

# Requirements and Design Documentation (RDD)

Version 1.00

## SE2P – Praktikum – WS2012

Jan-Tristan Rudat, 2007852, [jan-tristan.rudat@haw-hamburg.de](mailto:jan-tristan.rudat@haw-hamburg.de)

Martin Slowikowski, 1999166, [martin.slowikowski@haw-hamburg.de](mailto:martin.slowikowski@haw-hamburg.de)

Chris Addo, 2010200, [christopher.addo@haw-hamburg.de](mailto:christopher.addo@haw-hamburg.de)

Jens Eberwein, 2007797, [jens.eberwein@haw-hamburg.de](mailto:jens.eberwein@haw-hamburg.de)

### Changelog:

Version	Author	Datum	Anmerkungen
0.01	Rudat	14.10.2012	RDD erstellt + Requirements
0.02	Slowikowski	16.10.2012	UML
1.00	Slowikowski	17.10.2012	Aufgabenplan, Milestone 1
1.xx	Xxx	Xxx	Xxx
2.xx	Xx		MS2
3.01	Slowikowski	07.11.2012	Regressionstests

# Inhaltsverzeichnis

1 Motivation .....	4
2 Randbedingungen .....	4
2.1 Entwicklungsumgebung .....	4
2.2 Werkzeuge .....	4
2.3 Sprachen .....	4
3 Requirements und Use Cases (Sequenzdiagramme).....	4
3.1 Allgemeine Anforderungen .....	4
3.2 Anforderungen .....	5
3.2.1 Durchlauf akzeptierter Werkstücke .....	5
3.2.2 Aussortieren von zu flachen Werkstücken.....	6
3.2.3 Aussortieren von Werkstücken mit Bohrung nach oben mit Metalleinsatz .....	6
3.3 Fehlerszenarien .....	6
3.3.1 Fehlermeldung „Rutsche voll“ .....	6
3.3.2 Werkstück wurde vom Band genommen .....	7
3.3.3 Werkstück wurde mitten auf dem Band hinzugefügt .....	7
3.4 Diagramme .....	8
3.4.1 Durchlauf akzeptierter Werkstücke .....	8
3.4.2 Aussortieren von zu flachen Werkstücken.....	9
3.4.3 Aussortieren von Werkstücken mit Bohrung nach oben mit Metalleinsatz .....	10
3.4.4 Fehlerbehandlung .....	11
4 Design .....	12
4.1 System .....	12
4.2 Datenmodell .....	12
4.3 Verhaltensmodell .....	12
5 Implementierung.....	12
5.1 Algorithmen.....	12
5.2 Patterns .....	12
5.3 Mapping Rules .....	12
6 Testen .....	12
6.1 Unit-Test/Komponenten Test.....	12
6.2 Integration Test/System Test .....	12
6.3 Regressionstest.....	12
6.3.1 Aktorik HAL Testablauf .....	12
6.3.2 RS232 Testablauf .....	13

6.3.3 Ampelkontrollthread Testablauf .....	13
6.3.4 Sensorik HAL Testablauf .....	14
6.4 Abnahmetest .....	17
6.5 Testplan .....	17
6.6 Testprotokolle und Auswertungen.....	17
7 Projektplan .....	17
7.1 Verantwortlichkeiten.....	17
7.2 Projektstrukturplan .....	17
8 Lessons Learned .....	17
Glossar .....	17
Abkürzungen .....	17
Anhänge.....	17

# 1 Motivation

Im Rahmen des Studienganges "Technische Informatik" an der HAW Hamburg, soll im Rahmen des vierten Semesters ein Kurs namens Software Engineering 2 absolviert werden.

In diesem Kurs werden vertiefende Grundlagen des Software Engineering vermittelt, sowie eine Aufgabe erteilt, in welcher die Software für eine Werkstücksortieranlage entwickelt werden soll.

Die Werkstücksortieranlage besteht aus zwei Förderbandmodulen, welche über zwei GEME Rechner gesteuert werden. Beide Rechner sind über eine serielle Schnittstelle miteinander verbunden.

Aus diesem Kontext ist das vorliegende Dokument entstanden.

