

Requirements / Design and Test Dokumentation

(RDT)

Version 0.9

ESEP – Praktikum – SoSe 2024

Team – ESEP-1.2

Mjawaz, Yousef, 2636830, yousef.mjawaz@haw-hamburg.de
Althiab, Hadi, 2637134., hadi.althiab@haw-hamburg.de
Shaya, Hussam, 2552404., hussam.shaya@haw-hamburg.de
Shahrour, 2587833, samer.shahrour@haw-hamburg.de
Khalifeh, Kamal, 2608231, kamal.khalifeh@haw-hamburg.de

Inhaltsverzeichnis:

1	Teamorganisation	4
1.1	Verantwortlichkeiten.....	4
1.2	Absprachen.....	4
1.3	Repository-Konzept	4
2	Projektmanagement	5
2.1	Prozess.....	5
2.2	Projektorganisation	5
2.3	Risiken	5
2.4	Qualitätssicherung.	5
3	Problemanalyse	5
3.1	Analyse des Kundenwunsches	5
3.1.1	Stakeholder	5
3.1.2	Systemkontext des Systems	5
3.1.3	Anforderungen.....	6
3.1.4	Use Cases	8
3.2	Anlage: Analyse der technischen Gegebenheiten.....	9
3.2.1	Technischer Aufbau und Hardwarekomponenten	9
3.2.2	Werkstücke	10
3.2.3	Anforderungen aus dem Verhalten und technischen Besonderheiten	10
3.3	Softwareebene	11
3.3.1	Systemkontext der Software	11
3.3.2	Resultierende Anforderungen an die Software	11
3.3.3	Schnittstellen: Nachrichten und Signale	11
	Grobkonzept des technischen Systementwurfes.....	14
4	Software-Design.....	14
4.1	Software Architektur	14
4.2	Software Struktur	15
4.3	Verhaltensmodellierung.....	16
5	Implementierung: Besonderheiten	17
6	Qualitätssicherung.....	18
6.1	Teststrategie	18
6.2	Testszenarien/Abnahmetest.....	18
6.3	Testprotokolle und Auswertungen.....	21

7	Technische Schulden.....	22
8	Lessons Learned	22
9	Anhang.....	22
9.1	Glossar.....	22
9.2	Abkürzungen	22
9.3	Beantwortete Fragen	22

Teamorganisation

1.1 Verantwortlichkeiten

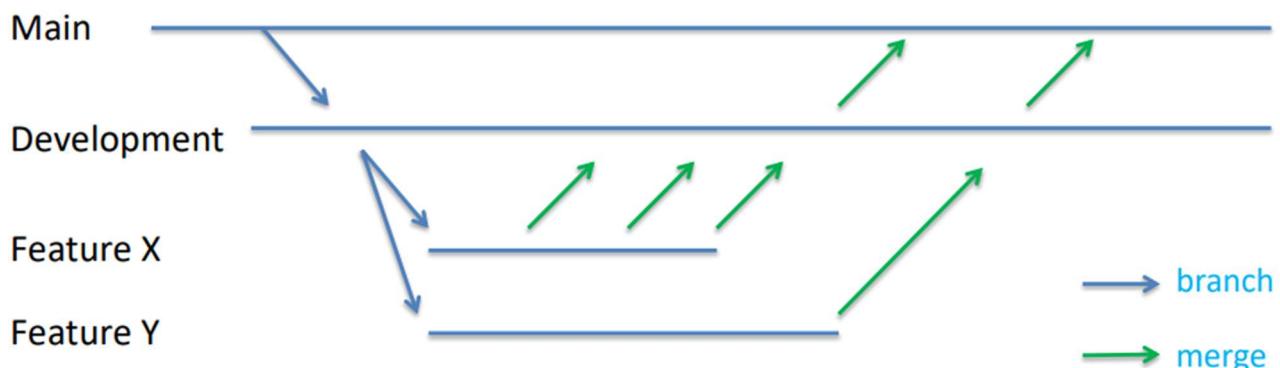
Verantwortlichkeit	Person/en
Tester	Kamal
FSMs	Hussam
Softwarearchitektur	Hadi, Hussam
Qualitätssicherung	Samer, Yousef
Implementierung	Samer, Yousef

1.2 Absprachen

Listen Sie hier die Absprachen im Team auf, z. B. Jour Fixe, Kommunikation, Response-Latenz,

- Sonntags Online-Meeting via Discord ab 11 Uhr
- Terminabsagen müssen frühzeitig in der WhatsApp-Gruppe angekündigt werden.

1.3 Repository-Konzept



Wie in der Vorlesung gezeigt: Es wird ein Main Branch geben, auf dem eine lauffähige Version zu finden ist. Ein anderer Branch ist der Development Branch, auf dem Features und deren Integration getestet werden.

Diese beiden Branches sind online im Git-Repository verfügbar. Es werden aber auch andere lokale Branches vorhanden sein, auf denen Features aktiv implementiert und überarbeitet werden. Wenn ein Feature fertig implementiert wurde, dann wird dieser auf den Development Branch gemerged.

2 Projektmanagement

2.1 Prozess

2.2 Projektorganisation

2.3 Risiken

2.4 Qualitätssicherung.

- Beim Bearbeiten dieses Dokuments benutzt jedes Teammitglied eine andere Farbe für seinen selbstgeschriebenen Text. Erst wenn die Mehrheit der Teammitglieder diesen Text gelesen hat und mit dem Inhalt einverstanden ist, wird dieser Text in die Farbe schwarz umgewandelt.
- Es wurde auf eine Sprache beim Codieren geeignet: Englisch.

