auf dem Raspberry wird zunächst das Betriebssystem *Raspbian GNU/Linux 9 (stretch)* installiert. Anschließend wird für Wartungszwecke die VNC Option aktiviert. Damit kann sich mit einem Rechner, der sich im gleichen Netzwerk befindet und auf dem *VNC Viewer* installiert ist, auf den Raspberry geschaltet werden. Dieser kann so ohne Bildschirm und Tastatur genutzt werden. Um den Raspberry für diese Arbeit nutzen zu können, müssen Änderungen an den Netzwerkeinstellungen vorgenommen werden. Außerdem sind ein Webserver und eine Datenbank zu installieren und einzurichten.

#### Netzwerkeinstellungen

Damit der Raspberry sich mit den Steuerungen verbinden kann und immer unter der gleichen IP-Adresse zu finden ist, muss eine statische IP-Adresse des Raspberrys mittels DHCP-Client eingerichtet werden. Die vergebene IP-Adresse für den Raspberry wird auf *192.168.33.151* festgelegt. Für das Konzept 1, die Verbindung mittels vorhandenem Bussystem, wird diese feste IP in dem eth0 Interface eingetragen (Vergleiche [\[21\]](#)). Mit dieser Änderung sind die Netzwerkeinstellungen für das Konzept 1 abgeschlossen. Für das Konzept 2, die Nutzung des Raspberrys als WLAN-Router, werden zunächst die Einstellungen aus Konzept 1 im DHCP-Client rückgängig gemacht. Anschließend wird, der Anleitung [\[11\]](#) folgend, der Raspberry als WLAN-Router eingerichtet. Als IP-Adresse wird erneut die *192.168.33.151* verwendet. Das WLAN wird als 2,4-GHz-Netzwerk aufgespannt. Sowohl der Raspberry, als auch alle anderen verwendeten Hardwarekomponenten können im Falle einer Überlastung

des Frequenzbandes auf 5GHz umgestellt werden. Für den Namen des WLANs und das Passwort wurde

SSID: **Modellfabrik**

WPA2-Schlüssel: **atLabor484**

gewählt.

### Webserver und Datenbank

Die Installation des Webserver, der Datenbank, PHP und phpMyAdmin folgt den auf [5] beschriebenen Schritten von „Den Webserver installieren & aktivieren“ bis einschließlich „phpMyAdmin installieren“. Als Webserver wird der verbreitete Apache 2 Webserver auf dem Raspberry installiert. Anschließend wird die Datenbank MySQL installiert, der Benutzer: **root** mit dem Passwort: **atlabor** eingerichtet und zum Schluss PHP sowie phpMyAdmin installiert. Die Nutzung der MySQL Datenbank erfolgt nun über die phpMyAdmin Seite, die über [192.168.33.151/phpmyadmin](http://192.168.33.151/phpmyadmin) erreichbar ist. Hier wird eine Datenbank namens *Rückmeldedaten* erstellt. Für diese Datenbank wird ein weiterer Nutzer berechtigt. Loggt man sich bei phpMyAdmin als Nutzer: **atlabor** mit dem Passwort: **at** ein, so wird diese Datenbank sichtbar und kann bearbeitet werden. Alle weiteren Datenbanken, die während der Installation erstellt werden, werden unter diesem Nutzer nicht angezeigt. Für die vorliegende Arbeit ist dieser Zugang ausreichend und wird im Folgenden, wenn nicht anders beschrieben, genutzt.

#### 5.2.2. Einrichten der WLAN-Bridge

Das WLAN-Modul RP-AC51 von ASUS lässt sich nach der Bedienungsanleitung [18] einrichten. Hierfür wird lediglich ein Ethernet-Kabel und eine Steckdose benötigt. Die Steckdose wird in den Schaltschrank der Station 10 verbaut. Das WLAN-Modul wird an einem freien Ethernet-Port der Station 10 angeschlossen und in die Steckdose gesteckt. Mit dem Webinterface des WLAN-Moduls lässt es sich wie in der Bedienungsanleitung [18] beschrieben konfigurieren.

#### 5.2.3. Anzeigesysteme einrichten

Zur Anzeige gehören das Tablet zur Ansicht der mobilen Version der Webseite sowie der fest installierte Windows 10 PC zur Darstellung der Leitstand Webseite. Die Anzeige der Leitstandwebseite erfolgt über den an die Wand montierten Fernseher und den zusätzlichen

kleinen Monitor zur leichteren Bedienung. Das Tablet sowie der PC werden, wie im Anhang [A.4](#) beschrieben, konfiguriert und der Fernseher wird an die Wand montiert. Hier ist keine weitere Einrichtung nötig.

### 5.3. Datenbereitstellung und -verarbeitung

In diesem Unterkapitel wird das Vorgehen zur Realisierung der Konzepte aus Unterkapitel [4.2](#) beschrieben. Dafür wird zunächst auf die Datengewinnung im Steuerungsprogramm eingegangen. Anschließend wird das Empfangen und Verarbeiten der Daten beschrieben, bevor abschließend die Speicherstruktur in der Datenbank erläutert wird.

#### 5.3.1. Datengewinnung

Für die Darstellung der Durchlaufdiagramme werden die in der Konzeption (Abschnitt [4.1.2](#)) beschriebenen Alternativen 2 und 3 (vier und sechs Schnitte mit Ausgangslager) sowie die Gesamtanlage betrachtet. Aus den Abbildungen [4.2](#) und [4.3](#) lassen sich die Positionen der Zu- und Abgänge, also die Übergänge von einem in den anderen Schnitt, ablesen. Der Übergang ist immer der Zeitpunkt, zu dem das Produkt das Ausgangslager des vorherigen Arbeitsvorganges verlässt. Wie in der Anlagenbeschreibung (Unterkapitel [3.3](#)) erläutert, verfügt jeder Arbeitsvorgang über ein Ausgangslager. Die Übergänge lassen sich so durch die Sensoren und Events der einzelnen Arbeitsvorgänge definieren. Es ist z.B. der Übergang vom Schnitt Bestückung zum Schnitt Transfersystem (bei vier Schnitten) bzw. zum Schnitt Funktionsprüfung (bei sechs Schnitten) das Handling 1, das sich nach dem Ablegen des Produktes auf dem Produkträger wieder in Ausgangsposition befindet. Die ausformulierten Übergangsbedingungen der Betrachtung mittels der Detailtiefe 1 (ein Schnitt) ist in Tabelle A.1, die der Detailtiefe 2 (vier Schnitte) in Tabelle A.2 und die der Detailtiefe 3 (sechs Schnitte) in Tabelle A.3 im Anhang [A.7.1](#) aufgelistet.

Die Durchlaufzeit in den einzelnen Schnitten lässt sich durch

$$ZDL = \text{Zeitpunkt\_Abgang} - \text{Zeitpunkt\_Zugang} \quad (5.1)$$

berechnen. Dafür wird die SPS-Zeit der Station 10 mittels OPC UA zur Verfügung gestellt.

Zusätzlich zum Zu- und Abgang sowie der Durchlaufzeit werden in einigen Schnitten laut der Konzeption (Abschnitt [4.1.2](#)) weitere relevante Daten von der Steuerung gefordert. Es wird z.B. von der Station 40 die Information über das Ergebnis der Funktionsprüfung benötigt. Eine Tabelle zur Übersicht der geforderten Informationen für jeden Schnitt ist im Anhang [A.7.1](#) in A.4 dargestellt.

## Steuerungsprogramme

Zur Erfassung der zuvor beschriebenen Daten wird zu den bestehenden Steuerungsprogrammen der Stationen je ein Programmbaustein mit dem Namen `MES_DataLog` hinzugefügt. In diesem Programmbaustein werden die Übergänge von einem in einen anderen Schnitt sowie weitere relevante Rückmeldedaten mit Zählern erfasst. Gelangt ein Produkt in einen neuen Schnitt, ist die Übergangsbedingung also erfüllt, so wird der jeweilige Zähler um eins erhöht. Die Änderung der Steuerungsprogramme beschränkt sich auf diesen Baustein. So wird das Risiko, dass die Erfassung der Steuerungsdaten Einfluss auf den Betrieb der Anlage hat, verringert. Außerdem werden nur eigens erstellte Variablen mittels OPC UA übertragen und die vorhandenen nur gelesen und niemals beschrieben. Beispielhaft ist in Abbildung 5.1 der Zähler `MES_Schnitt_Funktionsprüfung` mit der Übergangsbedingung `MES_DataLog_DB.Trigger_Handling_work` dargestellt. Die Übergangsbedingung `MES_DataLog_DB.Trigger_Handling_work` beschreibt in diesem Fall die fallende Flanke des Events *Handling 1 arbeitet*.

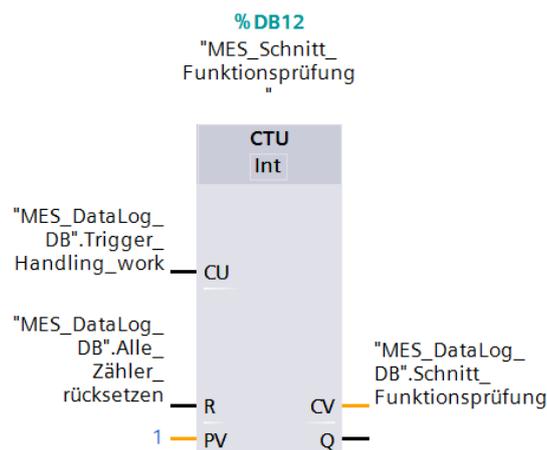


Abbildung 5.1.: Zähler des Zugangs Funktionsprüfung

Der vollständige Programmcode ist dem Anhang A.5 zu entnehmen.

Zur Datenübertragung mittels OPC UA werden alle Steuerungen auf die Firmware 2.0 aktualisiert. Im TIA-Projekt ist die Versionsnummer in der Hardwarekonfiguration einzustellen. Anschließend wird das TIA-Projekt nach der Anleitung [16] für die OPC UA Nutzung konfiguriert.

### 5.3.2. Empfangen und Verarbeiten der Daten

Zum Empfangen, Verarbeiten und Speichern der gewonnenen Daten wird, wie in der Konzeption (Abschnitt 4.2.5) entschieden, ein Pythonprogramm benötigt. Die Umsetzung wird in diesem Abschnitt beschrieben.

Zur Erstellung des Programms werden die Module *FreeOpcUa* [2] und *PyMySQL* [10] installiert und weitere Standardmodule wie z.B. *time* oder *multiprocessing* im Programm genutzt. Zusätzlich werden zu den Themengebieten Datenbank, OPC UA und Subprozess (Job) Module erstellt, die die eigenen Funktionen themenspezifisch zusammenfassen. Das Hauptprogramm ruft die benötigten Funktionen auf, sodass das Programm übersichtlich bleibt.

Bevor das Programm in seinen einzelnen Schritten beschrieben wird, werden die Aufgaben und Lösungsansätze zur Realisierung vorgestellt.

**Erfassung und Berechnung der Durchlaufzeit** Um die Durchlaufzeit eines Produktes zu erfassen, werden im Hauptprogramm neun Listen entsprechend der Schnitte angelegt. Jede Liste speichert die Zugangszeiten des eigenen Schnittes. Verlässt das Produkt den Schnitt, so ist durch den seriellen Ablauf der Produktion sichergestellt, dass die zuerst gespeicherte Zugangszeit zu der zuerst erfassten Abgangszeit gehört. Die Durchlaufzeit ergibt sich somit aus der Gleichung 5.1. Die Zu- und Abgangszeiten eines Produktes, das den Schnitt verlassen hat, ist nach der Berechnung nicht weiter vonnöten. Aus diesem Grund wird, nachdem ein Abgang eines Produktes erkannt wurde, die aktuelle Zeit für den Abgang zwischengespeichert und die zuerst gespeicherte Zugangszeit ausgelesen. Anschließend wird die Zugangszeit aus der Liste gelöscht. Mit der aktuellen Abgangszeit sowie der dazugehörigen Zugangszeit wird anschließend die Durchlaufzeit berechnet.

Das Problem bei dieser Lösung zur Berechnung der Durchlaufzeit ist, dass die Listen der Zugangszeiten bei einem Programmstopp verloren gehen. Wird das Programm also während der Produktion wieder erwartend gestoppt, so gehen die Durchlaufzeiten der Produkte, die sich in der Anlage befinden, verloren. Das Stoppen des Programmes ist während der Produktion nicht gestattet. Sollte dies dennoch passieren, so sollten die Listen mit den Zugangszeiten gespeichert werden. Die Speicherung findet für jede Liste in einer JSON-Datei statt. Diese wird bei Programmstart ausgelesen und bei Programmende werden die verbleibenden Zugangszeiten aus den Listen in die Dateien geschrieben.

**Auslesen der Daten mittels OPC UA** Zum Lesen der OPC UA Daten wird das Modul `opcua_verwaltung` erstellt. In dieses wird das installierte `FreeOpcUa` Modul importiert. Das erstellte Modul beinhaltet die Funktionen zum Verbindungsaufbau, dem Lesen der Daten und dem Verbindungsabbau, die an die Beispiele auf [2] angelehnt sind. Die gelesenen Daten werden in den Listen `Schnitt`, `Zähler` und `Zeit` gespeichert, deren Inhalte in den Tabellen A.5, A.6 und A.7 im Anhang A.7.2 einzusehen sind.

**Speichern der Daten in eine Datenbank** Zum Schreiben in die Datenbank wird das installierte Modul `PyMySQL` verwendet. Die erstellten Funktionen `verbinden`, `schreiben` und `verbindung_trennen` werden in dem Modul `Datenbank_Verwaltung` gespeichert. Informationen zum Umgang mit dem Modul `PyMySQL` können auf [10] eingesehen werden.