## 2 Randbedingungen

### 2.1 Entwicklungsumgebung

- Visual Paradigm 10.0 Enterprise
- Momentics 4.70 DIE
- QNX 6.5

### 2.2 Werkzeuge

- GIT
- TortoiseGIT
- Notepad++

### 2.3 Sprachen

- C
- C++
- Shellscript

## 3 Requirements und Use Cases (Sequenzdiagramme)

### 3.1 Allgemeine Anforderungen

Werkstücke werden in gewissen Zeitabständen aufs Band gelegt und Sensoren sortieren bestimmte Werkstücke aus.

Diese Werkstücke können auf das Band gelegt werden:

- mit richtiger Höhe
  - o Mit Metalleinsatz
    - Öffnung nach oben
    - Öffnung nach Unten
  - o Ohne Metalleinsatz
    - Öffnung nach oben
    - Öffnung nach Unten
- mit falscher Höhe

Folgende sollen aussortiert werden:

Höhe	Metall	Öffnung	Aussortieren
richtige Höhe	Mit Metalleinsatz	Öffnung nach oben	Ja (Band 2)
		Öffnung nach Unten	Ja (Band 2, wenden Band 1)
	Ohne Metalleinsatz	Öffnung nach oben	Nein
		Öffnung nach Unten	Nein (wenden Band 1)

mit falsche Höhe			Ja (Band 1)
------------------	--	--	-------------

## 3.2 Anforderungen

### 3.2.1 Durchlauf akzeptierter Werkstücke

**Akteur:** Arbeiter am Förderband

**Ziel:** Akzeptiertes Werkstück erreicht Ende des zweiten Förderbandes

**Auslöser:** Arbeiter legt ein Werkstück an den Anfang des ersten Förderbandes

**Vorbedingung:**

- Förderband 1 in Betrieb und bereit (Ampel: Grün)
- Anfang des ersten Förderbandes ist frei (Schranke 1 nicht unterbrochen)

**Erfolgsszenario 1:**

1. Ermittlung der Höhe des Werkstückes mit Höhenmessung
2. Werkstück hat akzeptierte Höhe und Bohrung zeigt nach **oben**, Werkstück wird angenommen
3. Öffnen der Weiche, Werkstück wird durchgelassen
4. Weiche wird geschlossen
5. Werkstück erreicht Ende des ersten Förderbandes
6. Transport des Werkstück auf Förderband 2, da dieses frei ist
7. Bohrung des Werkstücks zeigt nach oben
8. Werkstück enthält kein Metallkern
9. Weiche wird geöffnet, Werkstück wird durchgelassen
10. Weiche wird wieder geschlossen
11. Werkstück erreicht das Ende von Band 2, Band zwei bleibt stehen
12. Arbeiter nimmt das Werkstück vom Band 2

**Erfolgsszenario 2:**

1. Ermittlung der Höhe des Werkstückes mit Höhenmessung
2. Werkstück hat akzeptierte Höhe und Bohrung zeigt nach **unten**, Werkstück wird angenommen
3. Öffnen der Weiche, Werkstück wird durchgelassen
4. Weiche wird geschlossen
5. Werkstück erreicht Ende des ersten Förderbandes (Zustandsanzeige: Gelb blinkend)
6. Arbeiter nimmt Werkstück aus der Lichtschranke, wendet es und legt es zurück
7. Transport des Werkstück auf Förderband 2, da dieses frei ist
8. Bohrung des Werkstücks zeigt nach oben
9. Werkstück enthält kein Metallkern
10. Weiche wird geöffnet, Werkstück wird durchgelassen
11. Weiche wird wieder geschlossen
12. Werkstück erreicht das Ende von Band 2, Band zwei bleibt stehen
13. Arbeiter nimmt das Werkstück vom Band 2

**Nachbedingung:** Werkstück wird nach Erreichen der Lichtschranke am Ende von Band 2 entnommen

**Fehlerfälle:** siehe Fehlerszenarien:

- Werkstück wurde vom Band genommen

- Werkstück wurde mitten auf dem Band hinzugefügt

### 3.2.2 Aussortieren von zu flachen Werkstücken

**Akteur:** -

**Ziel:** Werkstücke, deren Höhe kleiner XXmm beträgt, werden von Band 1 aussortiert

**Auslöser:** Sensor meldet die Höhe des Werkstücks

**Vorbedingung:**

- Bandanlage in Betrieb (Zustandsanzeige: Grün)
- Werkstücke befinden sich auf dem Förderband

**Erfolgsszenario:**

1. Höhe des Werkstückes wird mit Hilfe der Höhenmessung erkannt
2. Werkstück zu flach
3. Weiche bleibt zu
4. Werkstück wird aussortiert