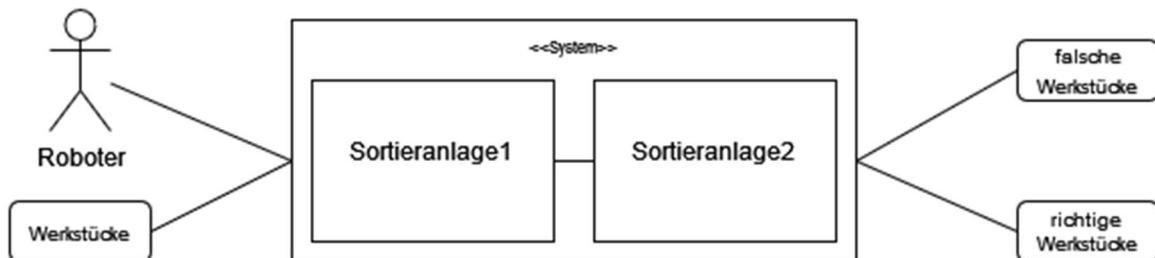
3 Problemanalyse

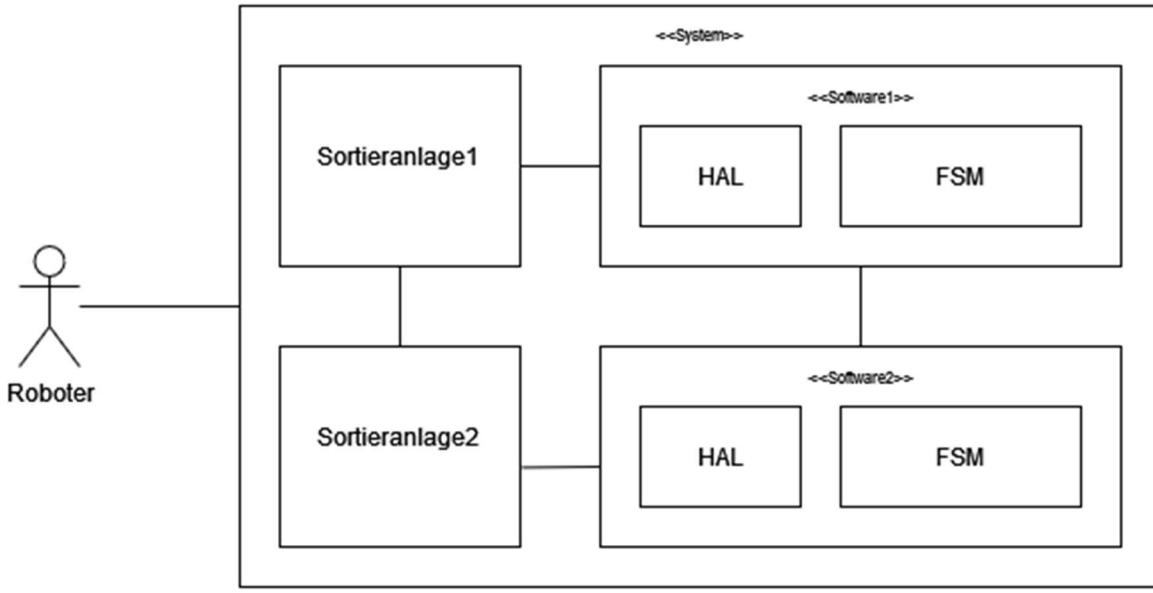
3.1 Analyse des Kundenwunsches

3.1.1 Stakeholder

Stakeholder	Interessen
HAW	Die Anlagen sollen nicht kaputt gehen
Prof. Lehmann/ Enrico	Die Aufgabe muss erfüllt sein. Damit das Team die PVL bekommt.

3.1.2 Systemkontext des Systems



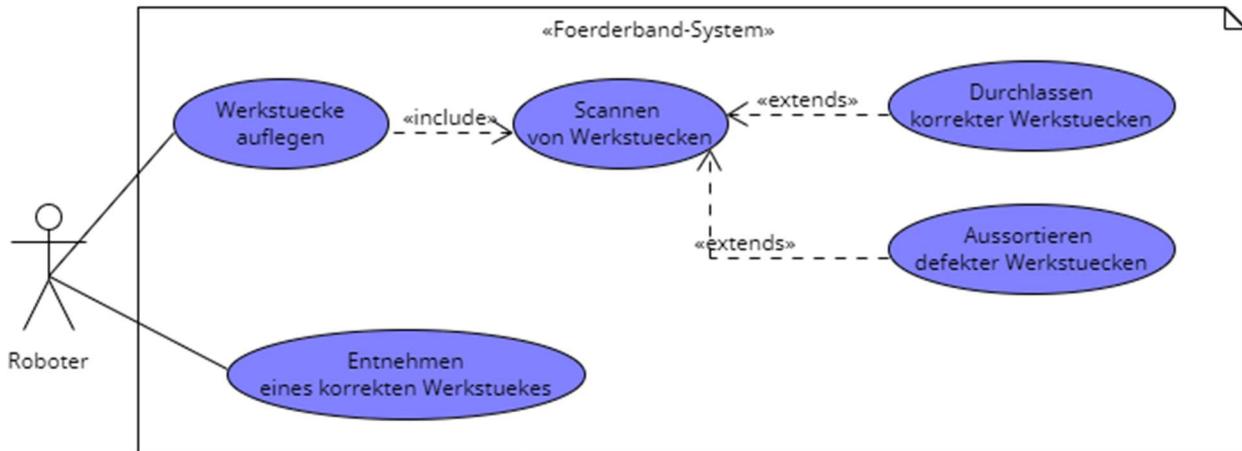


3.1.3 Anforderungen

Nr. / ID	Beschreibung
R1: Sortieren	<p>R1.1: Das System muss WSs voneinander unterscheiden können.</p> <p>R1.2: Reihenfolge: mit Bohrung mit Metall → mit Bohrung mit Metall → flaches Werkstück (12)</p> <p>R1.3: Aussortierte WSs (defekte oder nicht in der richtigen Reihenfolge) fallen in die Rutsche. (16,18)</p> <p>R1.4: Wenn FB1 nicht aussortieren kann, dann soll auf FB2 versucht werden zu sortieren. (17)</p> <p>R1.5: wird auf dem FB2 ein anderes WS oder eine andere Höhe erkannt als auf FB1, so soll das WS aussortiert werden. (19)</p> <p>R1.6: Daten, wie WS-ID, Typ und mittlere Höhe in mm vom WS sollen erfasst werden. (28-31)</p> <p>R1.7: Der Typ von einem WS ist im Regelfall am Ende von FB1 bekannt. (33)</p> <p>R1.8: Das System vergibt WS-ID am Anfang von FB1.(32)</p>
R2: Förderband	<p>R2.1: WSs können gelegt werden, wenn der Anfang vom FB1 frei ist. (21)</p> <p>R2.2: Transport auf beiden Laufbändern erfolgt während Höhenmessung nur langsam. (24)</p> <p>R2.3: Es darf kein WS vom FB fallen. (25)</p> <p>R2.4: Laufbänder stoppen, wenn kein WS darauf ist. (34)</p> <p>R2.5: Laufbänder stoppen, wenn ein Fehler auftritt (Siehe R5).</p> <p>R2.6: FB2 stoppt, wenn ein WS am Ende von FB2 ankommt (WS bleibt auf FB2). (28_0)</p> <p>R2.7: Möglichst höher Durchsatz an WSs soll erreicht werden. (35)</p>

R3: Anzahl WSs	<p>R3.1: Mehrere WSs dürfen sich auf FB1 befinden. (22)</p> <p>R3.2 Es darf sich maximal ein WS auf FB2 befinden. (23)</p>
R4: Ampel, LEDs, Signalisierung	<p>Es muss an der jeweiligen Rutsche signalisiert werden, wenn:</p> <ul style="list-style-type: none"> • R4.1: Betriebzustand läuft (Ampel Grün dauerhaft). (73) • R4.2: Service-Mode gestartet wurde (Ampel Grün blinkend). (74) • R4.3 Warnungen auftreten (Ampel Gelb). (Siehe R5) (75) • R4.4: Fehler auftreten (Ampel Rot). (Siehe R5, R8)
R5: Fehler/Warnung Erfassung	<p>Es gibt einen Fehler, wenn:</p> <ul style="list-style-type: none"> • R5.1: Beide Rutschen voll sind und ein notwendiges Aussortieren ist nicht mehr möglich ist. (56) • R5.2: Ein WS verschwindet. (54) • R5.3: Ein WS wird außerhalb des Anfangsbereiches hinzugefügt. (55) • R5.4: der Mindestabstand zwischen den WSs nicht eingehalten wurde. • R5.5: mehrere WSs sich auf dem FB2 befinden. <p>Es gibt eine Warnung, wenn:</p> <ul style="list-style-type: none"> • R5.6: eine Rutsche voll ist oder wenn beide Rutschen voll sind.
R6: Bedientaster	<p>R6.1: Start: in Betriebszustand wechseln (wenn kurz gedrückt). (62) In Service-Mode wechseln (wenn lange gedrückt) -> Selbsttest/Kalibrierung. (61)</p> <p>R6.2: Stop: in Ruhezustand wechseln (62), wenn kein Fehler oder Warnung (63).</p> <p>R6.3: Reset: Fehlerquittierung. (64)</p> <p>R6.4: E-Stop: Die beiden Anlagen stehen still (65), bis der E-Stopp-Schalter herausgezogen und dann der Reset-Taster gedrückt wurden (67).</p>
R7: Remote Control	<p>R7.1: Die Anlage soll mit einem Remote-System gesteuert und überwacht werden (105).</p> <p>R7.2: Die Kommunikation soll auf MQTT basieren (107).</p> <p>R7.3: Die Remote-Control-Anwendung soll eine graphische Oberfläche haben (108). Entweder eine native Anwendung oder eine Browser Anwendung (109).</p> <p>R7.4: Die Button-Panel beider Anlagen ist in der Oberfläche sichtbar und benutzbar (107).</p> <p>R7.5: Relevante Informationen (z.B. Konsolen ausgeben, volle Rutsche) sollen auch in der Anwendung ausgegeben werden (111-112).</p>
R8: Fehlerzustände	<p>R8.1: Kein Fehler -> Rote Lampe aus (89).</p> <p>R8.2: Wenn ein Fehler neu auftritt, geht die Anlage in den Zustand „Anstehend unquittiert“ -> rote Lampe blinkt schnell (1 Hz) (78). Bis Reset-Taste gedrückt wurde (79)</p> <p>R8.3: Wenn die Anlage im Zustand „Anstehend unquittiert“ ist und der Reset-Taster gedrückt wurde, geht die Anlage in den Zustand „Anstehend quittiert“ -> rote Lampe leuchtet dauerhaft (91).</p> <p>R8.4: Wenn ein Fehler automatisch vom System behoben wird, dann geht die Anlage in den Zustand „Gegangen unquittiert“ -> rote Lampe blinkt langsam (0.5 Hz) (92), bis Reset-Taste gedrückt wurde (85)</p>