**Berechnung und Speicherung der Daten im Subprozess** Zur Berechnung der Nutzdaten und dem Speichern dieser in der Datenbank wird, wie in der Konzeption (Abschnitt 4.2.5) beschrieben, ein Subprozess erstellt, um die Durchlaufzeit des Hauptprogrammes gering zu halten. Das dazugehörige Ablaufdiagramm ist in Abbildung 5.2 zusehen.

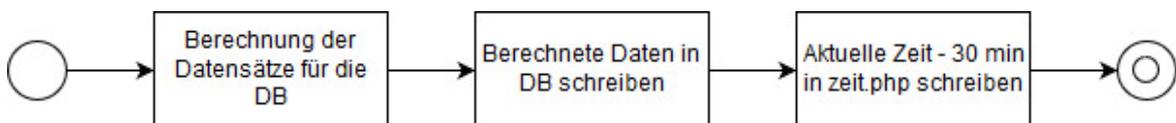


Abbildung 5.2.: Ablaufdiagramm des Subprozesses (Job)

Die Funktion `Job` aus dem gleichnamigen Modul berechnet zunächst die Datensätze für die Datenbanktabellen. Anschließend werden diese in die Datenbank geschrieben. Hierfür wird das Modul `Datenbank_Verwaltung` und die Funktion `schreiben` genutzt. Zum Schluss wird der Startzeitpunkt der Darstellung der Live-Daten auf der Webseite berechnet und in eine PHP-Datei geschrieben. Dazu werden von der aktuellen Zeit 30 Minuten abgezogen. Somit werden immer die letzten 30 Minuten der Produktion im Durchlaufdiagramm dargestellt.

**Hauptprogramm** Nachdem die grundsätzlichen Aufgaben des Programms genauer beschrieben wurden, wird nun das Ablaufdiagramm aus Abbildung 5.3 erläutert.

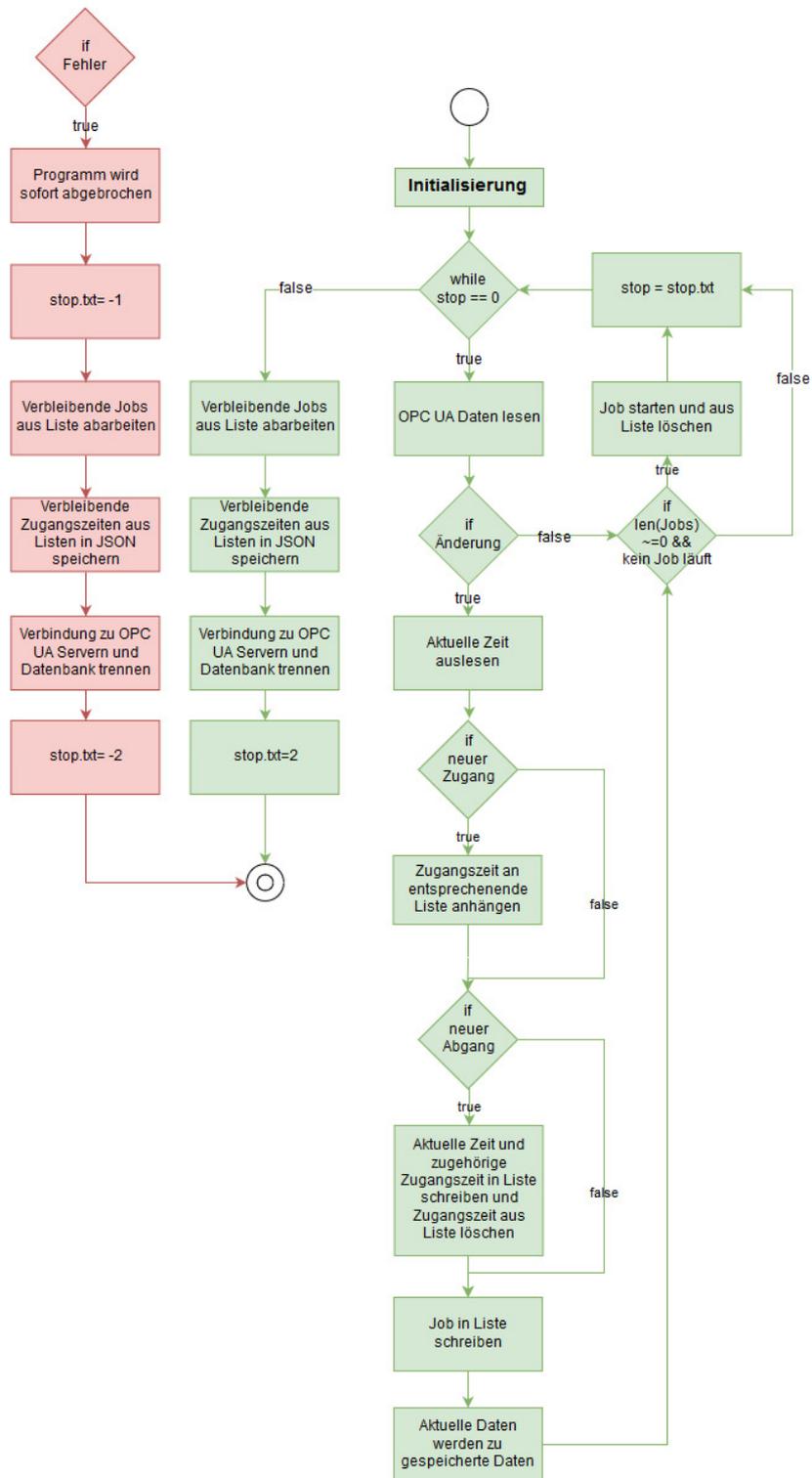


Abbildung 5.3.: Ablaufdiagramm Python Hauptprogramm

Das Programm beginnt mit einer Initialisierung. Hier werden die JSON-Dateien ausgelesen und gegebenenfalls deren Inhalte in die Listen für die Zugangszeit gespeichert. Anschließend wird eine Verbindung zu den OPC UA Servern und der Datenbank hergestellt und die OPC UA Daten gelesen und durch einen Job in die Datenbank geschrieben. Zum Schluss wird in die Datei *stop.txt* eine 0 gespeichert, um der Webseite zu signalisieren, dass das Programm läuft und auch die Variable `stop` mit 0 initialisiert.

Solange das Programm nicht gestoppt wird, läuft es zyklisch in einer *while*-Schleife und vergleicht die empfangenen OPC UA Daten mit den zuletzt gespeicherten. Wird eine Änderung der Daten festgestellt, so wird die aktuelle Zeit der Station 10 gespeichert. Hat sich ein Zugang eines Schnittes erhöht, so wird die aktuelle Zeit in die entsprechende Liste des Schnittes gespeichert. Bei einer Änderung des Abganges eines Schnittes wird zum einen die Zugangszeit aus der Liste des Schnittes an der Stelle 0 ausgelesen und zum anderen die aktuelle Zeit als Abgangszeit gespeichert. Anschließend wird der Job mit den aktuellen Daten in die Jobliste geschrieben und die aktuellen Daten gespeichert, um diese wieder mit den gelesenen Daten vergleichen zu können. Wurden die aktuellen Daten gespeichert oder wurde zuvor keine Änderung der OPC UA Daten erkannt, so wird geprüft, ob kein Job aktiv ist und in der Jobliste noch unbearbeitete Jobs gespeichert sind. Ist dies der Fall, so wird ein neuer Job gestartet und aus der Jobliste entfernt. Zum Schluss wird *stop.txt* ausgelesen. Steht hier eine 0, wird die *while*-Schleife erneut durchlaufen. Wurde von der Webseite eine 1 in die Datei geschrieben, werden die in der Jobliste verbliebenen Jobs nacheinander abgearbeitet und die verbleibenden Zugangszeiten in die JSON-Dateien geschrieben. Zum Schluss beendet das Programm die Verbindungen zu den OPC UA Servern und der Datenbank und schreibt eine 2 in *stop.txt*, damit die Webseite erkennt, dass das Programm beendet wurde. Auf die Kommunikation zwischen dem Programm und der Webseite wird im Unterkapitel 5.4 genauer eingegangen.

Der bisher beschriebene Programmablauf ist der normale Ablauf ohne Fehler (im Ablaufdiagramm grün dargestellt). Es kann allerdings sein, dass die Verbindung zum OPC UA Server abbricht oder ein anderes Problem im Programm auftritt. Für diesen Fall soll das Programm nicht einfach abstürzen, sondern zuvor die verbleibenden Jobs abarbeiten, die Zugangszeiten in den JSON-Dateien abspeichern und die Verbindungen zu den OPC UA Servern und der Datenbank trennen. Außerdem soll in *stop.txt* bei einem Absturz eine -1 und nach Beendigung des Programms eine -2 geschrieben werden, um der Webseite mitzuteilen, dass es zu einem Fehler gekommen ist. Dieses Szenario ist in der Abbildung 5.3 in rot dargestellt. Um den Absturz zu verhindern, wird um den normalen Programmcode ein `try` gesetzt. Kommt es in diesem Abschnitt zu einem Fehler, springt das Programm zu dem eingefügten `except` und führt den darunter stehenden Programmcode (in rot dargestellt) aus.

### 5.3.3. Speicherstruktur der Datenbank

Zum Speichern der Daten wird, wie beschrieben, eine MySQL Datenbank verwendet. Die Datenbank trägt den Namen *Rückmeldedaten* und beinhaltet neun Datenbanktabellen, jeweils eine für jeden Schnitt. In jeder Tabelle wird die Bestandsänderung (1 oder 0), ein Zeitstempel, der Zugang, der Abgang, der Bestand und die Durchlaufzeit des Schnittes abgespeichert. Zusätzlich zu diesen Daten werden weitere erforderliche Daten speziell für jeden Schnitt abgespeichert. Die Tabelle *Funktionsprüfung* hat z.B. zusätzlich die drei Datenfelder `el_OK`, `el_NOK` und `geprüft`. Die Tabelle A.8 im Anhang A.7.3 zeigt die in der Datenbank gespeicherten Tabellen mit deren Datenfeldern.

## 5.4. Graphische Oberfläche zur Visualisierung der Rückmeldedaten

Dieses Unterkapitel beschreibt die Realisierung der in der Konzeption (Unterkapitel 4.1) beschriebenen Oberfläche mittels einer Webseite. Wie in der Konzeption (Abschnitt 4.2.5) beschrieben, werden vier Hauptwebseiten (mobile und Leitstandseite je einmal aktuell und einmal historisch) benötigt. Außerdem sind die Startseiten für diese zu entwickeln. Nachfolgend werden zunächst die verwendeten Bibliotheken beschrieben. Anschließend wird auf die Struktur der Webseiten eingegangen, bevor das Auslesen der Datenbank erläutert wird. Abschließend werden die Darstellungen der Daten in Durchlaufdiagrammen und Tabellen beschrieben und Hinweise zur Auswertung der historischen Daten gegeben.

### 5.4.1. Verwendete Bibliotheken

Bevor die Webseiten umgesetzt werden, stellt sich die Frage, wie die Durchlaufdiagramme und tabellarischen Werte aus der Datenbank gelesen und angezeigt werden sollen. Hierfür wird für die Durchlaufdiagramme eine geeignete Bibliothek benötigt. Es wird sich für die JS-Bibliotheken Highcharts-6.2.0 und Highstock-6.2.0 der Firma HIGHCHARTS entschieden. Eine ausführliche Dokumentation mit Anwendungsbeispielen ist auf [4] einzusehen. Diese Bibliotheken benötigen zusätzlich die JS-Bibliotheken jquery-3.3.1 und jquery-3.3.1.min.js [6] zum Umgang mit den Datensätzen sowie die Bibliotheken moment.js, moment.min.js und moment-timezone.js [8] zur Berechnung der Zeit. Der Unterschied zwischen den Bibliotheken Highcharts und Highstocks ist die Möglichkeit in Highstock, zusätzlich zur Anzeige der Daten den Betrachtungszeitraum der angezeigten Daten mittels Buttons und Navigationsleiste zu verändern. So können zunächst alle vorhandenen Daten in das Diagramm geladen werden. Anschließend hat der Nutzer die Möglichkeit, den Betrachtungszeitraum zu wählen. Dieser

Vorteil wird bei der Darstellung der historischen Daten genutzt. Bei den aktuellen Daten ist der Betrachtungszeitraum fest vorgegeben. Aus diesem Grund wird für diese Durchlaufdiagramme die Bibliothek Highcharts verwendet, die die Grundlage zu Highstock bildet.

## 5.4.2. Struktur und Design der Webseite

Die Webseiten werden nach den in der Konzeption (Abschnitt 4.1.4) beschriebenen Vorgaben entwickelt. Zunächst wird die Startseite beschrieben und anschließend auf die Entwicklung der aktuellen Leitstandseite eingegangen. Danach wird abgeleitet aus der Leitstandseite die mobile Seite für die aktuellen Daten entwickelt und abschließend die Entwicklung der historischen Webseiten beschrieben.