**Nachbedingung:** Aussortiertes Werkstück befindet sich auf der Rutsche

**Fehlerfälle:** siehe Fehlerszenarien:

- Rutsche ist voll

### 3.2.3 Aussortieren von Werkstücken mit Bohrung nach oben mit Metalleinsatz

**Akteur:** -

**Ziel:** Werkstücke mit Metalleinsatz, deren Bohrung nach oben liegt, werden von Band 2 aussortiert

**Auslöser:** Sensor meldet Metall im Werkstück

**Vorbedingung:**

- Bandanlage in Betrieb (Zustandsanzeige: Grün)
- Ein Werkstück befindet sich auf dem Förderband

**Erfolgsszenario:**

1. Metallsensor auf Band 2 erkennt Metallkern im Werkstück
2. Weiche bleibt geschlossen
3. Werkstück wird aussortiert

**Nachbedingung:** Aussortiertes Werkstück befindet sich auf der Rutsche

**Fehlerfälle:** siehe Fehlerszenarien:

- Rutsche ist voll

## 3.3 Fehlerszenarien

### 3.3.1 Fehlermeldung „Rutsche voll“

**Akteur:** Arbeiter am Förderband

**Ziel:** Behebung des Fehlers: Rutsche entleeren

**Auslöser:** Sensor meldet Rutsche voll

**Vorbedingung:**

- Bandanlage in Betrieb (Zustandsanzeige: Grün)
- Ein Werkstück wurde auf die Rutsche geschoben

**Ablauf der Fehlerbehebung:**

1. Bandstopp, Zustandsanzeige blinkt rot (Fehlerzustand: „anstehend unquittiert“, schnelles Blinken 1 Hz)
2. Arbeiter sieht den Fehler
3. Arbeiter drückt die Quittierungstaste
4. Zustandsanzeige: rotes Dauerlicht (Fehlerzustand: „anstehend quittiert“)
5. Arbeiter nimmt Werkstücke von der Rutsche
6. Arbeiter betätigt die Starttaste
7. Rote Leuchte erlischt

**Nachbedingung:** Bandanlage wieder in Betrieb (Zustandsanzeige: Grün)

### 3.3.2 Werkstück wurde vom Band genommen

**Akteur:** Arbeiter am Förderband

**Ziel:** Das entnommene Werkstück wird an den Anfang von Band eins gelegt

**Auslöser:** Sensor meldet, dass ein Werkstück fehlt

**Vorbedingung:**

- Bandanlage in Betrieb (Zustandsanzeige: Grün)
- Ein Werkstück wird vom Band genommen

**Ablauf der Fehlerbehebung:**

1. Bandstopp, Zustandsanzeige blinkt rot (Fehlerzustand: „anstehend unquittiert“, schnelles Blinken 1Hz)
2. Arbeiter sieht den Fehler
3. Arbeiter drückt die Quittierungstaste
4. Zustandsanzeige: rotes Dauerlicht (Fehlerzustand: „anstehend quittiert“)
5. Arbeiter legt das vom Band genommene Werkstück an den Anfang von Band 1
6. Arbeiter betätigt die Starttaste
7. Rote Leuchte erlischt

**Nachbedingung:** Bandanlage wieder in Betrieb (Zustandsanzeige: Grün)

### 3.3.3 Werkstück wurde mitten auf dem Band hinzugefügt

**Akteur:** Arbeiter am Förderband

**Ziel:** Werkstück wird wieder vom Band genommen

**Auslöser:** Sensor meldet, dass ein Werkstück zu viel auf dem Band ist

**Vorbedingung:**

- Bandanlage in Betrieb (Zustandsanzeige: Grün)
- Ein Werkstück wird mitten auf dem Band hinzugefügt