3.1.4 Use Cases



Titel	Prüfprozess eines Werkstückes
Akteure	Kunde, WS, Anlage
Ziel	Es wird überprüft, ob ein WS bei LBE2 vom Roboter entnommen werden kann.
Auslöser	Der Kunde legt ein WS bei LBB1
Vorbedingung	beide Anlagen laufen fehlerfrei. Anlagen sind im Betriebszustand, kein WS befindet sich auf FB1/2.
Nachbedingung	Das Werkstück ist entweder aussortiert oder vom Roboter entnommen worden.
Erfolgsszenario	<ol style="list-style-type: none"> 1. Kunde legt ein WS bei LBB1. 2. Das WS bekommt eine ID vom System zugewiesen. 3. WS kommt beim HS an und FB1 wird langsamer 4. HS fängt an, Daten über WS zu erfassen. 5. WS kommt bei LBHS1 an und unterricht diese. 6. Das WS kommt beim MS an. 7. MS fängt an, Daten über WS zu erfassen 8. WS kommt bei LBS1 und unterricht diese. 9. Das System prüft die vorher gemessenen Daten (in Schritt 7 und 4): Sind sie gültig? Wenn weiter mit 10, sonst mit 15. 10. Die Weiche öffnet sich und WS kommt bei LBE1 an und unterricht diese. 11. FB1 teilt FB2 mit, dass ein WS weiter transportiert wird. 12. FB2 läuft und FB1 stoppt, wenn sich keine weiteren WSs drauf befinden. 13. Wiederhole Schritte 3-10 auf Anlage 2 14. FB2 stoppt und Roboter entnimmt das WS -> Ende 15. Weiche/Auswerfer schiebt WS in die Rutsche und LBC wird unterbrochen. 16. Wiederhole Schritte 1-9.

Titel	Prüfung des E-Stopp-Schalters
Akteure	Kunde, WS, Anlage, E-Stopp-Schalter

Ziel	Es wird überprüft, ob alle Anlagen nach Betätigung des E-Stopp-Schalters ordnungsgemäß stillstehen und ob sie nach Betätigung der Reset-Taste an allen Anlagen wieder in den Betriebszustand versetzt werden können.
Auslöser	Der Kunde drückt den E-Stopp-Schalter
Vorbedingung	Alle Anlagen laufen fehlerfrei.
Nachbedingung	Alle Anlagen stehen still, bis der E-Stopp-Schalter herausgezogen und der Reset-Taster an allen Anlagen gedrückt wurde
Erfolgsszenario	<ol style="list-style-type: none"> 1. Kunde zieht zu einem beliebigen Zeitpunkt den E-Stopp-Schalter an einer beliebigen Anlage. 2. Die Anlage, deren Schalter gezogen wurde, informiert die andere Anlage über das Ereignis 3. Alle Anlagen kommen zum Stillstand 4. Alle grünen Lampen gehen aus und die roten Lampen fangen an zu blinken (Mit einer Frequenz von 1Hz) 5. Kunde zieht den E-Stopp-Schalter heraus und die Roten Lampen blinken nun mit einer Frequenz von 0,5 Hz 6. Der Kunde drückt den Reset-Taster an allen Anlagen und die roten Lampen erlöschen. 7. Anlagen wechseln in den Betriebszustand, dabei werden die Grünen Lampen wieder aktiviert.

Titel	WS kommt in die Rutsche
Akteure	Kunde, WS, Anlage
Ziel	Es wird überprüft, ob ein defektes oder unzulässiges WS in die Rutsche kommt.
Auslöser	Der Kunde legt ein defektes oder unzulässiges WS bei LBB1
Vorbedingung	Alle Anlagen laufen fehlerfrei und ein defektes oder unzulässiges WS wird auf dem FB1 gelegt.
Nachbedingung	Das WS kommt in die Rutsche
Erfolgsszenario	<ol style="list-style-type: none"> 1. Der Kunde legt ein defektes oder unzulässiges WS bei LBB1 2. Das WS wird bis HS1 transportiert. 3. Die Datenerfassung fängt an (und das System analysiert sie) 4. Das WS unterbricht LBH1 5. Das WS wird zu MS transportiert und MS prüft, ob das WS ein Metalleinsatz enthält 6. Das WS wird weiter zu LBS1 transportiert 7. Das WS kann zu keinem der vordefinierten Werkstückarten zugeordnet. 8. Die Weiche bleibt geschlossen und WS geht auf die Rutsche

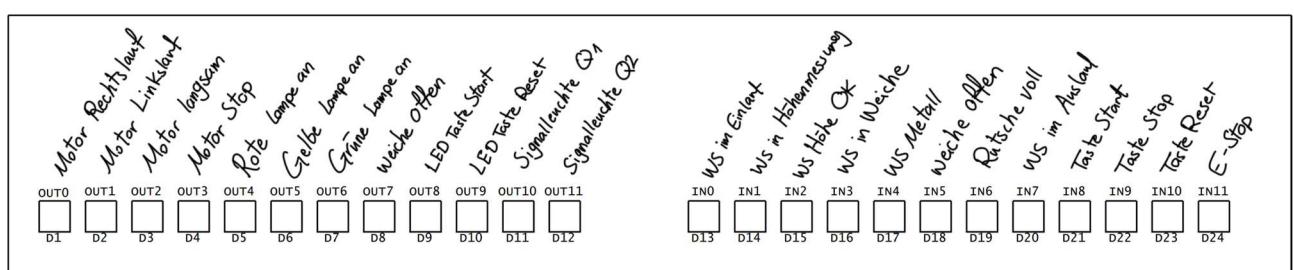
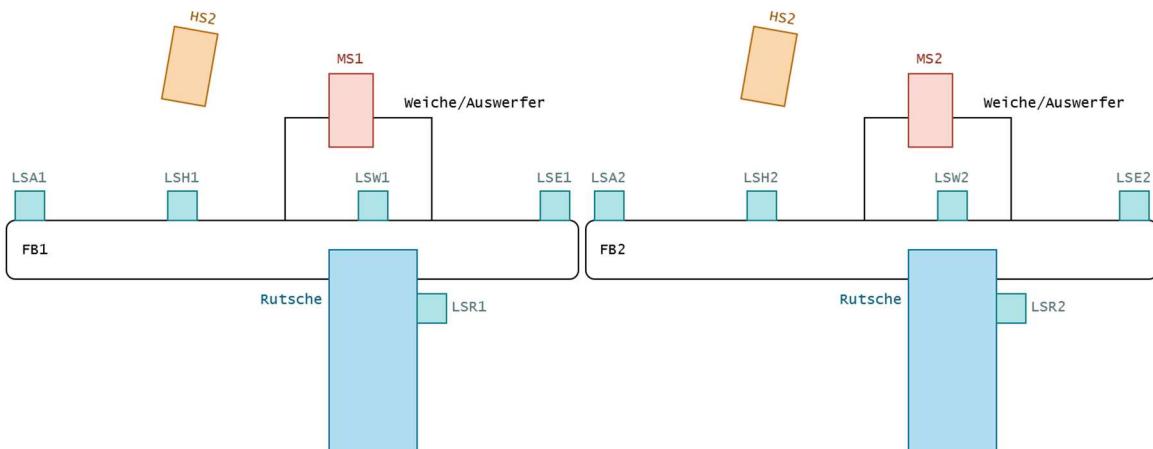
3.2 Anlage: Analyse der technischen Gegebenheiten

3.2.1 Technischer Aufbau und Hardwarekomponenten

Pro Anlage:

- FB: kann vor- und rückwärts fahren.