### Startseite

Die Startseiten zu der mobilen und Leitstandseite sind über die URL [192.168.33.151/Rueckmeldedaten/home\\_m.html](http://192.168.33.151/Rueckmeldedaten/home_m.html) (Mobil) und [192.168.33.151/Rueckmeldedaten/home\\_1.html](http://192.168.33.151/Rueckmeldedaten/home_1.html) (Leitstand) erreichbar. Die Hauptaufgaben der Startseite sind die Navigation zu den Anzeigen der jeweiligen Seite (aktuell und historisch) und die Steuerung des Raspberrys. Die Navigation zu den aktuellen oder historischen Daten erfolgt über Buttons, die die entsprechende Seite durch einen Hyperlink aufrufen. Zusätzlich zu dem Namen der HTML-Datei der Seite wird ein `?detailtiefe=6` hinter das `.html` geschrieben. Das bewirkt, dass die Seite mit der Detailtiefe *sechs Schnitte* geladen wird. Sollte irgendwann der Wunsch entstehen, nach dem Laden die Detailtiefe *vier Schnitte* darzustellen, kann dies mit einer Änderung der 6 in eine 4 erreicht werden. Zusätzlich zu den Verlinkungen der Hauptseiten werden das Video zur Modellfabrik und die PDF des EForum Artikels [34] auf der Startseite verlinkt. Das Steuern des Raspberrys beschränkt sich auf das Starten und Stoppen des Pythonprogramms sowie das Herunterfahren des Raspberrys. Diese Aktionen werden durch Buttons ausgelöst. Das Starten des Pythonprogramms und das Herunterfahren des Raspberrys erfolgt über den Aufruf von PHP-Sripten, die mittels `shell_exec` Konsoleneingaben ausführen können. Beim Start des Pythonprogramms wird zusätzlich zu der Konsoleneingabe eine 3 in `stop.txt` geschrieben. Das Stoppen des Pythonprogramms erfolgt durch ein PHP-Script, das eine 1 in `stop.txt` schreibt.

**Kommunikation Startseite und Pythonprogramm** Das Starten und Stoppen des Programmes soll nur abwechselnd möglich sein. Es soll also ein Programmstart nur möglich sein, wenn das Programm nicht läuft und der Programmstopp nur ermöglicht werden, wenn es zuvor gestartet wurde. Das Herunterfahren darf nur bei gestopptem Programm erfolgen. Damit die Webseite den Status des Programms erhält, wird `stop.txt` ausgelesen. In dieser

wird der Status des Pythonprogramms gespeichert. Die Datei kann verschiedene Status enthalten (siehe Tabelle 5.1), die entweder von der Webseite oder dem Pythonprogramm geschrieben werden.

Tabelle 5.1.: Status *stop.txt*

Status	Bedeutung	geschrieben durch
0	Python läuft	Python
1	Python soll gestoppt werden	Webseite
2	Python ist gestoppt	Python
3	Python wird gestartet	Webseite
-1	Python wird nach Fehler gestoppt	Python
-2	Python ist nach Fehler gestoppt	Python

Durch das zyklische Lesen der Datei *stop.txt* kann die Startseite den Status des Pythonprogramms auslesen.

**Anzeige des Status des Pythonprogrammes auf der Startseite** Die JS-Funktion `lese_python`, die in `home.js` gespeichert ist, wird auf den Startseiten zyklisch ausgeführt. Diese fragt den Status des Programms ab und aktiviert bzw. deaktiviert die Buttons zur Steuerung des Raspberrys entsprechend. Die Tabelle 5.2 zeigt das Verhalten der Buttons *Python starten*, *Python stoppen* und *Herunterfahren* bei den verschiedenen Status.

Tabelle 5.2.: Verhalten der Button auf Status des Pythonprogramms

Status	Python starten	Python stoppen	Herunterfahren
0	unsichtbar	sichtbar	grau und Steuerung deaktiviert
1	unsichtbar	sichtbar Text: Wird gestoppt	grau und Steuerung deaktiviert
2	sichtbar	unsichtbar	Steuerung aktiviert
3	sichtbar Text: Wird gestartet	unsichtbar	grau und Steuerung deaktiviert
-1	unsichtbar	sichtbar Text: Fehler! Wird gestoppt	grau und Steuerung deaktiviert
-2	sichtbar Text: Python nach Fehler starten	unsichtbar	Steuerung aktiviert

Zum Ablauf des Startens und Stoppens des Pythonprogramms zeigt die Abbildung 5.4 (a) das Sequenzdiagramm bei Starten und Stoppen des Programms durch die Webseite und (b) das Stoppen durch einen Programmfehler.

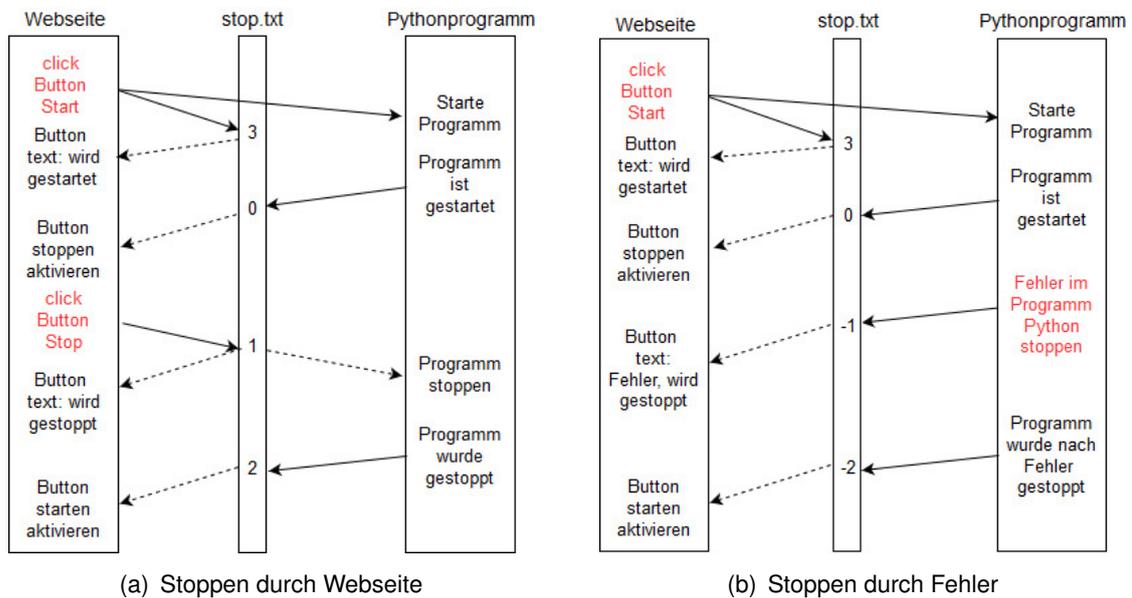


Abbildung 5.4.: Sequenzdiagramm zum Starten und Stoppen des Pythonprogramms

### Leitstand aktuell

Die aktuelle Leitstandseite wird wie in der Konzeption beschrieben in vier Bereiche unterteilt, welche sich wiederum aus mehreren Bereichen zusammensetzen. Die Aufteilung der Webseite zeigt die Abbildung 5.5.

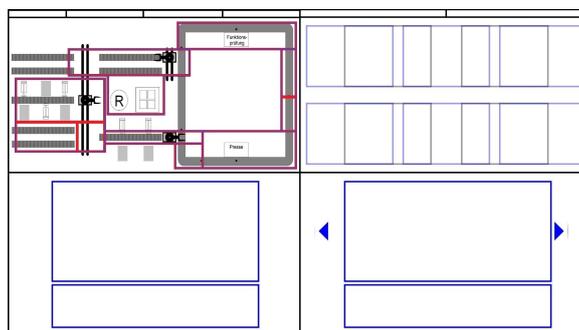


Abbildung 5.5.: Aufteilung der Leitstand Webseite aktuell in Bereiche

Im Folgenden werden die Quadranten einzeln betrachtet und die Aufteilung dieser in Bereiche erläutert. Begonnen wird mit dem Bereich `Navigation`, der im oberen linken Quadranten zu finden ist. Anschließend wird der sich im oberen rechten Quadranten befindende Bereich `Übersicht` beschrieben, bevor der Bereich `Gesamtanlage` im unteren linken Quadranten beschrieben wird. Zum Schluss wird noch die Unterteilung des Bereiches `Detailschnitt` im unteren rechten Quadranten dargestellt.

Die Unterteilung des Bereiches **Navigation** ist schematisch in Abbildung 5.6 dargestellt.

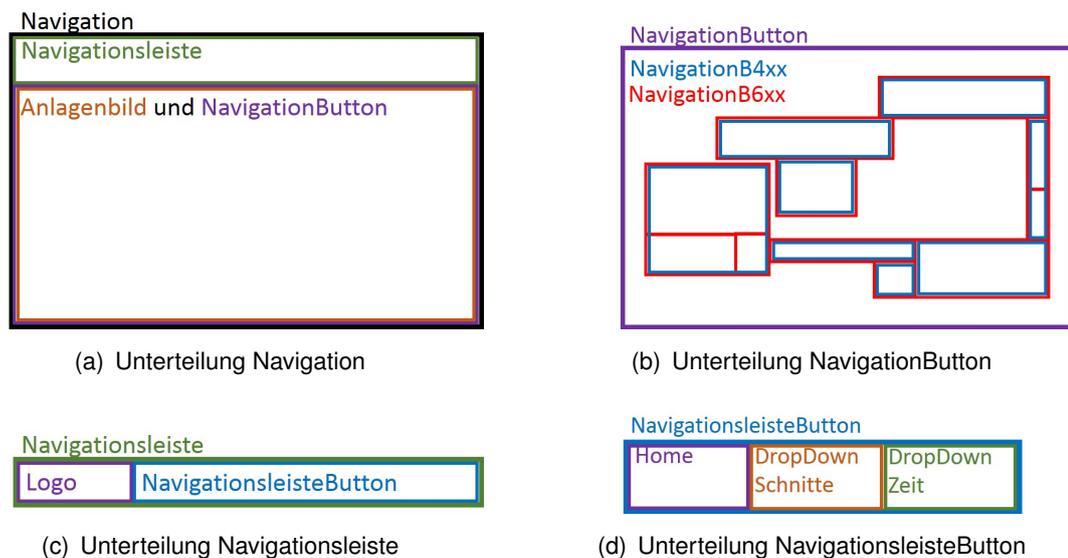


Abbildung 5.6.: Unterteilung des Bereiches Navigation

Dieser unterteilt sich in drei Bereiche (siehe (a)). Zum einen in den Bereich `Navigationsleiste` (grün) und zum anderen in die Bereiche `Anlagenbild` (orange) und `NavigationButton` (lila). Die Bereiche `Anlagenbild` und `NavigationButton` befinden sich räumlich am selben Platz, allerdings liegt der Bereich `Anlagenbild`, der die schematische Darstellung als Bild-Datei beinhaltet, eine Ebene unter dem Bereich `NavigationButton`. Der Bereich `NavigationButton` beinhaltet die Buttons zum Klicken der verschiedenen Schnitte (Siehe (b)). Es befinden sich zum einen die Buttons für vier Schnitte (blau) und zum anderen für sechs Schnitte (rot) in diesem Bereich. Die Bezeichnung der Buttons lautet `NavigationB4xx` oder `NavigationB6xx` je nach dem, ob der Button zu vier oder sechs Schnitten gehört. Es kann vorkommen, dass für einen Schnitt mehreren Buttons benötigt werden. Aus diesem Grund gibt das erste x den Schnitt an, zu dem er gehört und das zweite x wird von 1 hochgezählt, wenn mehr als ein Button verwendet wird. Alle Buttons werden transparent dargestellt. So ist das Anlagenbild sichtbar, aber die Buttons können geklickt werden. Je

nach Auswahl der Detailtiefe werden im späteren Verlauf nur die benötigten Buttons durch ein JS aktiviert und sichtbar gemacht. Die Navigationsleiste (siehe (c)) teilt sich in den Bereich Logo (lila) und NavigationsleisteButton (blau). Im Bereich Logo wird das HAW Logo als Bild-Datei eingefügt und der Bereich NavigationsleisteButton beinhaltet die Aktionsfelder zur Navigation (siehe (d)). Der Button Home (lila) führt zurück auf die Startseite. Die Bereiche DropDownSchnitte (orange) und DropDownZeit (grün) beinhalten jeweils ein DropDown-Menü, wie in der Konzeption (Abschnitt 4.1.4) vorgesehen wurde.

Die Aufteilung des Bereiches **Übersicht** zeigt die Abbildung 5.7.