### Ablauf der Fehlerbehebung:

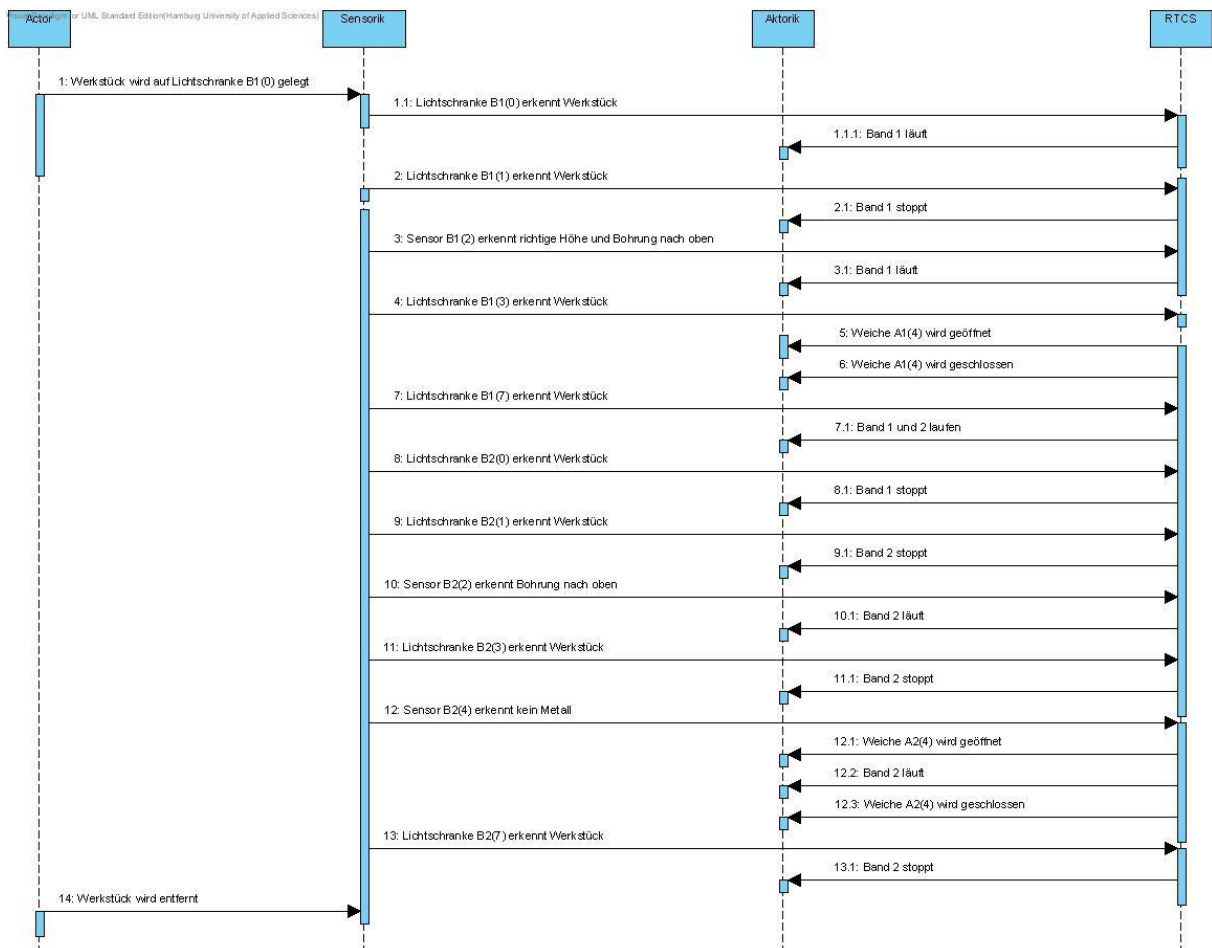
1. Bandstopp, Zustandsanzeige blinkt rot (Fehlerzustand: „anstehend unquittiert“, schnelles Blinken 1Hz)
2. Arbeiter sieht den Fehler
3. Arbeiter drückt die Quittierungstaste
4. Zustandsanzeige: rotes Dauerlicht (Fehlerzustand: „anstehend quittiert“)
5. Arbeiter entfernt das hinzugefügte Werkstück wieder vom Band
6. Arbeiter betätigt die Starttaste
7. Rote Leuchte erlischt