- Rutsche: beinhaltet Platz für max. 4 Werkstücke
- LEDs:
 - 2 LEDs (Q1, Q2) -> zur Signalisierung, Funktion steht noch nicht fest
 - 2 Taster-LEDs
 - 24 LEDs (auf dem Board) 12 sind für Output und 12 sind für Input gedacht, um die laufenden Funktionen der Anlage anzeigen
- Sensoren (5 LBs, HS, MS): zur Erfassung von Daten über die WKs (Position, Höhe, Metalleinsatz)
 - LBH: ist nicht zuverlässig (kann mit einem Stück Plastik unterbrochen werden kann)
-> darf nicht benutzt werden
 - HS: hat Rauschen von +- ... **TODO**
- Ampel (Rot, Gelb, Grün): zur Signalisierung von Zuständen/Warnungen. Mehrere Lampen können gleichzeitig leuchten.
- Tasten (Start, Stopp, Reset): zur Steuerung der Anlage -> s. R6
- E-Stopp: zum sofortigen Stoppen der Anlagen im Notfall.
- Weiche/Auswerfer: dient der Aussortierung der WSs
-> darf nicht minuten-lang aktiviert sein.



3.2.2 Werkstücke

Um zwischen den Werkstücken zu unterscheiden, werden mehrere Sensoren benutzt wie z.B. Ein HS und ein MS. Die Messdaten müssen vernünftig erfasst, kombiniert und interpretiert werden.

3.2.3 Anforderungen aus dem Verhalten und technischen Besonderheiten

3.3 Softwareebene

Unterschiede zwischen den Anlagen	Anlage 1	Anlage 2
Mehrere WSs gleichzeitig	✓	✗
Vergeben von WS-ID	✓	✗
Überprüfung, ob WS eine ID hat	✗	✓
WS stoppt bei LBE	✓, bis kein WS auf Anlage 2 befindet	✓, bis der Roboter das WS entnimmt

3.3.1 Systemkontext der Software

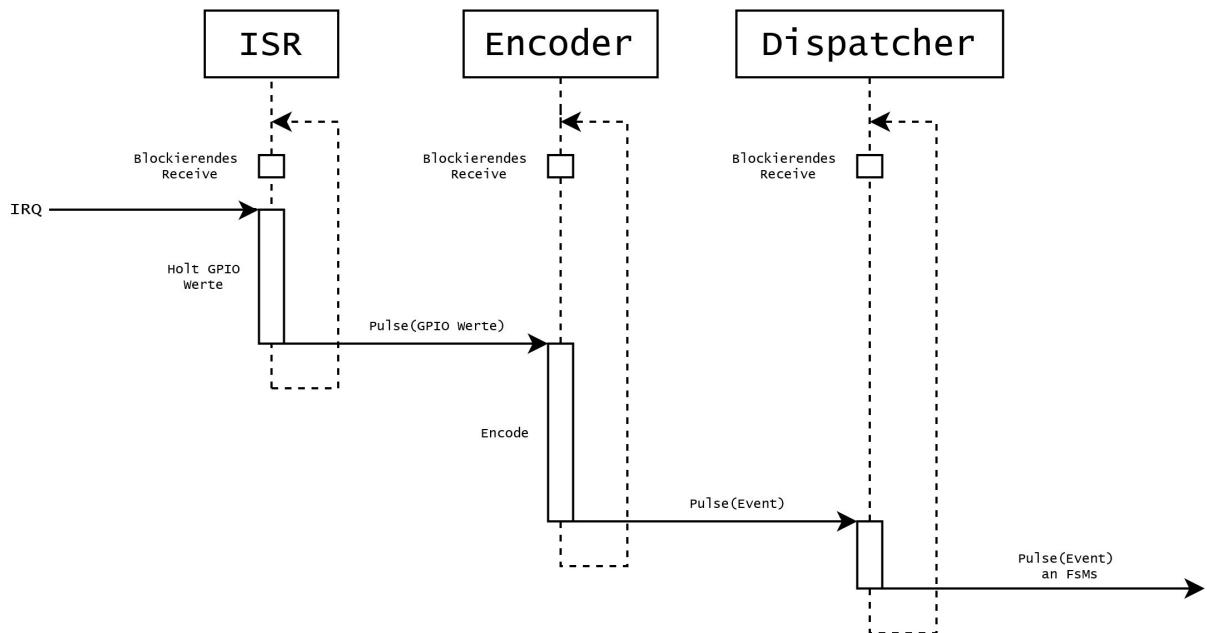
3.3.2 Resultierende Anforderungen an die Software

3.3.3 Schnittstellen: Nachrichten und Signale

Trigger/Events, die durch A1 erzeugt werden können	Bedeutung
M1_LBB_BLOCKED	Anlage 1: Lichtschranke am Anfang blockiert
M1_LBB_FREE	Anlage 1: Lichtschranke am Anfang frei
M1_LBS_BLOCKED	Anlage 1: Lichtschranke an der Weiche blockiert
M1_LBS_FREE	Anlage 1: Lichtschranke an der Weiche frei
M1_LBC_BLOCKED	Anlage 1: Lichtschranke an der Rutsche blockiert
M1_LBC_FREE	Anlage 1: Lichtschranke an der Rutsche frei
M1_LBE_BLOCKED	Anlage 1: Lichtschranke am Ende blockiert
M1_LBE_FREE	Anlage 1: Lichtschranke am Ende frei
M1_START_PRESSED	Anlage 1: Start gedrückt
M1_START_RELEASED	Anlage 1: Start nicht mehr gedrückt
M1_STOP_PRESSED	Anlage 1: Stop gedrückt
M1_STOP_RELEASED	Anlage 1: Stop nicht mehr gedrückt
M1_RESET_PRESSED	Anlage 1: Reset gedrückt
M1_RESET_RELEASED	Anlage 1: Zurücksetzen nicht mehr gedrückt
M1_ESTOP_PRESSED	Anlage 1: Not-Aus gedrückt
M1_ESTOP_RELEASED	Anlage 1: Not-Aus nicht mehr gedrückt
M1_IN_HEIGHT_MEASUREMENT	Anlage 1: In Höhenmessung
M1_LEFT_HEIGHT_MEASUREMENT	Anlage 1: Höhenmessung verlassen
M1_METAL_DETECTED	Anlage 1: Metall erkannt

Trigger/Events, die durch A2 erzeugt werden können	Bedeutung
M2_LBB_BLOCKED	Anlage 2: Lichtschranke am Anfang blockiert
M2_LBB_FREE	Anlage 2: Lichtschranke am Anfang frei
M2_LBS_BLOCKED	Anlage 2: Lichtschranke an der Weiche blockiert

M2_LBS_FREE	Anlage 2: Lichtschranke an der Weiche frei
M2_LBC_BLOCKED	Anlage 2: Lichtschranke an der Rutsche blockiert
M2_LBC_FREE	Anlage 2: Lichtschranke an der Rutsche frei
M2_LBE_BLOCKED	Anlage 2: Lichtschranke am Ende blockiert
M2_LBE_FREE	Anlage 2: Lichtschranke am Ende frei
M2_START_PRESSED	Anlage 2: Start gedrückt
M2_START_RELEASED	Anlage 2: Start nicht mehr gedrückt
M2_STOP_PRESSED	Anlage 2: Stop gedrückt
M2_STOP_RELEASED	Anlage 2: Stop nicht mehr gedrückt
M2_RESET_PRESSED	Anlage 2: Reset gedrückt
M2_RESET_RELEASED	Anlage 2: Zurücksetzen nicht mehr gedrückt
M2_ESTOP_PRESSED	Anlage 2: Not-Aus gedrückt
M2_ESTOP_RELEASED	Anlage 2: Not-Aus nicht mehr gedrückt
M2_IN_HIGHT_MEASUREMENT	Anlage 2: In Höhenmessung
M2_LEFT_HIGHT_MEASUREMENT	Anlage 2: Höhenmessung verlassen
M2_METAL_DETECTED	Anlage 2: Metall erkannt



E-Stop Funktionalität:

Der E-Stop-Interrupt wird genauso wie die anderen Interrupts behandelt (siehe 5.1 Software-Architektur).