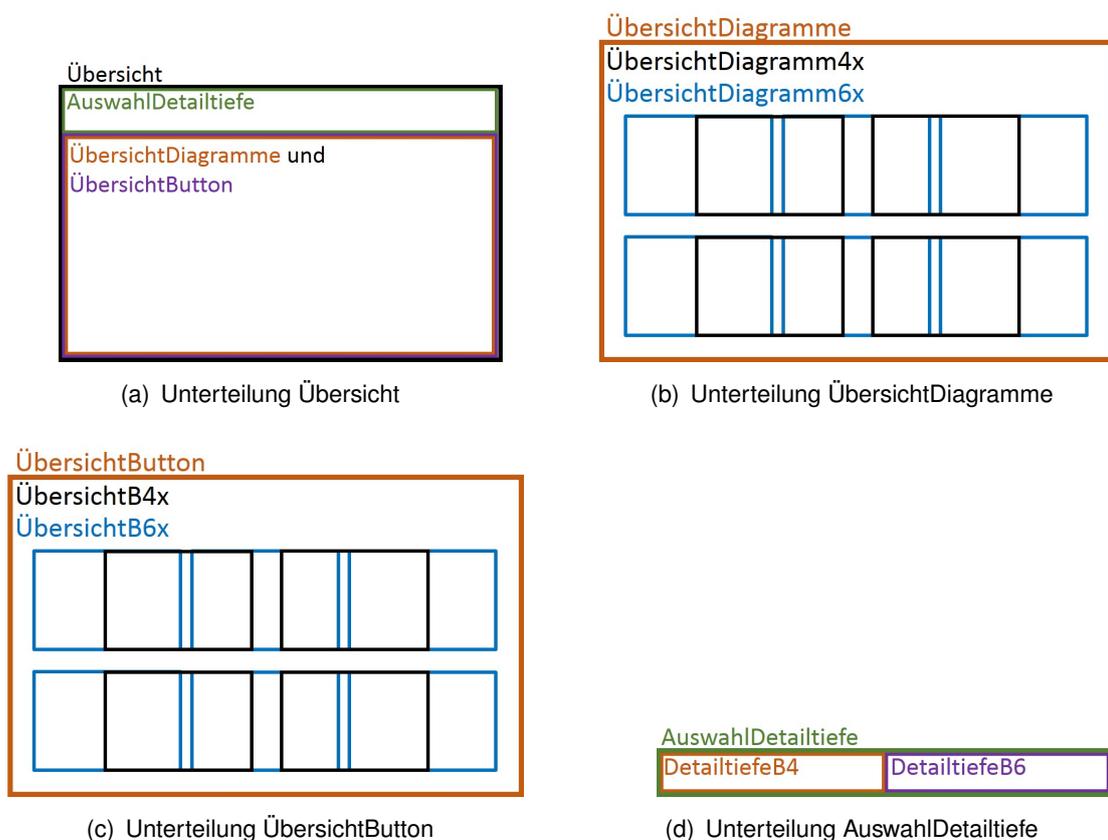


Abbildung 5.7.: Unterteilung des Bereiches Übersicht

Wie in (a) zu sehen, unterteilt sich der Bereich in AuswahlDetailtiefe (grün), sowie ÜbersichtDiagramme (orange) und ÜbersichtButton (lila). Die Bereiche ÜbersichtDiagramme und ÜbersichtButton liegen übereinander. Um die Buttons klicken zu können, liegt dieser Bereich eine Ebene höher als der Bereich ÜbersichtDiagramme. Der Bereich ÜbersichtDiagramme (siehe (b)) beinhaltet die Bereiche ÜbersichtDiagramm4x (schwarz) und ÜbersichtDiagramm6x (blau)

zur Darstellung der Durchlaufdiagramme mit vier oder sechs Schnitten. Das x steht hier für die Nummer des Schnittes. Wie bei dem Bereich `NavigationButton` werden im Verlauf der Realisierung die Diagramme, die gerade nicht relevant sind, durch JS ausgeblendet. Der Bereich `ÜbersichtButton` ist wie der Bereich `ÜbersichtDiagramme` aufgebaut (siehe (c)). Die Bereiche für die Buttons sind mit `ÜbersichtB4x` und `ÜbersichtB6x` benannt, wobei das x wieder für die Schnittnummer steht. Der Bereich `AuswahlDetailtiefe` (siehe (d)) beinhaltet die zwei Bereiche `DetailtiefeB4` (orange) und `DetailtiefeB6` (lila), um, wie in der Konzeption (Abschnitt 4.1.4) beschrieben, zwischen den Detailtiefen wechseln zu können.

Der Bereich **Gesamtanlage** beinhaltet, wie in Abbildung 5.8 zu sehen, zwei Bereiche. In dem Bereich `GesamtanlageDiagramm` (grün) wird das Durchlaufdiagramm der Gesamtanlage und im Bereich `GesamtanlageTabelle` (orange) die in der Konzeption (Abschnitt 4.1.2) beschriebenen, dazugehörigen Daten in tabellarischer Form dargestellt.

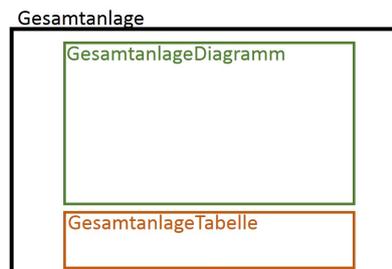


Abbildung 5.8.: Unterteilung des Bereiches Gesamtanlage

Wie der Bereich `Gesamtanlage` beinhaltet der Bereich **Detailschnitt** auch die Bereiche zur Darstellung des Durchlaufdiagramms und die Tabelle. Wie die Abbildung 5.9 zeigt, sind diese mit `DetailschnittDiagramm` (grün) und `DetailschnittTabelle` (orange) benannt. In den Bereichen `DetailschnittDiagramm` und `DetailschnittTabelle` werden die ausgelesenen Daten des geforderten Schnittes dargestellt.

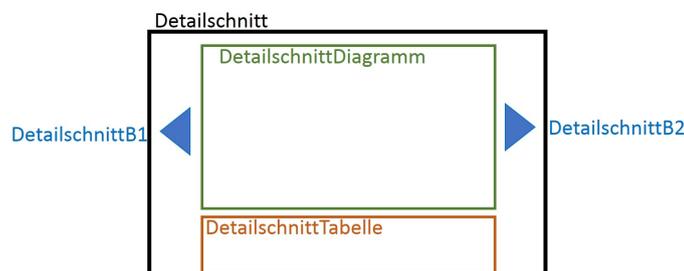


Abbildung 5.9.: Unterteilung des Bereiches Detailschnitt

Zusätzlich werden die zwei Bereiche zum Wechseln des Schnittes mit den Namen `DetailschnittB1` und `DetailschnittB2` (blau) eingefügt. Die sich darin befindenden Buttons ermöglichen es, den vorherigen oder nächsten Schnitt direkt anzuzeigen.

### Mobil aktuell

Der Aufbau der mobilen Seite zur Darstellung der aktuellen Werte wird wie in der Konzeption (Abschnitt 4.1.4) beschrieben vorgenommen. Zur Realisierung können die in Abschnitt 5.4.2 beschriebenen Bereiche verwendet werden, wobei eine neue Positionierung der Bereiche vorgenommen wird. Der Bereich `Navigationsleiste` mit allen sich darin befindenden Bereichen erstreckt sich nun über die gesamte Breite der Webseite. Der Bereich `AuswahlDetailtiefe` wird unten links in der Ecke positioniert und darüber der Bereich `Navigation` um 90° gedreht. Die Bereiche `Übersicht`, `Gesamtanlage` und `Detailschnitt` werden alle, wie auch in der Konzeption beschrieben, in dem nun noch freien Bereich positioniert. Je nach Auswahl der Ansicht wird der geforderte Bereich durch eine JS-Funktion in den Vordergrund gestellt. Die Aufteilung der mobilen Seite, mit der Ebene `Gesamtanlage` im Vordergrund, zeigt die Abbildung 5.10.

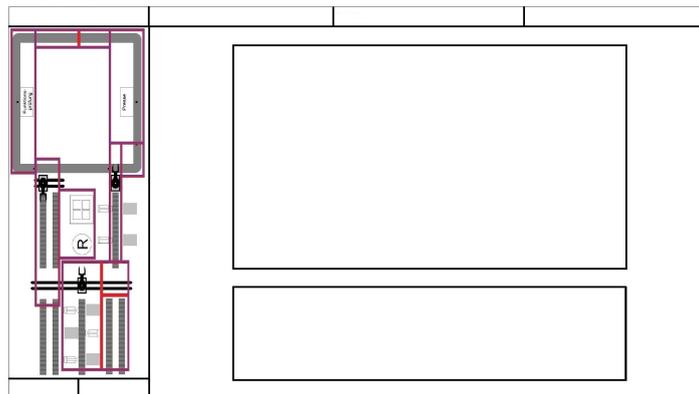


Abbildung 5.10.: Unterteilung der mobilen Webseite für aktuelle Daten am Beispiel Gesamtanlage

### Webseiten historisch

In der Konzeption (Abschnitt 4.1.4) werden die Änderungen des Layouts für die historische Webseite beschrieben. Gefordert werden Buttons zur Änderung des Betrachtungszeitraumes der Daten und ein Button zum erneuten Laden der Daten aus der Datenbank. Die Buttons zum Wechsel des Betrachtungszeitraumes werden, wie in Abschnitt 5.4.1 beschrieben,

durch die JS Bibliothek *Highstock* realisiert und zählen zu dem Bereich der Durchlaufdiagramme `GesamtanlageDiagramm` und `DetailschnittDiagramm`. Es ist also nur ein weiterer Button in die Navigationsleiste hinzuzufügen. Wie die Abbildung 5.11 zeigt, wird der Bereich `Refresh` (orange) wie in der Konzeption gefordert platziert.



Abbildung 5.11.: Unterteilung der Navigationsliste für die historische Webseite

Die Seiten für die aktuellen Daten lesen die Daten mittels JS-Funktion im Sekundentakt aus der Datenbank aus. Die Webseiten für die historischen Daten sollen die Daten beim Laden der Seite oder nach Nutzerinteraktion aus der Datenbank lesen. Um dies zu ermöglichen, müssen für die Detailtiefe *vier Schnitte* vier und für die Detailtiefe *sechs Schnitte* sechs Bereiche für Diagramme und Tabellen vorhanden sein. Die jeweils 10 Bereiche `DetailschnittDiagrammX` und `DetailschnittTabelleX` werden alle auf der gleichen Position dargestellt. Je nach Forderung der Detailtiefe und des anzuzeigenden Schnittes werden die Bereiche durch eine JS-Funktion im Vordergrund platziert. Genau wie die Durchlaufdiagramme für die Detailansicht müssen auch die Daten für die Diagramme im Bereich Übersicht ausgelesen und dargestellt werden. Dafür werden statt des Bereiches `ÜbersichtDiagramme` zwei Bereiche `ÜbersichtDiagramm4` und `ÜbersichtDiagramm6` verwendet, die jeweils die Bereiche für die vier oder sechs Durchlaufdiagramme beinhalten. Je nach Auswahl der Detailtiefe wird eine JS-Funktion den geforderten Bereich in den Vordergrund stellen.

### 5.4.3. Steuerung der Webseiten

#### Aktuelle Webseiten

Das Verhalten der aktuellen Webseite wird durch die JS-Funktionen `auswahl_1` (Leitstand) und `auswahl_m` (Mobil) bestimmt. Diese werden bei dem Klick auf einen Button auf der Webseite aufgerufen. Die HTML-Datei übergibt die Variable `eingabe_detailtiefe`, die die geforderte Detailtiefe angibt, sowie die Variable `eingabe_anzeige`, die den geforderten Inhalt übergibt. Alle Buttons, bis auf `DetailschnittB1` und `DetailschnittB2`, übergeben als `eingabe_anzeige` die Nummer des darzustellenden Schnittes. Beim Klicken der Buttons `DetailschnittB1` und `DetailschnittB2` ist keine feste Angabe des geforderten Schnittes möglich, da immer der vorherige Schnitt

$$eingabe\_anzeige = anzeige - 1 \quad (5.2)$$

oder der nächste Schnitt

$$\text{eingabe\_anzeige} = \text{anzeige} + 1 \quad (5.3)$$

dargestellt werden soll. Die JS-Funktionen sind in der Datei *index.js* abgelegt und werden im Folgenden beschrieben.

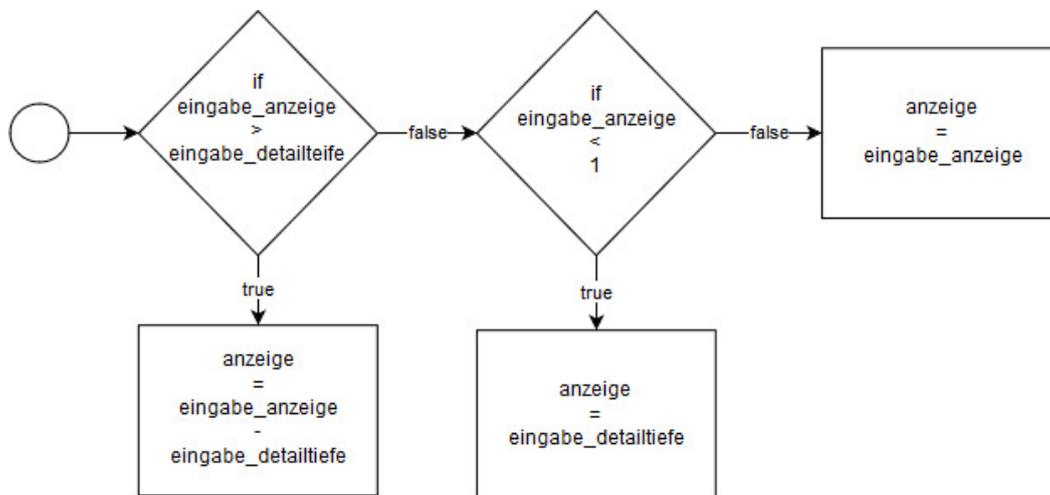
**auswahl\_1** Das Verhalten der aktuellen Leitstandwebseite wird durch die Funktion *auswahl\_1* in der JS-Datei *index.js* gesteuert. Zu Beginn der JS-Funktion werden die Variablen *eingabe\_detailtiefe* und *eingabe\_anzeige* in den globalen Variablen *detailtiefe* und *anzeige* gespeichert. Dabei ist zu beachten, dass die Variable *anzeige* nur Zahlenwerte von  $1 \leq \text{anzeige} \leq \text{eingabe\_detailtiefe}$  annehmen darf. Die Zuordnung der Variable *anzeige* zu den Schnitten ist in Tabelle 5.3 aufgelistet.

Tabelle 5.3.: Zuordnung der Schnitte zu der Variable *anzeige* bei vier und sechs Schnitten

Wert	Zuordnung vier Schnitte	Zuordnung sechs Schnitte
1	Bestückung	Bestückung
2	Transfersystem	Funktionsprüfung
3	Endkontrolle	Presse
4	Fertigprodukt_Lager_u_Handarbeit	Endkontrolle
5		Handarbeit
6		Fertigprodukt_Lager

Die Variable *eingabe\_anzeige* kann durch die Gleichungen 5.2 und 5.3 Zahlenwerte von  $0 \leq \text{eingabe\_anzeige} \leq \text{eingabe\_detailtiefe} + 1$  annehmen. Aus diesem Grund muss die Variable vor der Speicherung geprüft und gegebenenfalls umgerechnet werden.