**Nachbedingung:** Bandanlage wieder in Betrieb (Zustandsanzeige: Grün)

## 3.4 Diagramme

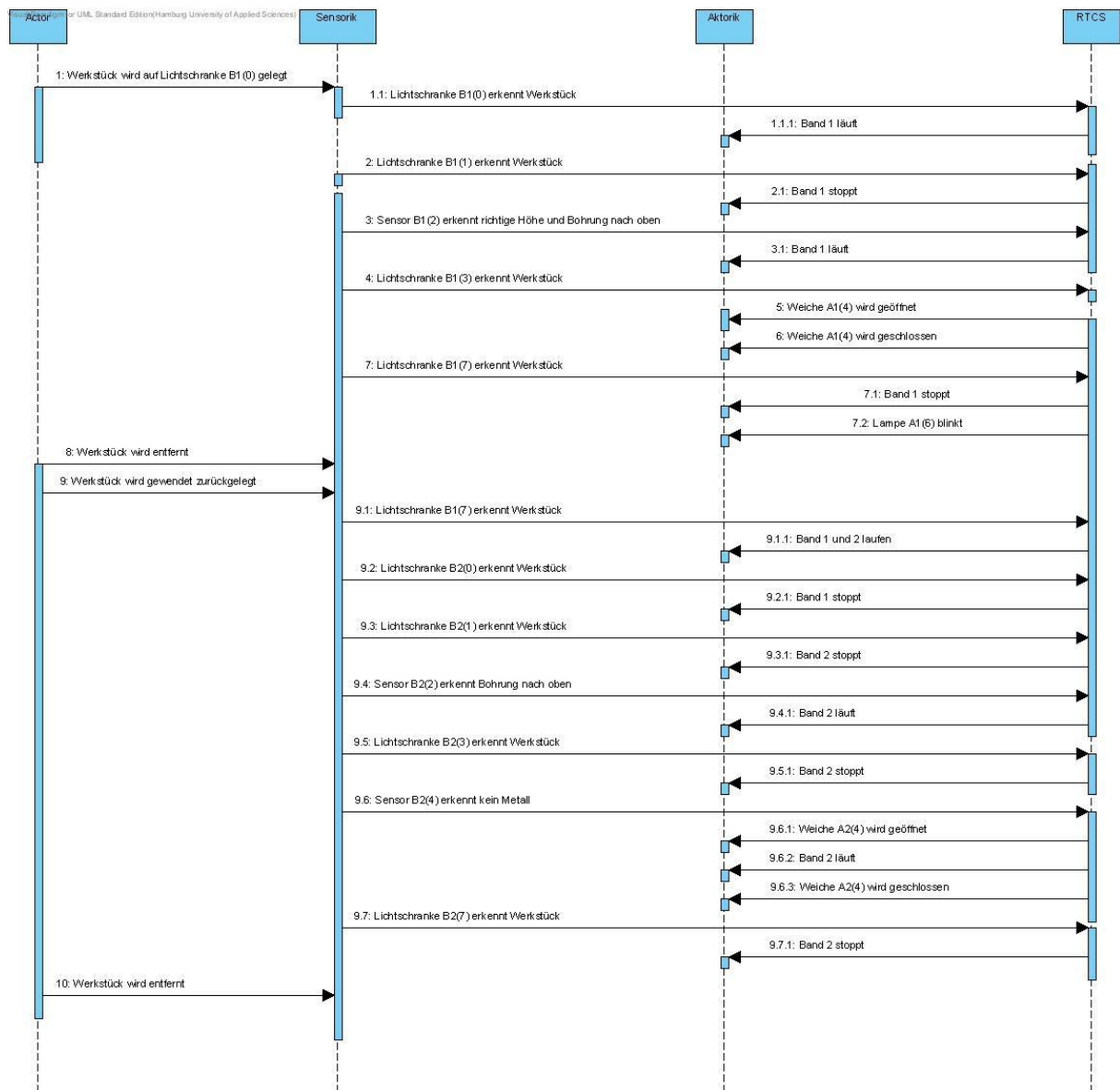
### 3.4.1 Durchlauf akzeptierter Werkstücke