E-Stop-spezifische Behandlung:

- Der Encoder prüft das E-Stop-Pin als erstes. So wird die E-Stop-Pulse-Message als erstes an den Dispatcher geschickt.
- Die Pulse-Message des E-Stops hat eine höhere Priorität als die anderen Pulse-Messages und

wird deshalb schneller weitergeleitet und von der FSM schneller ausgewertet.

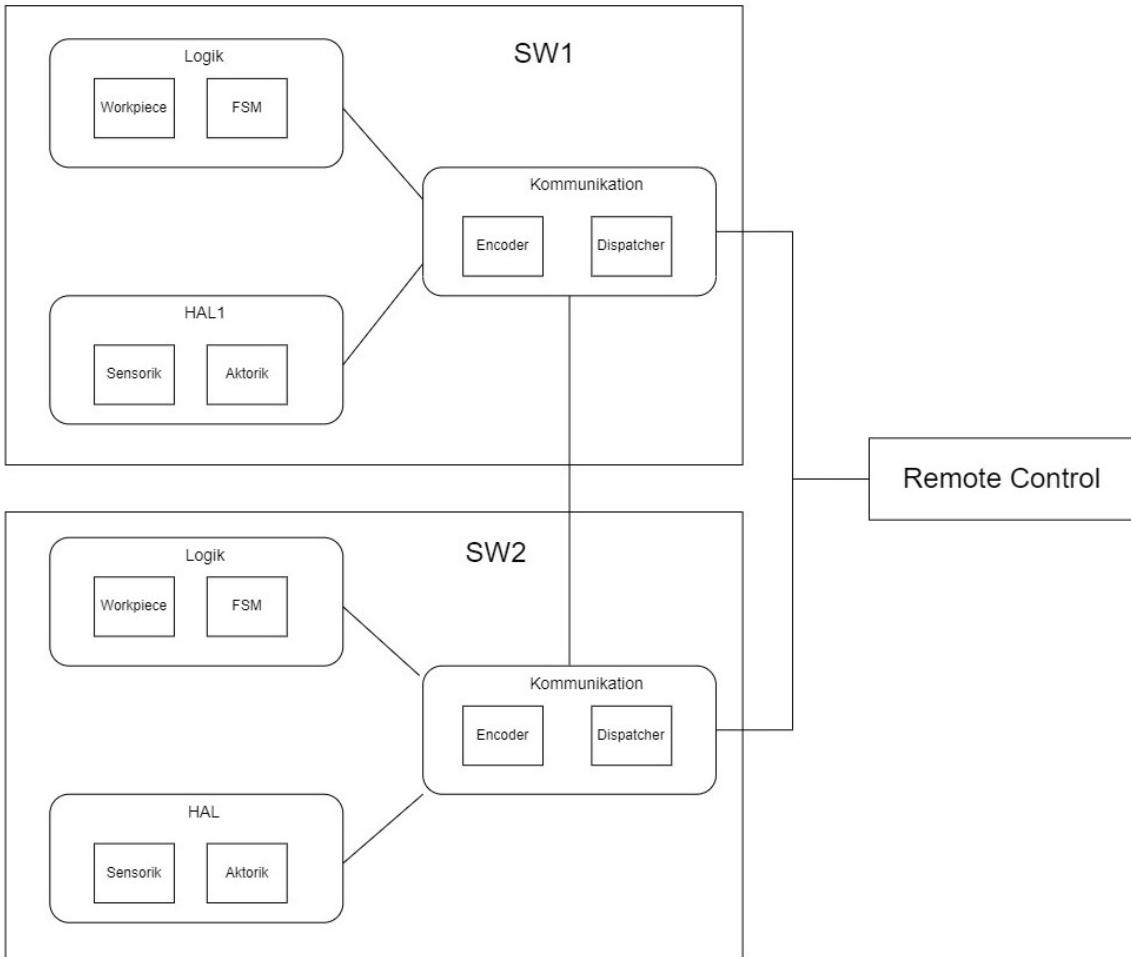
Das hat zur Folge, dass die Anlagen darauf schneller reagieren.

Wenn der E-Stop gedrückt wird, stehen alle Anlagen still, bis der E-Stop herausgezogen wird und der Reset-Taster auf allen Anlagen gedrückt wird. Der Betrieb und die Reihenfolge der Werkstücke fangen dann wieder neu an.

Grobkonzept des technischen Systementwurfes

4 Software-Design

4.1 Software Architektur



Die Softwarearchitekturen SW1 und SW2 der Anlagen 1 und 2 sind nahezu identisch und unterscheiden sich lediglich in der Funktionsweise der FSMs und der Kommunikation mit dem Remote-Controller. Beide Architekturen bestehen aus drei Hauptkomponenten: HAL, Logik und Kommunikation

- Die HAL kapselt die hardwarenahe Interaktion und beinhaltet die folgenden Klassen:

- Aktorik: Ermöglicht die Ansteuerung der Aktoren der jeweiligen Anlage (z.B. Förderband, Lampe).
- Sensorik: Dient zur Abfrage des Sensorzustands (z.B. Lichtschranke). Diese Klasse wird voraussichtlich nicht verwendet, da Zustandsänderungen der Sensoren über Interrupts signalisiert werden.

- Die Kommunikation ist für die interne und externe Kommunikation verantwortlich und umfasst:

- Encoder: Bei Empfang eines Interrupts übermittelt die ISR den aktuellen Zustand des GPIO-Registers als Pulse-Nachricht an den Encoder. Der Encoder analysiert diesen Zustand, identifiziert geänderte Pins und sendet für jeden der veränderten Pins eine separate Pulse-Nachricht an den Dispatcher.

Diese Nachrichten enthalten sowohl die Pin-Nummer als auch die Art der Zustandsänderung (high-to-low oder low-to-high).

- Dispatcher: Empfängt der Dispatcher eine Pulse-Nachricht vom Encoder, leitet er diese gezielt an die Komponenten weiter, die für die Verarbeitung solcher Nachrichten zuständig sind, wie beispielsweise Logik von SW1, Logik von SW2 oder die Remote-Control-Komponente.