Der Ausschnitt aus der Funktion *anzeige\_1* zur Umrechnung des darzustellenden Schnittes ist in Abbildung 5.12 als Ablaufdiagramm gezeigt.

Abbildung 5.12.: Ablaufdiagramm zur Berechnung der Variable *anzeige*

Zunächst wird geprüft, ob *eingabe\_anzeige* größer als die geforderte Detailtiefe ist. Ist dies der Fall, wurde der Button *DetailschnittB2* geklickt und vorher wurde der letzte Schnitt der aktuellen Detailtiefe dargestellt. Der neue darzustellende Schnitt lässt sich durch

$$anzeige = eingabe\_anzeige - eingabe\_detailtiefe \quad (5.4)$$

berechnen.

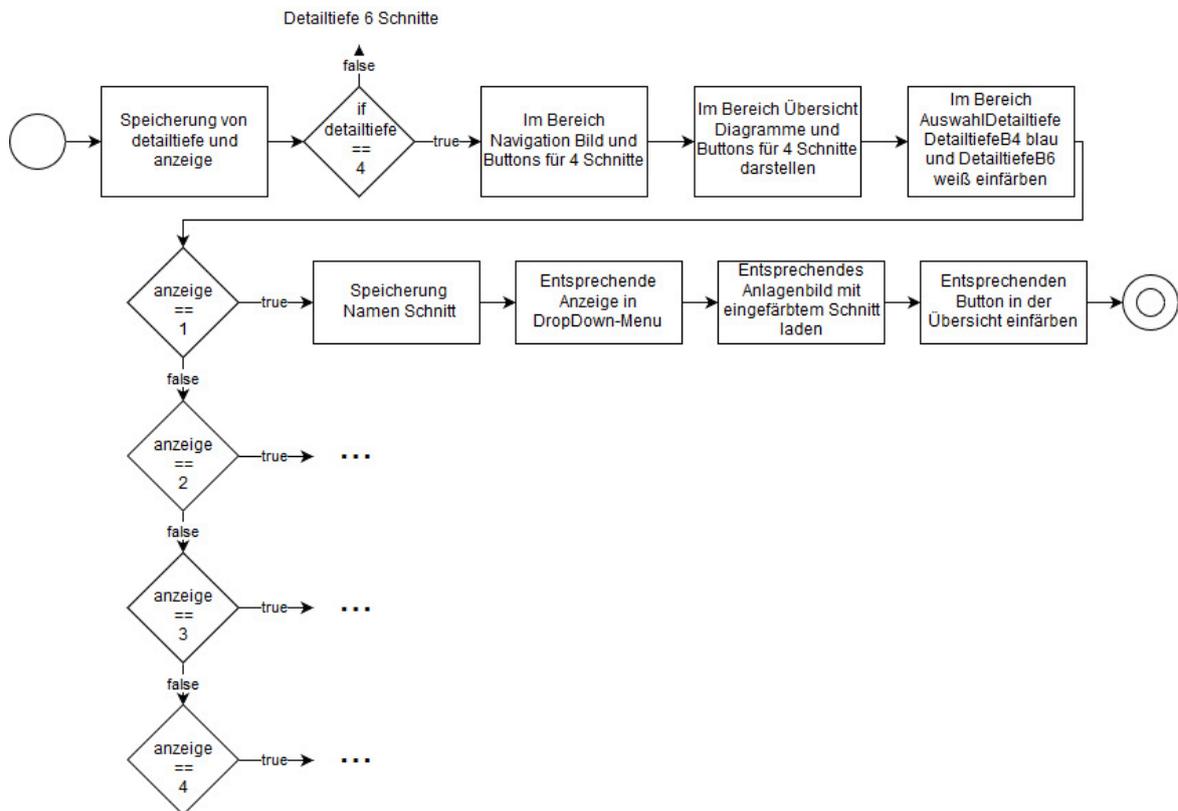
Ist *eingabe\_anzeige* kleiner als 1, wurde der Button *DetailschnittB1* geklickt und vorher war der Schnitt *Bestückung* aktiv. In diesem Fall ergibt sich der darzustellende Schnitt zu

$$anzeige = eingabe\_detailtiefe. \quad (5.5)$$

Ist keine der vorherigen Bedingungen eingetreten, ergibt sich der anzuzeigende Schnitt aus

$$anzeige = eingabe\_anzeige. \quad (5.6)$$

Das Ablaufdiagramm der gesamten JS-Funktion für den Fall, dass die Detailtiefe 4 gewählt wurde, ist in [Abbildung 5.13](#) dargestellt.

Abbildung 5.13.: Ablaufdiagramm der JS-Funktion `auswahl_I`

Nach der Speicherung der Detailtiefe und des darzustellenden Schnittes wird zwischen den Detailtiefen vier und sechs Schnitten unterschieden. Bei vier Schnitten wird in dem Bereich `Navigation` das Anlagenbild für vier Schnitte geladen und die Navigationsbuttons `NavigationB4xx` sichtbar, sowie die `NavigationB6xx` unsichtbar gemacht. In dem Bereich `Übersicht` werden die Diagramme und Buttons für vier Schnitte sichtbar, und die für sechs Schnitte unsichtbar gemacht. Anschließend wird der Button `DetailtiefeB4` zur Angabe der Detailtiefe *vier Schnitte* blau und der `DetailtiefeB6` zur Angabe der Detailtiefe *sechs Schnitte* weiß eingefärbt. Abschließend wird eine *Switch Case*-Anweisung ausgeführt. Als Variable wird `anzeige` verwendet, die in diesem Fall die Werte 1 bis 4 annehmen kann. In jedem Fall (1 bis 4) wird zunächst der Name für den geforderten Schnitt gespeichert, die Anzeige des Bereiches `DropDownSchnitteSelect` geändert, das Anlagenbild mit eingefärbtem Schnitt geladen und die Buttons im Bereich `Übersicht` entsprechend eingefärbt.

Der Ablauf bei der Detailtiefe *sechs Schnitte* ist entsprechend. Hier werden allerdings das Anlagenbild, sowie die Buttons und Diagramme für sechs Schnitte geladen, sichtbar gemacht und eingefärbt. Die *Switch Case*-Anweisung hat nun sechs Fälle für jeden einzelnen Schnitt.

**auswahl\_m** Die Funktion `auswahl_m` steuert das Verhalten der mobilen Webseite. Sie ist ähnlich aufgebaut wie die `auswahl_1`. Allerdings werden, wie in 5.4.2 beschrieben, die Bereiche *Übersicht*, *Gesamtanlage* und *Detailschnitt* übereinander positioniert und der geforderte Bereich im Vordergrund dargestellt. Das Ablaufdiagramm der Funktion für vier Schnitte zeigt die Abbildung 5.14.

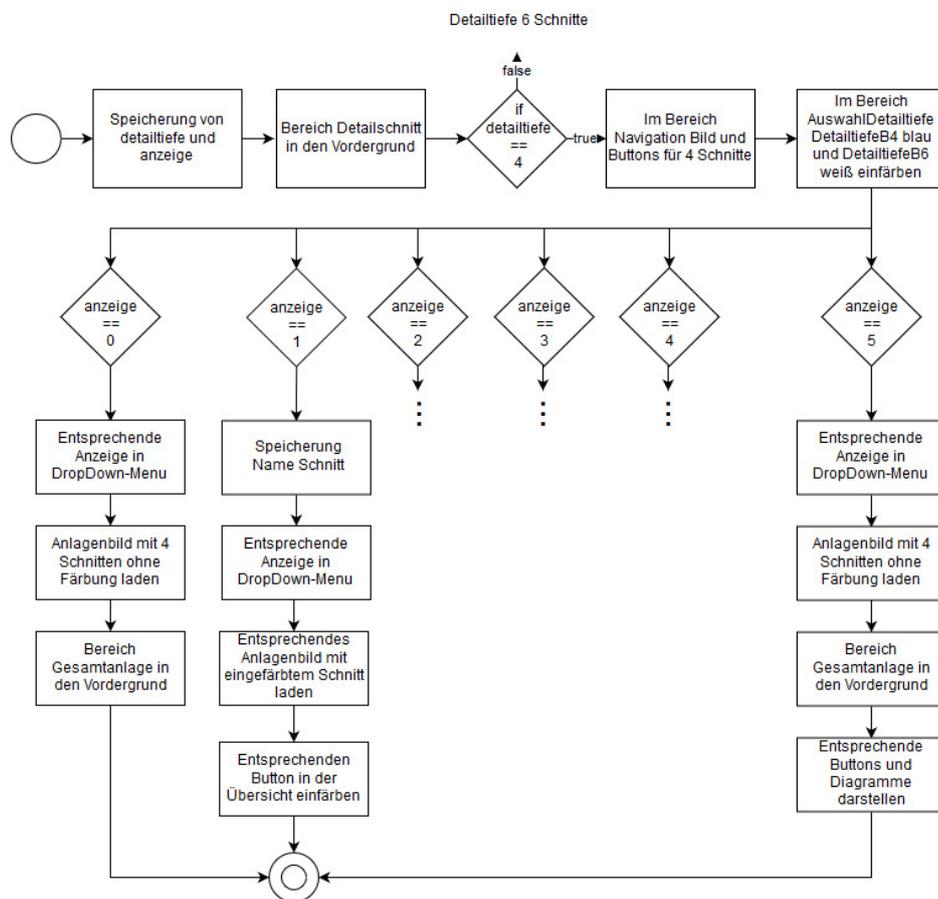


Abbildung 5.14.: Ablaufdiagramm der JS-Funktion `auswahl_m`

In den meisten Fällen (1-4 und 1-6), wird der Bereich *Detailschnitt* dargestellt. Aus diesem Grund wird vor der Prüfung der Detailtiefe dieser in den Vordergrund gestellt. Anschließend wird die Detailtiefe geprüft und die entsprechenden Navigationsbuttons sichtbar

gemacht, sowie der entsprechende Button zur Anzeige und Auswahl der Detailtiefe eingefärbt. Zum Schluss wird auch in dieser Funktion eine *Switch Case*-Anweisung ausgeführt. Zusätzlich zu den Werten 1 bis 4 bzw. 1 bis 6, die bereits beschrieben wurden, kann die Variable `anzeige` und damit die Fälle der *Switch Case*-Anweisung die Werte 0 und 7 annehmen. Dadurch wird angezeigt, dass der Bereich Gesamtanlage (Fall 0), Übersicht mit vier Schnitten (Fall 5) oder Übersicht mit sechs Schnitten (Fall 7) dargestellt werden soll. Um die Berechnung des Schnittes, die in Abbildung 5.12 dargestellt ist, nicht zu verfälschen, übergibt die HTML-Datei in diesem Fall zusätzlich die Variable `übersicht`, die `true` gesetzt wird, wenn ein anderer Bereich als `Detailschnitt` angezeigt werden soll. In den Fällen 1 bis 4 bzw. 1 bis 6 wird der Name des Schnittes gespeichert, die Anzeige des Drop-down-Menu für die Schnitte aktualisiert und das korrekte Anlagenbild dargestellt. Die Fälle 0, 5 (wenn `übersicht == true`) und 7 (wenn `übersicht == true`) stellen zusätzlich den geforderten Bereich in den Vordergrund. Die Fälle 5 und 7 machen außerdem die geforderten Diagramme und Buttons sichtbar.

**Anzeige des Status des Pythonprogrammes auf den aktuellen Seiten** Die Anzeige der aktuellen Daten ist nur sinnvoll, wenn das Pythonprogramm gerade läuft. Ist es durch einen Fehler beendet oder noch nicht gestartet worden, so soll ein Warnungsfenster erscheinen. Dieses soll den Nutzer daran erinnern, das Programm zu starten. Zur Realisierung wird ein weiterer Bereich `Warnung` in die aktuellen HTML-Dateien eingefügt und in diesem zwei Buttons zum Starten des Programms und zum Zurückkehren zur Startseite sowie ein Infotext eingefügt. Die JS-Funktion `lese_python` in der Datei `index.js` fragt den Status des Programmes zyklisch ab und stellt den Bereich `Warnung` in den Vordergrund, wenn das Programm nicht läuft (Status in `stop.txt != 0`). Der Bereich `Warnung` ist in Abbildung 5.15 gezeigt.

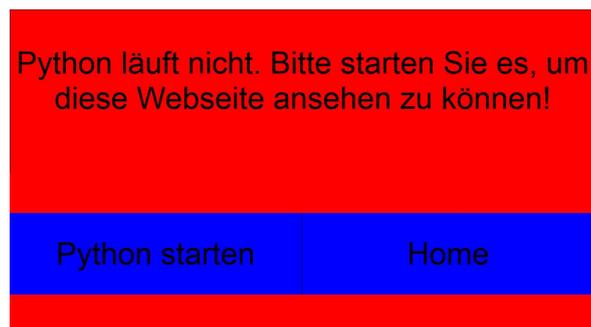


Abbildung 5.15.: Bereich Warnung bei nicht eingeschaltetem Pythonprogramm

## Historische Webseiten

Die Steuerung der historischen Webseiten übernehmen die JS-Funktionen `auswahl_l` (Leitstand) und `auswahl_m` (Mobil) in der JS-Datei `index_historie.js`. Der Ablauf dieser orientiert sich an den gleichnamigen Funktionen der aktuellen Webseiten. Allerdings werden in den Fällen 1 bis 4 bzw. 1 bis 6 zusätzlich die Bereiche `DetailschnittDiagramm`x und `DetailschnittTabelle`x des geforderten Schnittes in den Vordergrund gestellt. Zum Schluss wird geprüft, ob der *Daten Neu Laden* Button geklickt wurde und gegebenenfalls die Daten neu ausgelesen.

### 5.4.4. Auslesen der Datenbank

Das Abrufen der Daten aus der Datenbank wird für die aktuellen Webseiten durch die JS-Funktion `zurückgegebene_daten`, für die historischen Webseiten durch die JS-Funktion `lese_daten` angestoßen. Hier wird durch die Funktion `.getJSON` der Bibliothek jQuery das entsprechende PHP-Script aufgerufen, das die geforderten Daten aus der Datenbank ausliest und anschließend in Durchlaufdiagrammen und in tabellarischer Form ausgibt. Die Funktion `.getJSON` ist eine AJAX-Funktion, die einen asynchronen Ablauf ermöglicht und den serverseitigen php-Interpreter aktiviert. Das PHP-Script für die aktuellen Daten liest nur Daten aus der Datenbank, die jünger als 30 Minuten sind. Da der Raspberry nicht mit dem Internet verbunden ist und dadurch die Systemuhr nicht konsistent zu der SPS-Zeit läuft, schreibt das Pythonprogramm die über OPC UA empfangene `aktuelleZeit - 30Minuten` in eine PHP-Datei. Diese Zeit nutzt das PHP-Script zum Auslesen der Datenbank, um die letzten Daten zu lesen. Für die historischen Daten wird ein PHP-Script aufgerufen, das die Datenbank vollständig ausliest. Anschließend kann der Nutzer über die Highstock Bibliothek verschiedene Zeiträume betrachten. Das Lesen der Daten findet für die aktuellen Seiten zyklisch statt. Bei den historischen Daten wird die Datenbank nur beim Laden der Seite und bei Klick auf den *Daten Neu Laden* Button gelesen.

### 5.4.5. Darstellung der Durchlaufdiagramme

Die Theorie der Durchlaufdiagramme wird in den Grundlagen (Abschnitt 2.1.3) behandelt. Abbildung 2.6 zeigt ein solches Durchlaufdiagramm. Im vorliegenden Anwendungsfall bestehen Aufträge immer aus einem Produkt. Außerdem ändert sich die Durchlaufzeit verschiedener Aufträge nicht, da immer dasselbe Produkt gefertigt wird. Aus diesem Grund wird auf der Abszisse der Durchlaufdiagramme in dieser Arbeit nicht wie in der Theorie die Vorgabestunden, sondern die Anzahl der Aufträge und damit Produkte dargestellt.

### 5.4.6. Darstellung der Tabellen

Die Tabellen für die einzelnen Schnitte werden mit dem HTML-Tag `<table>` dargestellt. Diese wird mit der JS-Funktion `tabelle_zeichnen` als HTML Code erstellt. Diese JS-Funktion erwartet ein Array von Daten, das von dem PHP-Script zurückgegeben wird und schreibt mit dem erhaltenen Array den HTML-Quellcode. Die Daten in den Tabellen beziehen sich immer auf den aktuell im Durchlaufdiagramm dargestellten Zeitraum.

### 5.4.7. Hinweise zur Auswertung der Daten auf den historischen Webseiten

Zur Auswertung der historischen Daten werden alle Daten aus der Datenbank geladen. Durch die Änderung des Betrachtungszeitraumes über die Zeitbuttons lassen sich verschiedene Betrachtungszeiträume für das Durchlaufdiagramm und die Tabelle einstellen. Um eine plausible Auswertung der Rückmeldedaten zu erhalten, darf ein Betrachtungszeitraum keinen kompletten Reset der Modellfabrik beinhalten. Das bedeutet, dass der Zu- und Abgang im betrachteten Durchlaufdiagramm nicht auf 0 zurückgesetzt worden sein darf. Andernfalls ist die Mittelwertbildung über den Betrachtungszeitraum nicht korrekt.

## 5.5. Zusammenfassung des realisierten Gesamtsystems

Realisiert ist ein System zur Gewinnung, Übertragung und Visualisierung der Rückmeldedaten der Modellfabrik. Bei der Realisierung wird sich an die in der Konzeption (Kapitel 4) beschriebenen Vorgaben gehalten. Das Anzeigesystem, bestehend aus dem LG Electronics 75UK6200 Fernseher, dem Windows 10 Rechner Fujitsu Q 957 und dem TrekStor PrimeTab, ist in den Räumlichkeiten der Modellfabrik aufgestellt. Der Webserver und die Datenbank sind auf dem Raspberry eingerichtet und der OPC UA Client ist mit einem Pythonprogramm realisiert. Die vier Webseiten zur Darstellung der aktuellen sowie historischen Daten Mobil und Leitstand sind entwickelt. Zur Visualisierung der Durchlaufdiagramme werden die Bibliotheken Highcharts und Highstock verwendet. Die Startseiten ermöglichen den Zugang zu den Hauptseiten sowie weiteren Informationen über Hyperlinks. Zusätzlich ist das Starten und Stoppen des Pythonprogramms sowie das sichere Herunterfahren des Raspberrys über die Webseite möglich. Das System kann sowohl mit dem vorhandenen Bussystem (Konzept 1) als auch mit dem eingerichteten WLAN (Konzept 2) betrieben werden.

Der geforderte Gesamtaufbau, der in Abbildung 4.17 dargestellt ist, kann wie gewünscht umgesetzt werden. Zum weiteren Betrieb des Systems bleibt das Konzept 2 (Übertragung mit WLAN) implementiert. Das Netzwerk ist als 2,4GHz-Netzwerk aufgebaut. Sollte es zu einer Überlastung des Frequenzbandes kommen, kann dieses entweder auf 5GHz umgestellt, oder zu der Nutzung des Bussystemes zurückgekehrt werden.

Die Abbildung 5.16 zeigt das realisierte Anzeigesystem in Aktion.

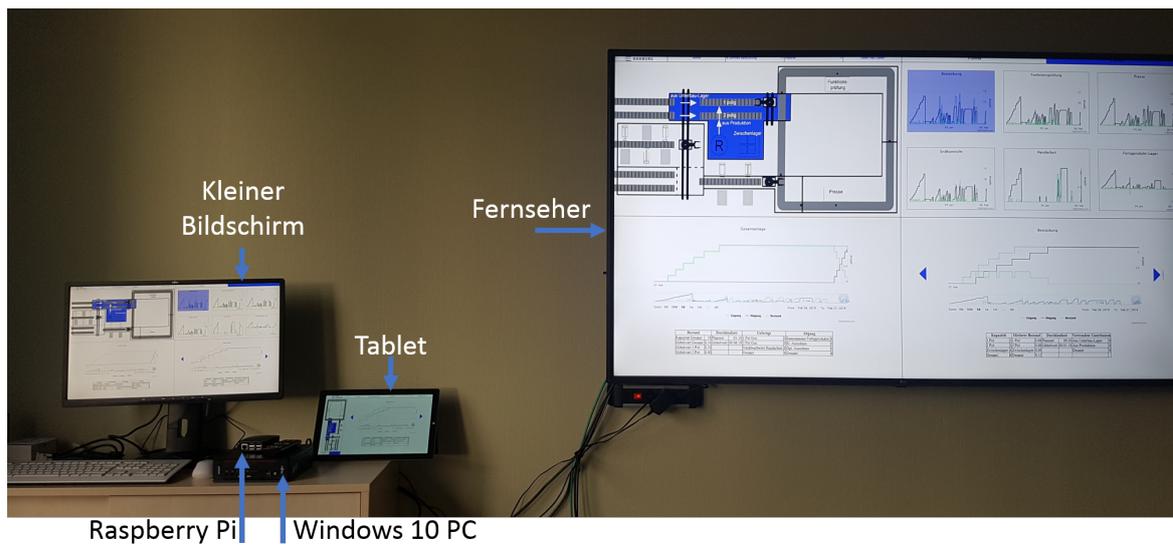


Abbildung 5.16.: Realisiertes Anzeigesystem

# 6. Verifikation

In diesem Kapitel werden zunächst die gestellten Anforderungen aus den Kapiteln 4.1, 4.2 und 4.3 geprüft. Anschließend wird das Gesamtsystem getestet und abschließend ein Fazit gezogen.

## 6.1. Überprüfung der Anforderungen

### 6.1.1. Hardware

Die Anforderungen an die Hardware sind in der Konzeption (Abschnitt 4.3) beschrieben. Es werden ein Raspberry Pi Starter Set, die Anzeigesysteme und eine WLAN Bridge erworben. Die Anzeigesysteme bestehen aus einem Fernseher mit Windows 10 Rechner als Quelle und einem Tablet zur mobilen Anzeige. Die Auflistung der Gesamtkosten zeigt die Tabelle 6.1. Betrachtet man die Gesamtkosten in der Tabelle, ist zu sehen, dass das zur Verfügung stehende Budget von 3.500€ nicht überschritten wird.

Tabelle 6.1.: Auflistung der Gesamtkosten der Hardware [33]

Position	Anbieter	Preis
Raspberry Pi Starter Set	Reichelt	64,50€
Dauerhaftes Anzeigesystem	Conrad und Cancom	2599,87€
Tablet	Alternate	396,98€
WLAN Bridge	Reichelt	42,90€
Sonstiges (Kabel und Steckdosen)		149,05€
<b>Gesamtkosten</b>		<b>3.253,30 €</b>

### 6.1.2. Visualisierung der Rückmeldedaten

In diesem Unterkapitel werden die realisierten Webseiten beschrieben und mit den Anforderungen aus der Konzeption (Abschnitt 4.1.1) verglichen.

Es werden zwei Webseiten zur Visualisierung der aktuellen Daten und zwei zur Darstellung der historischen Daten entwickelt. Bei eingeschaltetem System ist dauerhaft eine der Leitstandseiten in den Räumlichkeiten der Modellfabrik sichtbar. Dabei kann zwischen der aktuellen und historischen Seite gewählt werden. Über die mobile Webseite lassen sich auf dem Tablet entweder die aktuelle oder historische Seite betrachten. Damit sind die Anforderungen 5, 6 und 7 erfüllt.

Die Rückmeldedaten auf den aktuellen und historischen Webseiten werden in mehreren Durchlaufdiagrammen und in tabellarischer Form dargestellt. Die Durchlaufdiagramme und Tabellen bilden dabei verschiedene Schnitte und Detailtiefen ab. Mit Betrachtung aller Durchlaufdiagramme einer Detailtiefe lässt sich der Produktionsablauf vom Einlasten bis zum Auslasten abbilden. Für die intuitive und benutzerfreundliche Bedienung werden verschiedene Möglichkeiten zur Navigation zur Verfügung gestellt. Die Webseiten erfüllen damit die Anforderungen 1 bis 4.

Die Webseiten in den Räumlichkeiten der Modellfabrik und auf dem Tablet führen beide zu einer Startseite. Über die Startseite können Informationen zur Modellfabrik abgerufen werden. Zusätzlich kann über Hyperlinks zu den aktuellen und historischen Daten navigiert werden und der Raspberry gesteuert werden. Die erstellte Startseite erfüllt somit die Anforderung 8.

Die zuvor genannten Punkte zeigen, dass sämtliche Anforderungen aus Kapitel 4.1.1 erfüllt werden.

### 6.1.3. Datenbereitstellung und -verarbeitung

In der Konzeption (Abschnitt 4.2.1) werden Anforderungen an die Datenbereitstellung und -übertragung gestellt. Anhand dieser wird sich in dem Abschnitt 4.2.4 für einen Lösungsansatz entschieden und dieser realisiert. Die Anforderungen 1 bis 6 sowie 8 bis 10 werden in dem genannten Abschnitt bereits bewertet. Es bleibt die Erfüllung der Anforderungen 7 und 11 bis 13 zu bewerten.

Die Anforderung 12 setzt voraus, dass die Oberfläche von vielen Endgeräten nutzbar ist. Anforderung 13 fordert, dass möglichst keine Zusatzsoftware zur Darstellung der graphischen Oberfläche benötigt wird. Beide Anforderungen werden durch die Erstellung der Oberflächen mittels Webseiten erfüllt. Jedes Endgerät mit Webbrowser kann die Webseiten darstellen.

Die Anforderung 7 beinhaltet die Übertragung der Rückmeldedaten zunächst über das vorhandene Bussystem und anschließend durch ein WLAN Netzwerk. Mit Anforderung 11 wird die Darstellung der Live-Daten nach spätestens 5 Sekunden gefordert. Diese Anforderungen können nur durch einen Gesamttest verifiziert werden. Dieser wird nachfolgend (Unterkapitel 6.2) beschrieben.

## 6.2. Test des Gesamtsystems

Zum Testen des Gesamtsystems werden nacheinander verschiedene Szenarien gestartet. Dies dokumentiert die korrekte Übertragung, Berechnung und Darstellung der Daten. Zur besseren Übersicht werden nach den einzelnen Tests die Zähler der Rückmeldedaten auf 0 zurückgesetzt. Die Betrachtung der Daten ist also immer nur für den aktuellen Test möglich.

### 6.2.1. Normalbetrieb

Zur Verifikation der Funktion des Gesamtsystems im Normalbetrieb wird auf das Video auf der angefügten DVD verwiesen. Dies zeigt die Fertigung eines einpoligen Produktes und parallel die Darstellung der Rückmeldedaten auf der aktuellen Leitstandseite mit der Detailtiefe *sechs Schnitte*. Die verwendete Platine ist elektrisch in Ordnung und der Oberbau wird korrekt, wie in Abbildung 3.1 gezeigt, aufgedrückt. Das Produkt durchläuft den Prozess, wie in der Anlagenbeschreibung (Unterkapitel 3.2) beschrieben. Außerdem werden die in der Realisierung (Unterkapitel 5.1) beschriebenen Annahmen zur Modellfabrik eingehalten.