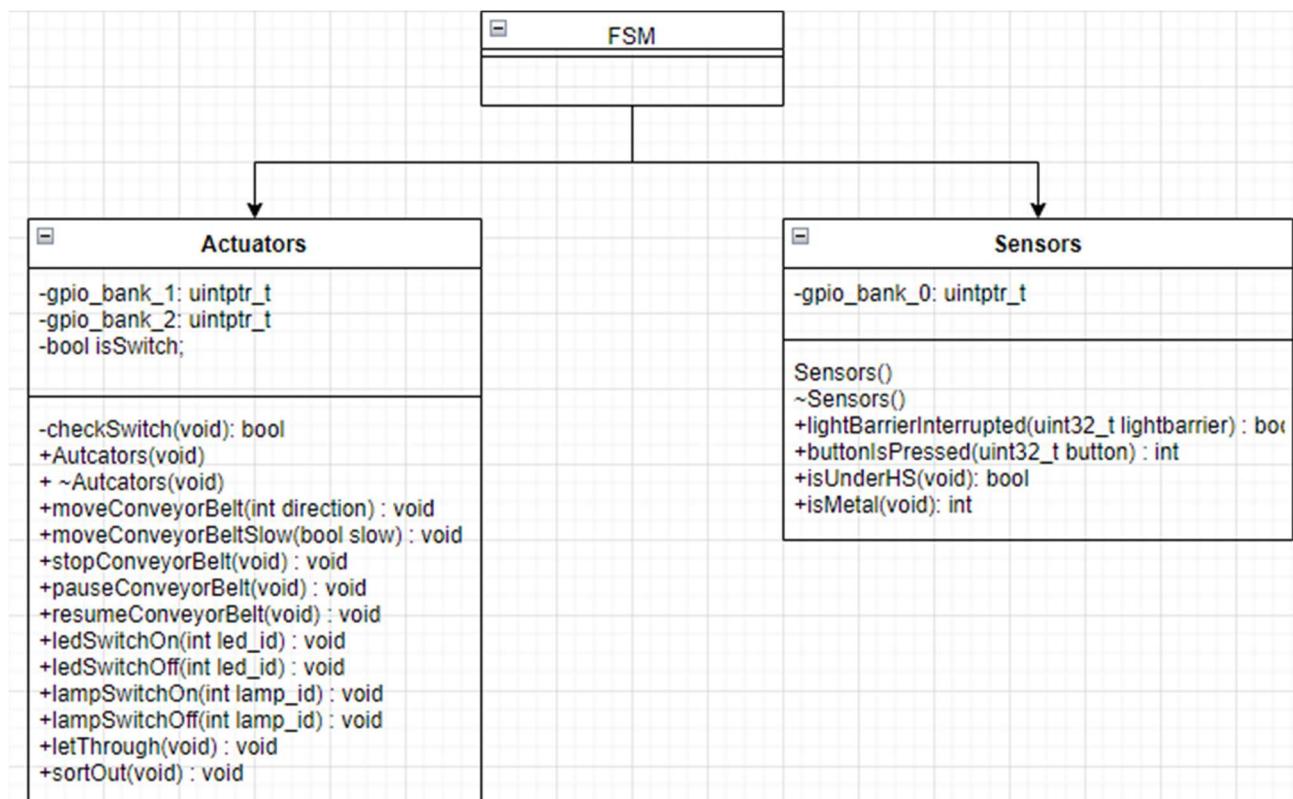
Die externe Kommunikation zwischen den beiden Anlagen erfolgt über das QNET-Protokoll nach dem Client/Server-Prinzip. Der Client (von Anlage 1 oder 2) sendet Nachrichten, wie z.B. E-Stop-Signale, an den Server (der jeweils anderen Anlage). Der Server verarbeitet diese Nachrichten, führt entsprechende Aktionen aus und informiert den Client über den Erfolg oder Misserfolg der Operation.

- Die Systemlogik ist in zwei Klassen unterteilt:

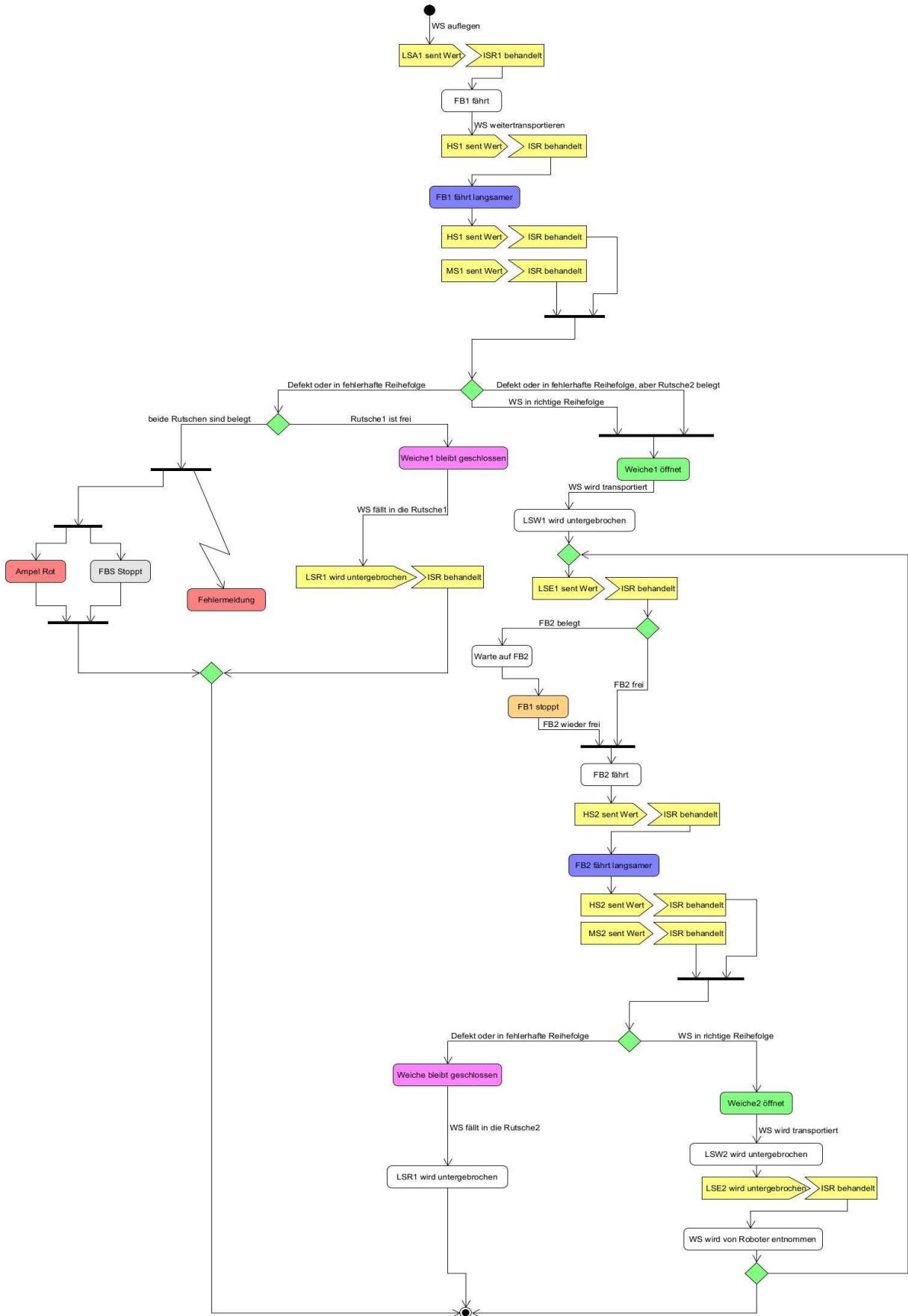
- Workpiece: Diese Klasse modelliert die Werkstücke und ihre spezifischen Eigenschaften.
- FSM (Finite State Machine): Die FSM-Klasse bildet die Zustandslogik des Systems ab. Sie definiert alle möglichen Zustände und steuert die Übergänge zwischen ihnen basierend auf Events, die vom Dispatcher empfangen werden.