Mit diesem Video wird die Darstellung der Rückmeldedaten und die Funktion der Leitstandseiten ersichtlich. Außerdem können durch diesen Test die Anforderungen 5 (keine Daten verpassen) und 11 (Daten nach spätestens 5 Sekunden darstellen) aus der Konzeption (Abschnitt 4.2.1) für diesen Fall verifiziert werden. Das Video zeigt, dass die Darstellung der Daten weniger als 5 Sekunden dauert und alle erwarteten Daten angezeigt werden. Zusätzlich wird aus dem Video die korrekte Berechnung der darzustellenden Rückmeldedaten und die Darstellung der Durchlaufdiagramme und Tabellen sowohl auf der aktuellen (Zeitraum im Video 00:00-03:30) wie auch auf der historischen Leitstandseite (Zeitraum im Video 3:30-05:12) ersichtlich. Die Funktion ist auf den mobilen Seiten ebenfalls gegeben. Zum Schluss kann mit dem Video die Berechnung der Durchlaufzeit verifiziert werden. Betrachtet man den Zeitpunkt 00:12 (Zugang Bestückung) und den Zeitpunkt 01:00 (Abgang Bestückung) im Video, so wird daraus ersichtlich, dass die angegebene letzte Durchlaufzeit von 48 Sekunden korrekt ist. Dies ist auch für die anderen Schnitte möglich.

### 6.2.2. Elektrischer Fehler

Im zweiten Schritt werden ein ein- und ein zweipoliges Produkt in Auftrag gegeben. Allerdings werden drei ein- und drei zweipolige defekte Platinen zur Bestückung verwendet. Im Anschluss werden *el. OK* Platinen zur Bestückung verwendet. Da es in diesem Test um die Reaktion der Webseite auf elektrische Fehler geht, wird die Produktion der Gutteile nicht weiter betrachtet.

Das als erstes an der Funktionsprüfung geprüfte *el. NOK* Zwischenprodukt durchläuft den Prozess als in der Anlagenbeschreibung (Unterkapitel 3.2) erläutert. Es wird in dem Schnitt Presse kein Oberbau aufgedrückt. Das Zwischenprodukt wird in der Endkontrolle demonstert und der Unterbau im Zwischenlager abgelegt. Da nach dem Einlagern des Unterbaues noch Produkte zu bestücken sind, wird der Unterbau wieder aus dem Zwischenlager entnommen und erneut bestückt. Die Abbildung 6.1 zeigt die Tabellen der Endkontrolle (a) und der Bestückung (b) zum Zeitpunkt, an dem der Unterbau aus dem Zwischenlager auf TB 6 transportiert wird. Die Tabelle der Endkontrolle zeigt nun ein *el.NOK* Produkt und ein wieder eingelasteten Unterbau an. Die Tabelle Bestückung zeigt zusätzlich zu den vier bisherigen Unterbauten aus dem Unterbaulager auch einen Unterbau aus der Produktion an.

Bestand	Durchlaufzeit	NOK	Opt. Fehler	Wiedereinlastung Unterbauten	Ausschuss
Kapazität: 5	Planzeit: 00:38	El. NOK: 1	A: 0	Gesamt: 1	El. Ausschuss: 0
Gesamt: 1	Letzte: 00:00:23	Opt. NOK: 0	B: 0		Opt. Ausschuss: 0
		Gesamt: 1	C: 0		

(a) Endkontrolle

Kapazität	Bestand	Durchlaufzeit	Verwendete Unterbauten
1 Pol: 1	1 Pol: 1	Planzeit: 00:50	Aus Unterbau-Lager: 4
2 Pol: 1	2 Pol: 0	Letzte: 00:00:50	Aus Produktion: 1
Zwischenlager: 4	Zwischenlager: 0		Gesamt: 5
Gesamt: 6	Gesamt: 1		

(b) Bestückung

Abbildung 6.1.: Tabellen der relevanten Schnitte nach Wiedereinlastung eines Unterbaus

Die folgenden fünf Zwischenprodukte, die alle *el. NOK* sind, werden nach der Funktionsprüfung der Station 40 in dem Schnitt Funktionsprüfung aufgestaut. Nachdem alle *el. NOK* Produkte an der Station 40 geprüft wurden, werden die gestauten Produkte wieder losgelassen und betreten mit einer kleinen zeitlichen Verzögerung, die durch die Steuerung vorgegeben ist, den Schnitt Presse. Hier werden die Zwischenprodukte alle durchgeschleust und betreten anschließend nacheinander den Schnitt Endkontrolle. In der Endkontrolle werden die ersten vier Zwischenprodukte demonstert und die Unterbauten eingelagert. Es ist kein weiterer Unterbau mehr zu bestücken. Deshalb verbleiben die vier Unterbauten im Zwischenlager.

Das fünfte Zwischenprodukt, das in der Endkontrolle ankommt, wird ausgeschoben. Damit wird es zum elektrischen Ausschuss, da die Kapazität des Zwischenlagers ausgeschöpft ist. Die Abbildung 6.2 zeigt die Tabellen der Schnitte Funktionsprüfung (a), Endkontrolle (b) und Bestückung (c) nach dem Ausschieben des el. NOK Produktes. Die Tabelle Funktionsprüfung (a) zeigt die Prüfergebnisse der sechs el. NOK Produkte und zusätzlich die Ergebnisse der zwei el. OK Produkte an. Die Endkontrolle (b) zeigt zum einen einen el. Ausschuss und zum anderen die Wiedereinlastung von insgesamt fünf Unterbauten, von denen noch vier im Bestand des Zwischenlagers im Schnitt Bestückung (c) liegen.

Bestand	Durchlaufzeit	Prüfergebnis
Kapazität: 9	Planzeit: 00:44	El. OK: 2
Gesamt: 0	Letzte: 00:00:44	El. NOK: 6
		Gesamt Geprüft: 8

(a) Funktionsprüfung

Bestand	Durchlaufzeit	NOK	Opt. Fehler	Wiedereinlastung Unterbauten	Ausschuss
Kapazität: 5	Planzeit: 00:38	El. NOK: 6	A: 0	Gesamt: 5	El. Ausschuss: 1
Gesamt: 1	Letzte: 00:00:46	Opt. NOK: 0	B: 0		Opt. Ausschuss: 0
		Gesamt: 6	C: 0		

(b) Endkontrolle

Kapazität	Bestand	Durchlaufzeit	Verwendete Unterbauten
1 Pol: 1	1 Pol: 0	Planzeit: 00:50	Aus Unterbau-Lager: 7
2 Pol: 1	2 Pol: 0	Letzte: 00:00:50	Aus Produktion: 1
Zwischenlager: 4	Zwischenlager: 4		Gesamt: 8
Gesamt: 6	Gesamt: 4		

(c) Bestückung

Abbildung 6.2.: Tabellen der relevanten Schnitte nach vier Wiedereinlastungen und einem Ausschuss

### 6.2.3. Optischer Fehler

Im letzten Test werden insgesamt sechs Produkte, drei davon ein- und drei zweipolig, in Auftrag gegeben. Die verwendeten Platinen sind alle elektrisch in Ordnung. Die Zwischenprodukte werden allerdings mit fehlerhaften Oberbauten versehen. Die fehlerhaften Endprodukte kommen in der Reihenfolge

- 1x Fehler A (kein Logo)
- 2x Fehler B (Logo um 180° verdreht)
- 3x Fehler C (Logo um  $\pm 90^\circ$  verdreht)

in der optischen Endkontrolle an. Die fehlerhaften Produkte verbleiben im Handarbeitsplatz, bis auch das letzte Endprodukt die optische Endkontrolle durchlaufen hat. Das letzte Endprodukt hat den optischen Fehler C. Da im Handarbeitsplatz schon zwei Produkte mit dem Fehler C liegen und damit die Kapazität ausgeschöpft ist, wird dieses in der Endkontrolle ausgeschoben. Alle in dem Handarbeitsplatz verbleibenden Produkte werden anschließend nachbearbeitet und aus dem Fertigproduktlager entnommen. Die Abbildung 6.3 zeigt die Tabellen der Schnitte Endkontrolle (a), Handarbeit (b) und Fertigprodukt-Lager (c), nachdem alle Produkte entnommen wurden. Die Tabelle Endkontrolle zeigt die Erkennung von sechs optischen Fehlern an. Außerdem können die Fehler einmal A, zweimal B und dreimal C abgelesen, werden. Der optische Ausschuss im Schnitt Endkontrolle zeigt eine 1. In der Tabelle Handarbeit sind die nachbearbeiteten Fehler einmal A, zweimal B und dreimal C abzulesen, und im Fertigprodukt-Lager werden fünf nachbearbeitete Produkte sowie fünf entnommene Fertigprodukte aufgelistet.

Bestand	Durchlaufzeit	NOK	Opt. Fehler	Wiedereinlastung	Unterbauten	Ausschuss
Kapazität: 5	Planzeit: 00:38	El. NOK: 0	A: 1	Gesamt: 0		El. Ausschuss: 0
Gesamt: 0	Letzte: 00:00:38	Opt. NOK: 6	B: 2			Opt. Ausschuss: 1
		Gesamt: 6	C: 3			

(a) Endkontrolle

Kapazität	Bestand	Durchlaufzeit	Nachbearbeitete Fehler
Fehler A: 2	Fehler A: 0	Planzeit: 00:20	A: 1
Fehler B: 2	Fehler B: 0	Letzte: 00:00:27	B: 2
Fehler C: 2	Fehler C: 0		C: 2
Gesamt: 6	Gesamt: 0		Gesamt: 5

(b) Handarbeit

Bestand	Durchlaufzeit	Gefertigt	Entnommene Fertigprodukte
Kapazität: 12	Planzeit: 00:17	1 Pol Gut: 0	Gesamt: 5
Gesamt: 0	Letzte: 00:00:05	2 Pol Gut: 0	
		Nachbearbeitet Handarbeit: 5	
		Gesamt: 5	

(c) Fertigprodukt-Lager

Abbildung 6.3.: Tabellen der relevanten Schnitte nach Entnahme der Fertigprodukte

### 6.2.4. Fazit

Die drei Tests Normalbetrieb, Elektrischer Fehler und Optischer Fehler können fehlerfrei durchgeführt werden. Die korrekte Berechnung und Darstellung der Rückmeldedaten ist mit diesen Tests verifiziert. Zusätzlich kann die Anforderung 11 aus dem Abschnitt 4.2.1 als erfüllt angesehen werden. Das Video im Anhang zeigt, dass die Rückmeldedaten rechtzeitig dargestellt werden. Die Übertragung mittels vorhandenen Bussystem und WLAN (Anforderung 7, Abschnitt 4.2.1) ist ebenfalls erfüllt, da der Betrieb des Systems mit beiden Übertragungswegen möglich ist.

# 7. Zusammenfassung und Ausblick

## 7.1. Zusammenfassung

In der Industrie spielt die Verbindung der Unternehmensleitebene und Fertigungsebene eine immer wichtiger werdende Rolle. Um Aufträge effizient und flexibel planen und fertigen zu können, wird eine Verbindungsebene zwischen den Fertigungsmaschinen und Planungstools wie z.B. Enterprise-Resource-Planing (ERP) gefordert. Diese fehlende Verbindung kann durch so genannte Manufacturing Execution Systems (MES) geschlossen werden. Aufgabe eines MES ist unter anderem die Erfassung von betrieblichen Rückmeldedaten wie z.B. der Bestand in der Anlage oder in einem Teilabschnitt und die Durchlaufzeit.

Im Fachbereich Automatisierungstechnik der HAW Hamburg befindet sich eine flexible Fertigung (Modellfabrik), deren Rückmeldedaten bisher nicht erfasst wurden. Da diese Daten, wie oben beschrieben, jedoch in der Industrie immer wichtiger werden, wurde im Rahmen dieser Masterarbeit ein System zur Erfassung und Darstellung der Rückmeldedaten entwickelt und realisiert. Dafür wurden zunächst verschiedene Konzepte betrachtet, ein Gesamtkonzept erarbeitet und dieses schließlich realisiert.

Für die Erfassung der Rückmeldedaten wurden die SPS-Programme der Modellfabrik um einen Programmbaustein erweitert. In diesem werden die gewünschten Rückmeldedaten aufgenommen. Die Steuerungen wurden auf die Firmware 2.0 aktualisiert, um den OPC UA Server nutzen zu können.

Die erfassten Daten werden unter Nutzung des OPC UA Protokolls an einen Raspberry Pi (Raspberry) übertragen. Die Übertragung erfolgte zunächst über das vorhandene Bussystem (Konzept 1). Anschließend wurde der Raspberry als WLAN-Router eingerichtet. Die Daten werden nun über WLAN übertragen (Konzept 2). Für den Raspberry wurde ein Pythonprogramm entwickelt, das die OPC UA Daten empfängt und nach der Bearbeitung in eine Datenbank speichert. Die Datenbank wurde zuvor auf dem Raspberry installiert und eingerichtet.

Außerdem wurde auf dem Raspberry der Apache 2 Webserver installiert. Auf diesem wurden die Webseiten zur Darstellung der Rückmeldedaten entwickelt und gespeichert. Zum einen wurden eine aktuelle und eine historische Webseite zur dauerhaften Darstellung in den Räumlichkeiten der Modellfabrik implementiert. Zum anderen wurden zwei Seiten entwickelt, um die aktuellen und historischen Daten auf mobilen Endgeräten zu betrachten. Für die dauerhafte Darstellung und die mobile Version wurden zusätzlich je eine Startseite entwickelt und implementiert. Die Startseiten sind über die URL [192.168.33.151/Rückmeldedaten/home\\_m.html](http://192.168.33.151/Rückmeldedaten/home_m.html) (Mobil) und [192.168.33.151/Rückmeldedaten/home\\_1.html](http://192.168.33.151/Rückmeldedaten/home_1.html) (Leitstand) erreichbar.

Das Ziel dieser Arbeit war die Erfassung und Darstellung betrieblicher Rückmeldedaten in Durchlaufdiagrammen. Zusätzlich sollten weitere Rückmeldedaten wie z.B. der Bestand und die Durchlaufzeit erfasst werden. Dieses Ziel ist mit dem entwickelten Gesamtsystem erfüllt. Die Webseiten stellen die zuvor erfassten und übertragenen Rückmeldedaten in Durchlaufdiagrammen sowie in tabellarischer Form dar. In der historischen Ansicht werden die in der Datenbank verfügbaren Daten der letzten 12 Monate dargestellt. Innerhalb dieser Daten ist die Betrachtung des mittleren Bestandes und der mittleren Durchlaufzeiten für beliebige Zeiträume möglich. Mit den aktuellen Daten kann die derzeitige Fertigung verfolgt werden. Dadurch können Probleme innerhalb der Fertigung frühzeitig erkannt werden.

Abschließend wurde die Funktion des Gesamtsystems in geeigneten Testszenarien überprüft. Diese zeigen den fehlerfreien Betrieb des Systems.

## 7.2. Ausblick

Diese Arbeit bildet den Grundstein für weiterführende Projekte. Der Betrachtungszeitraum der Durchlaufdiagramme der historischen Daten wird zur Zeit für jedes Diagramm einzeln eingestellt. Die Durchlaufdiagramme zeigen also nicht zwingend alle den selben Zeitraum. Bei Bedarf könnte zusätzlich der Zeitraum für alle Durchlaufdiagramme angepasst werden, wenn sich ein Betrachtungszeitraum ändert. Zwischen den zwei Varianten könnte dann durch eine Checkbox gewählt werden.

Des Weiteren könnten erfasste Rückmeldedaten zusätzlich zu den Durchlaufdiagrammen durch weitere Hilfsmittel wie z.B. Histogramme dargestellt werden. Diese Auswertung könnte die Streuung der Durchlaufzeiten aufzeigen.

Außerdem könnten die Rückmeldedaten zu einer Analyse und Optimierung des Prozesses genutzt werden. Die Auswertung könnte Aufschluss über Bottlenecks geben. Verschiedene Konzepte zur Prozessoptimierung könnten getestet und mit der in dieser Arbeit erstellten visuellen Auswertung gegenüber gestellt werden.

Durch die Wahl, die Daten auf einer Webseite darzustellen, könnte diese auf der HAW Webseite veröffentlicht werden. Dafür müsste zuvor festgelegt werden, welche Daten und Steuerungsfunktionen veröffentlicht und für jeden Nutzer zugänglich sein sollen. Es sind eine Risikoanalyse bezüglich Angriffen durchzuführen und Sicherheitsaspekte zu betrachten. Die veröffentlichte Webseite könnte in verschiedene Bereiche aufgeteilt werden. Es könnte einen allgemein zugänglichen Bereich für interessierte Nutzer geben. Dies wäre eine gute Möglichkeit, auf die Modellfabrik und den Fachbereich Automatisierungstechnik aufmerksam zu machen. Zusätzlich wäre ein Bereich z.B. für Studierende, die einen Laborversuch an der Anlage durchführen, möglich. Hier könnten sie sich bereits vor dem Laborversuch mit der Anlage auseinandersetzen. Die Durchlaufdiagramme könnten in diesem Fall zur Auswertung von Laboraufgaben genutzt werden.

Ein weiteres Potenzial liegt in der Erweiterung dieses MES ähnlichen Systems. Es könnte weitere Aufgaben eines MES wie z.B. die Steuerung des Prozesses erfüllen. OPC UA kann sowohl Daten vom Server lesen als auch Variablen setzen. Dies bietet die Möglichkeit, steuernd in den Prozess einzugreifen. So könnte z.B. der maximal mögliche Bestand in einem Schnitt oder der Gesamtanlage durch eine Eingabe verringert werden, um die Durchlaufzeiten zu verkürzen.

Abschließend wäre ein Benachrichtigungssystem denkbar. In der Industrie informieren solche Systeme die verantwortlichen Mitarbeiter über SMS oder e-Mail über dauerhafte Kapazitätsauslastungen oder -unterschreitungen sowie bei erhöhten Durchlaufzeiten. Durch die Benachrichtigungen können somit Fehler der Produktion oder Engpässe frühzeitig erkannt und behoben werden. Für die in der Lehre genutzte Modellfabrik wäre eine Warnung, die auf dem Anzeigesystem ausgegeben wird, sinnvoll. Diese könnte bei leeren Lagern oder nicht korrekt eingeschalteten Systemen, wie z.B. dem Roboter, ausgegeben werden.

# Literaturverzeichnis

- [1] Das Raspberry Pi 3 B+ Black-Bundle. [https://www.reichelt.de/das-raspberry-pi-3-b-black-bundle-rpi3-bdl-bf-2017-p211192.html?&trstct=pos\\_0](https://www.reichelt.de/das-raspberry-pi-3-b-black-bundle-rpi3-bdl-bf-2017-p211192.html?&trstct=pos_0) (Stand: 15.10.2018).
- [2] Free OPC-UA Library. <https://github.com/FreeOpcUa> (Stand: 25.07.2018).
- [3] Geany. <https://geany.org/> (Stand: 21.02.2019).
- [4] Highcharts. <https://www.highcharts.com/> (Stand: 22.10.2018).
- [5] Installation eines Raspberry Pi mit einem Webserver inkl. Php, MySQL & FTP. <https://www.intelligentes-haus.de/tutorials/smart-home-tutorials/installation-eines-raspberry-pi-mit-einem-webserver-inkl-php-mysql-ftp/> (Stand: 09.01.2019).
- [6] jQuery. <https://jquery.com/> (Stand: 25.10.2018).
- [7] Microsoft Expression Web 4. <https://www.microsoft.com/en-us/download/details.aspx?id=36179> (Stand: 21.02.2019).
- [8] Moments. <https://momentjs.com/> (Stand: 25.10.2018).
- [9] PowerPoint Kiosk-Modus - so funktioniert's. [https://praxistipps.chip.de/powerpoint-kiosk-modus-so-funktioniert\\_27101](https://praxistipps.chip.de/powerpoint-kiosk-modus-so-funktioniert_27101) (Stand: 21.02.2019).
- [10] PyMySQL. <https://pymysql.readthedocs.io/en/latest/index.html> (Stand: 28.07.2018).
- [11] Setting up a Raspberry Pi as an access point in a standalone network (NAT). <https://www.raspberrypi.org/documentation/configuration/wireless/access-point.md> (Stand: 02.10.2018).
- [12] Thonny Python IDE for beginners. <https://thonny.org/> (Stand: 21.02.2019).
- [13] *User Manual WISE-4000 Series*, 2016.

- [14] Siemens AG. Der OPC UA-Standard Ein Meilenstein für die Automatisierung in der digitalen Welt. <https://new.siemens.com/global/de/produkte/automatisierung/industrielle-kommunikation/opc-ua.html> (Stand: 22.01.2019).
- [15] Siemens AG. *Simatic S7-1500 Getting Started*, 2014. [https://www.automation.siemens.com/salesmaterial-as/interactive-manuals/getting-started\\_simatic-s7-1500/documents/DE/software\\_complete\\_de.pdf](https://www.automation.siemens.com/salesmaterial-as/interactive-manuals/getting-started_simatic-s7-1500/documents/DE/software_complete_de.pdf) (Stand: 20.02.2019).
- [16] Siemens AG. *OPC UA .NET Client für den SIMATIC S 7 - 1500 OPC UA Server*, 2017. [https://cache.industry.siemens.com/dl/files/901/109737901/att\\_955024/v1/109737901 OPC UA Client\\_S7-1500\\_DOKU\\_V13\\_de.pdf](https://cache.industry.siemens.com/dl/files/901/109737901/att_955024/v1/109737901 OPC UA Client_S7-1500_DOKU_V13_de.pdf) (Stand: 29.01.2019).
- [17] Siemens AG. *Prozessdaten erfassen und überwachen*, 2017. [https://cache.industry.siemens.com/dl/files/156/64396156/att\\_937401/v1/64396156\\_S7-1200\\_DataLogging\\_DOC\\_v2d0d1\\_de.pdf](https://cache.industry.siemens.com/dl/files/156/64396156/att_937401/v1/64396156_S7-1200_DataLogging_DOC_v2d0d1_de.pdf) (Stand: 22.01.2019).
- [18] ASUS. *Quick Start Guide*, 2017.
- [19] Peter Bühler, Patrick Schlaich, and Dominik Sinner. *Webtechnologien - JavaScript - PHP - Datenbank*. Springer-Verlag, Berlin Heidelberg New York, 1. aufl. 2018 edition, 2018.
- [20] Thomas J. Burke. *OPC Unified Architecture. Wegbereiter der 4. industriellen (R)Evolution*, 2017.
- [21] Tim Geiger. *Industrie 4.0 mit Node-RED und Raspberry Pi*. Hochschule für Angewandte Wissenschaften Hamburg, 2018.
- [22] KÖSTER Systemtechnik GmbH. *Technische Dokumentation der Modellfabrik*, 2016.
- [23] René Kiem. *Qualität 4.0 - QM, MES und CAQ in digitalen Geschäftsprozessen der Industrie 4.0*. Hanser Fachbuchverlag, München, 2016.
- [24] Jürgen Kletti. *MES - Manufacturing Execution System - Moderne Informationstechnologie unterstützt die Wertschöpfung*. Springer-Verlag, Berlin Heidelberg New York, 2. aufl. edition, 2015.
- [25] Jürgen Kletti and Rainer Deisenroth. *MES-Kompodium - Ein Leitfaden am Beispiel von HYDRA*. Springer-Verlag, Berlin Heidelberg, 2012.
- [26] Jürgen Lange, Frank Iwanitz, and Thomas J.Burke. *OPC - Von Data Access bis Unified Architecture*. Vde Verlag GmbH, Berlin, Offenbach, 5. durchges. aufl. edition, 2014.

- 
- [27] Peter Müller. *Einstieg in CSS - Webseiten gestalten mit HTML und CSS*. Rheinwerk Verlag GmbH, Bonn, 2. Aufl. edition, 2016.
- [28] Stefan Müller. *Manufacturing Execution Systeme (MES) - Status Quo und Ausblick in Richtung Industrie 4.0*. BoD - Books on Demand, Norderstedt, 2015.
- [29] Peter Nyhuis and Hans-Peter Wiendahl. *Logistische Kennlinien - Grundlagen, Werkzeuge und Anwendungen*. Springer-Verlag, Berlin Heidelberg New York, 3. Aufl. edition, 2012.
- [30] Valentin Plenk. *Angewandte Netzwerktechnik kompakt - Dateiformate, Übertragungsprotokolle und ihre Nutzung in Java-Applikationen*. Springer-Verlag, Berlin Heidelberg New York, 1. Aufl. 2017 edition, 2017.
- [31] Mareike Sauermann. *Design und Implementierung einer Schnittstelle zur automatisierten Aufnahme von Gantt- und Durchlaufdiagrammen für eine Flaschenabfüllung*. Technische Universität Hamburg-Harburg, 2004.
- [32] Thomas Theis. *Einstieg in PHP 7 und MySQL*. Rheinwerk, Bonn, 13. Aufl. edition, 2018.
- [33] Karin Volkmann. *Excel Tabelle Kostenaufstellung*, 2018.
- [34] Prof. Dr.-Ing. Florian Wenck. Eine vernetzte Montageanlage für die Automatisierungstechnik. *EForum*, pages 6–11, 2016/2017.
- [35] Elisabeth Wetsch. *Einstieg in CSS - Grundlagen und Praxis*. Galileo Press, Bonn, 2. Aufl. edition, 2010.

# **A. Anhang**

## **A.1. Bildschirmpräsentation Mobil und Leitstand (DVD)**

Der Anhang dieser Masterarbeit befindet sich auf der beiliegenden DVD, die beim Erst- und Zweitprüfer eingesehen werden kann.

## **A.2. Anleitung Inbetriebnahme der Modellfabrik (DVD)**

Der Ordner *Anleitung Inbetriebnahme der Modellfabrik* beinhaltet die Anleitung zur Inbetriebnahme der Modellfabrik.

## **A.3. Passwörter (DVD)**

Der Ordner *Passwörter* beinhaltet eine Textdatei, in der die in dieser Arbeit vergebenen Passwörter gespeichert sind.

## **A.4. Konfiguration des Anzeigesystems (DVD)**

Der Ordner *Konfiguration des Anzeigesystems* beinhaltet die Beschreibung der Konfiguration der Anzeigesysteme.

## **A.5. TIA-Projekt der Modellfabrik (DVD)**

Der Ordner *TIA-Projekt der Modellfabrik* beinhaltet das archivierte Steuerungsprojekt der Modellfabrik.

## **A.6. Quellcode Webseiten und Python (DVD)**

Der Ordner *Quellcode Webseiten und Python* beinhaltet die Quelldateien zur Darstellung der Webseite. Außerdem sind die Quelldateien des Pythonprogramms hier abgelegt.

## **A.7. Tabellen und Listen (DVD)**

Der Ordner *Tabellen und Listen* beinhaltet eine PDF-Datei, in der Tabellen des Steuerungsprogrammes, OPC UA und der Datenbank gespeichert sind.

### **A.7.1. Steuerungsprogramm**

### **A.7.2. OPC UA**

### **A.7.3. Datenbank**

## **A.8. Video Normalbetrieb (DVD)**

Der Ordner *Video Normalbetrieb* beinhaltet ein Video zur Darstellung der Webseiten im Betrieb.

# Versicherung über die Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §16(5) APSO-TI-BM ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

Hamburg, 6. März 2019

Ort, Datum

Unterschrift