- Der Remote-Controller verfügt über eine eigenständige Software, die es ihm ermöglicht, über die definierte Schnittstelle mit beiden Softwaresystemen zu kommunizieren

4.2 Software Struktur



4.3 Verhaltensmodellierung



5 Implementierung: Besonderheiten

6 Qualitätssicherung

6.1 Teststrategie

6.2 Testszenarien/Abnahmetest

T_01	Korrekte Werkstückreihenfolge auf dem Förderbandsystem überprüfen.
Requirements:	R1.1, R1.2, R1.7, R2.1, R2.2, R2.3, R2.4, R3.1, R4.2, R4.5
Vorbedingungen:	Am Anfang des FB1 wird jeweils ein WS platziert und losgelassen, bevor das nächste WS aufgelegt wird. Dabei muss zwischen den WSs ein Mindestabstand eingehalten werden. LBB1 wird unterbrochen. Die Ampeln leuchten Grün.
Testsequenz	Warte auf WS -> FB1 fährt -> FB1 langsamer fährt -> WS bei HS1 -> WS bei MS1 -> Weiche 1 öffnen / Auswerfer1 wird nicht aktiviert -> WS am Ende des FB1 -> LBB2 wird unterbrochen -> FB1 stoppt -> FB2 fährt -> FB2 langsamer fährt -> WS bei HS2 -> WS bei MS2 -> Weiche2 öffnen -> WS am Ende des FB2 -> FB2 stoppt -> WS von Roboter entfernen
Input	Die WSs sollen am Anfang von FB1 platziert werden, in folgender Reihenfolge: 1.WS mit Bohrung mit Metall 2.WS mit Bohrung mit Metall 3.Flaches WS
Output	Am Ende von FB2 werden die WSs in folgender Reihenfolge ankommen: 1.WS mit Bohrung mit Metall 2.WS mit Bohrung mit Metall 3.Flaches WS

T_02	WS nicht in der korrekten Reihenfolge platziert
Requirements:	R1.1, R1.3, R1.7, R2.1, R2.2, R2.4, R2.5, R3.1, R4.2
Vorbedingungen:	Am Anfang des FB1 wird jeweils ein WS platziert und losgelassen, bevor das nächste WS aufgelegt wird. Dabei muss zwischen den WSs ein Mindestabstand eingehalten werden. LBB1 wird unterbrochen. Die Rutsche1 hat noch freie Plätze. Die Ampeln leuchten Grün.
Testsequenz	Testsequenz für das fehlerhafte WS: Warte auf WS -> FB1 fährt -> FB1 fährt langsamer -> WS bei HS -> WS bei MS -> Weiche geschlossen bleiben / Auswerfer wird aktiviert -> WS in die Rutsche1 fallen
Input	Die WSs sollen am Anfang von FB1 platziert werden, in folgender Reihenfolge: 1.WS mit Bohrung mit Metall 2.WS mit Bohrung mit Metall 3.Beliebiges WS (aber nicht flaches WS)

	4.Flaches WS
Output	<p>Das 3. WS wird nach Prüfung durch HS1 und MS1 als falsch identifiziert. Die Weiche bleibt geschlossen/ Auswerfer wird aktiviert. WS fällt in die Rutsche 1 und wird von LB3 unterbrochen.</p> <p>Am Ende von FB2 werden die WSs in folgender Reihenfolge ankommen:</p> <ol style="list-style-type: none"> 1.WS mit Bohrung mit Metall 2.WS mit Bohrung mit Metall 3.Flaches WS

T_03	Defektes WS
Requirements:	R1.1, R1.3, R1.7, R2.1, R2.2, R2.4, R2.5, R3.1, R4.2
Vorbedingungen:	Ein defektes WS wird am Anfang von FB1 gelegt und losgelassen. LBB1 wird unterbrochen. Die Rutsche1 hat noch freie Plätze. Die Ampeln leuchten Grün.
Testsequenz	Warte auf WS -> FB1 fährt -> FB1 langsamer fährt -> WS bei HS -> WS bei MS -> Weiche geschlossen bleiben / Auswerfer wird aktiviert -> WS in die Rutsche1 fallen
Input	Defektes WS wird am Anfang vom FB1 platziert.
Output	Das defekte WS wird nach der Prüfung durch HS1 und MS1 als defekt identifiziert. Die Weiche bleibt geschlossen/ Auswerfer wird aktiviert. WS fällt in die Rutsche 1 und wird von LB3 unterbrochen.

T_04	WS nicht in der korrekten Reihenfolge platziert, wenn Rutsche 1 belegt ist.
Requirements:	R1.1, R1.3, R1.4, R1.7, R2.1, R2.2, R2.4, R2.5, R3.1, R3.2, R4.2, R4.5
Vorbedingungen:	Am Anfang des FB1 wird jeweils ein WS platziert und losgelassen, bevor das nächste aufgelegt wird. Dabei muss zwischen den WSs ein Mindestabstand eingehalten werden. LBB1 wird unterbrochen. Die Rutsche 1 hat keine freien Plätze mehr, aber in Rutsche2 sind noch freie Plätze verfügbar. Ampel1 leuchtet gelb und Ampel2 leuchtet Grün.
Testsequenz	<p>Testsequenz für das fehlerhafte WS:</p> <p>Warte auf WS -> FB1 fährt -> FB1 langsamer fährt -> WS bei HS1 -> WS bei MS1 -> Weiche1 öffnen / Auswerfer1 wird nicht aktiviert -> WS am Ende des FB1 -> LBB2 wird unterbrochen -> FB1 stoppt -> FB2 fährt -> FB2 langsamer fährt -> WS bei HS2 -> WS bei MS2 -> Weiche2 geschlossen bleiben / Auswerfer2 wird aktiviert. -> WS in die Rutsche2 fallen.</p>
Input	<p>Die WSs sollen am Anfang von FB1 platziert werden, in folgender Reihenfolge:</p> <ol style="list-style-type: none"> 1.WS mit Bohrung mit Metall 2.WS mit Bohrung mit Metall

	<p>3.Beliebiges WS (aber nicht flaches WS)</p> <p>4.Flaches WS</p>
Output	<p>Das WS wird nach Prüfung durch HS1 und MS1 und als falsch identifiziert. Die Weiche öffnet/ Auswerfer wird nicht aktiviert, da es keine freien Plätze mehr in der Rutsche1. Auf dem FB2 passieren die gleichen Schritte wie auf FB1 bisher, dass das WS in die Rutsche2 fällt und von LBC2 unterbrochen wird.</p> <p>Am Ende von FB2 werden die WSs in folgender Reihenfolge ankommen:</p> <ol style="list-style-type: none"> 1.WS mit Bohrung mit Metall 2.WS mit Bohrung mit Metall 3.Flaches WS

T_05	Ungültige WS-Position
Requirements:	R1.6, R2.2, R2.5, R3.1, R3.2, R5.3
Vorbedingungen:	Das FB1 / FB2 ist am Laufen. Ein beliebiges WS wird zwischen LBB1 und HS1 gelegt und losgelassen. LBB1 wird nicht unterbrochen. Die Rutsche1 haben noch freie Plätze. Ampeln leuchten Grün.
Testsequenz	WS auflegen -> FB1 langsamer fährt -> WS bei HS1 -> Das Förderbandsystem stoppt und die Ampeln leuchten Rot
Input	Beliebiges WS wird zwischen den LBB1 vor HS1 gelegt, sodass LBB1 nicht unterbrochen wird.
Output	Wenn FB1 fährt und das WS bei HS1 ist, es wird erkannt wird, dass es für WS keine ID existiert, da das WS nicht von der LBB1 registriert wird und das Förderbandsystem gestoppt wird.

T_06	Beide Rutschen sind voll
Requirements:	R1.1, R1.3, R2.1, R2.2, R2.3, R2.5, R3.1, R4.2, R4.5, R5.1
Vorbedingungen:	Ein fehlerhaftes WS wird am Anfang von FB1 gelegt und losgelassen. LBB1 wird unterbrochen. Die beiden Rutschen sind voll (beide Rutschen enthalten 4 WSs). Die Ampeln leuchten Gelb.
Testsequenz	Warte auf WS -> FB1 fährt -> FB1 langsamer fährt -> WS bei HS1 -> WS bei MS1 -> FB1 und FB2 werden gestoppt / Ampeln Rot
Input	<ol style="list-style-type: none"> a. Das WS, das mit Bohrung mit Metall oder flaches Werkstück in seiner Fehlerhaften Reihenfolge ist, am Anfang des FB1 auflegen. b. Das defekte WS am Anfang des FB1 auflegen.
Output	Das WS wird nach Prüfung durch HS1 und MS1 und als falsch identifiziert. Deshalb werden die FB1 und FB2 gestoppt und die Ampeln leuchten Rot.

6.3 Testprotokolle und Auswertungen

7 Technische Schulden

8 Lessons Learned

9 Anhang

9.1 Glossar

9.2 Abkürzungen

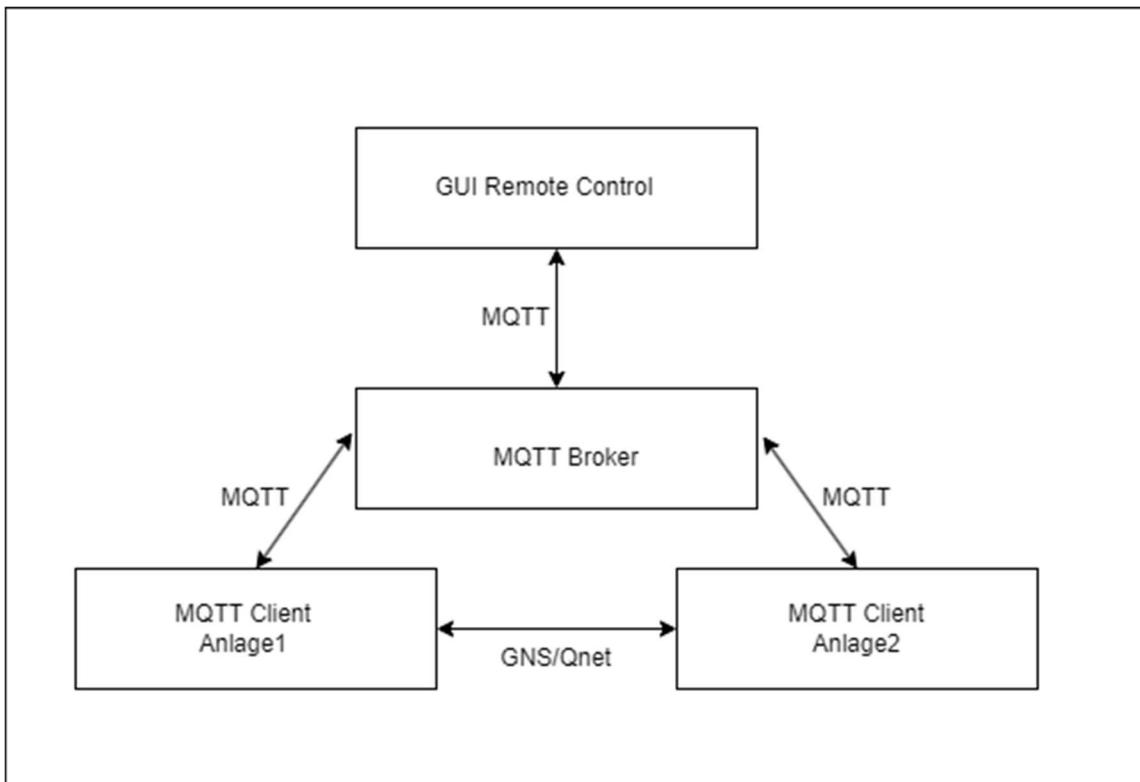
- WS/WSS -> Werkstück/Werkstücke
- HS:
 - HS1 -> Höhensor der ersten Anlage
 - HS2 -> Höhensor der zweiten Anlage
- MS:
 - MS1 -> Metallsensor der ersten Anlage
 - MS2 -> Metallsensor der zweiten Anlage
- FB:
 - FB1 -> Förderband der ersten Anlage
 - FB2 -> Förderband von der zweiten Anlage
- LB:
 - LB -> Lichtschranke (Lightbarrier)
 - LBB: (Beginn)
 - LBB1 -> LB am Anfang der ersten Anlage
 - LBB2 -> LB am Anfang der zweiten Anlage
 - LBH:
 - LBH1 -> LB beim HS der ersten Anlage
 - LBH2 -> LB beim HS der zweiten Anlage
 - LBS: (Switch)
 - LBS1 -> LB bei der Weiche der ersten Anlage
 - LBS2 -> LB bei der Weiche der zweiten Anlage
 - LBC: (Chute)
 - LBC1 -> LB bei der Rutsche der ersten Anlage
 - LBC2 -> LB bei der Rutsche der zweiten Anlage
 - LBE: (End)
 - LBE1 -> LB am Ende der ersten Anlage
 - LBE2 -> LB am Ende der zweiten Anlage

9.3 Beantwortete Fragen

- Was ist damit gemeint? „Das gleiche Namensschema mit dem Postfix „_RDD“ gilt für den Dateinamen von Ihrem RDD.“ (S. 8)
 - Am Ende jeder Datei den Präfix „_1_2“ schreiben (für Praktikumsgruppe 1 Team 2)
- Es gibt keine User Stories -> wie Stakeholder?
 - Profs, Studenten, HAW
- Lampe: was sind Warnungen?

- kein Eingriff vom Bediener notwendig, z.B.: volle Rutsche führt zu einer Warnung
- was ist Service Mode?
 - Selbsttest, Kalibrierungsmodus
- Was passiert nach dem die korrekte Reihenfolge einmal durchgelassen wurde?
 - Die Anlagen arbeiten weiter und die Reihenfolge wiederholt sich
- Wenn beide Rutschen voll sind und ein defektes WS erkannt wurde, soll der Bandstopp dann erfolgen, wenn das WS auf FB1 oder FB2 ist?
 - Auf FB2, da in der Zwischenzeit ein WS von Rutsche2 entfernt werden könnte und somit wieder Platz für das Aussortieren des defekten WS wäre.
- Was passiert, wenn die Anlagen fehlerfrei laufen und die Reset-Taste gedrückt wurde?
 - Nichts -> Reset-Taste ist nur bei Fehlerzuständen notwendig um einen Fehler zu quittieren.
- Was passiert, wenn die Anlagen wieder in den Betriebszustand wechseln, nachdem die Stop-Taste gedrückt wurde? Sollen sie weiter machen, wo sie aufgehört haben oder von neu anfangen?
 - Stop entspricht semantisch einer Pause -> nur an der jeweiligen Anlage wird pausiert, solange nichts Katastrophales (E-Stop) passiert -> Wenn danach wieder Start gedrückt wird, dann läuft die Anlage weiter.
 - E-Stop -> Zustände werden zurückgesetzt, WS werden entnommen. Reihenfolge wird neu anfangen -> Als hätte man die Anlagen komplett neu gestartet
- Was ist der Ruhezustand?
 - Anlagen sind an, aber Start-Taste wurde noch nicht gedrückt (Punkt Stop-Taste ist zu berücksichtigen).

10 Remote Control:



Ziel:

Die Anlagen sollen von einem Remote-System gesteuert und überwacht werden. Die Kommunikation soll auf MQTT basieren und die Remote-Control-Anwendung soll eine graphische Oberfläche haben. Diese Oberfläche soll als Web-Anwendung realisiert werden.

Anforderungen :

1. MQTT-Kommunikation:
Die Kommunikation zwischen den Anlagen und dem Remote-System erfolgt über MQTT.
2. Graphische Benutzeroberfläche:
Die Remote-Control-Anwendung soll eine graphische Oberfläche haben als Web-Anwendung implementiert wird.
3. Button-Panel :
Die Button-Panel der Anlagen sollen in der Oberfläche sichtbar und benutzbar sein.
4. Anzeige relevanter Informationen:
Relevante Informationen z.B., dass eine Rutsche voll ist, sollen graphisch dargestellt werden.
5. Konsolenausgaben:
Die Ausgaben auf der Konsole der Anlagen soll auch in der Remote-Control-Anwendung ausgegeben werden.

Schritte zur Implementierung :

1. Einrichtung des MQTT-Broker.
2. Hardware Anforderungen:
Raspberry Pi
Internetzugang für den Raspberry Pi (WLAN)
Verbindung zur Anlage (Kabel)
3. Entwicklung der Web-basierten Remote-Control-Anwendung mit Node-Red.