**Szenario 1, richtige Höhe, Bohrung nach oben, kein Metall:**

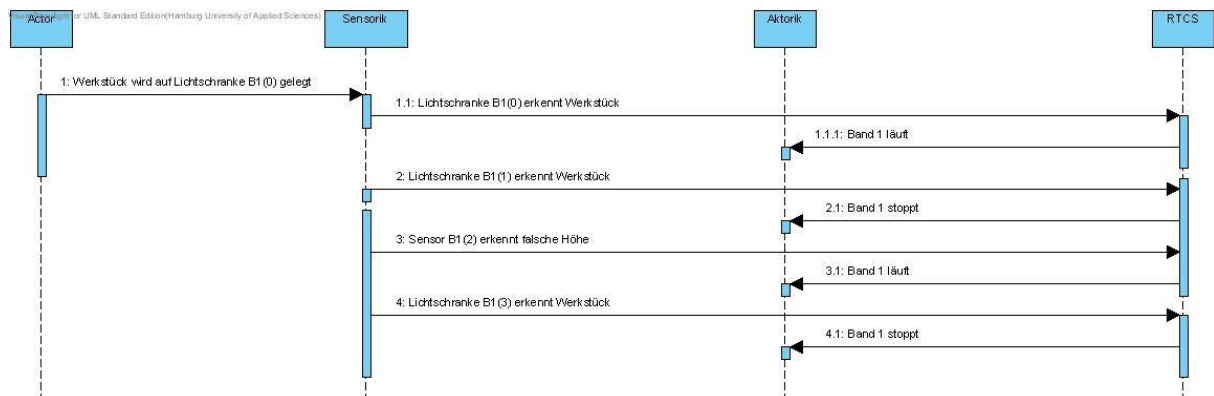




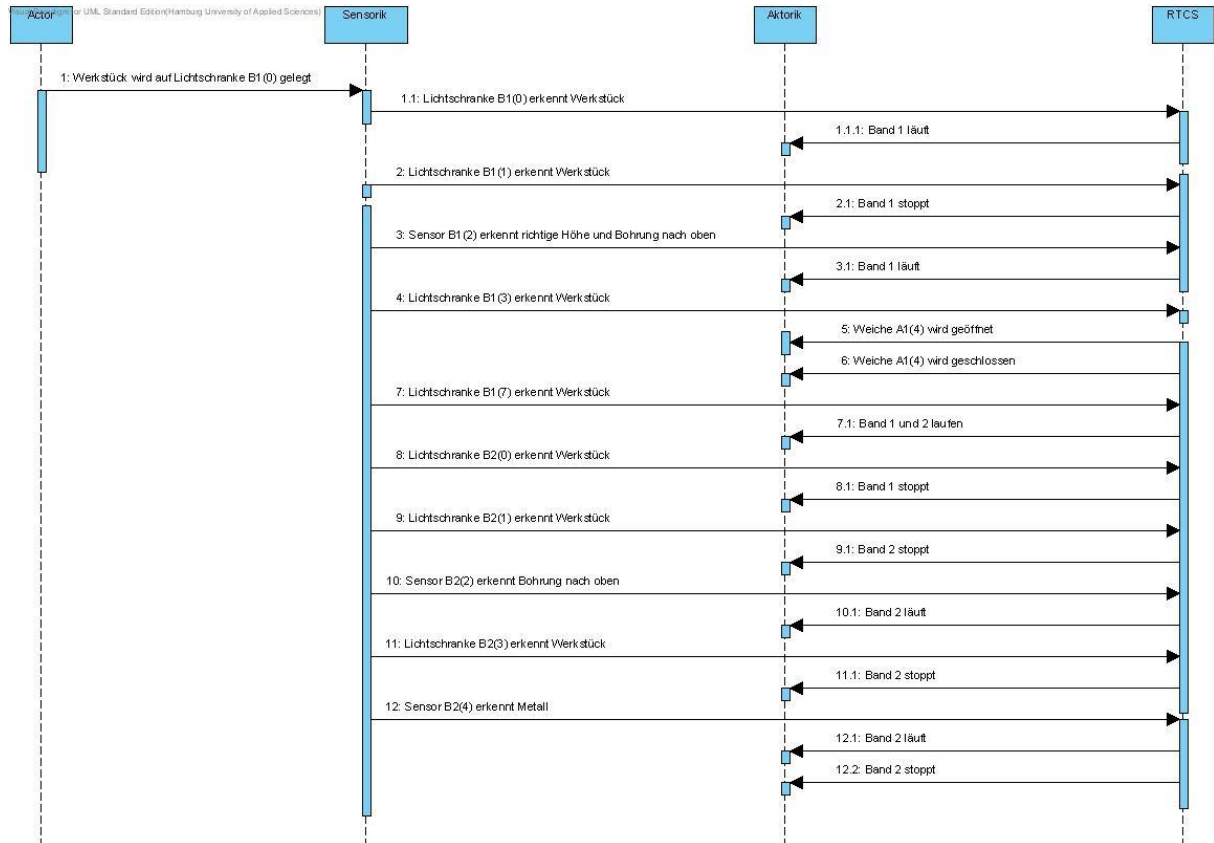
## Szenario 2, richtige Höhe, Bohrung nach unten, kein Metall:



### 3.4.2 Aussortieren von zu flachen Werkstücken

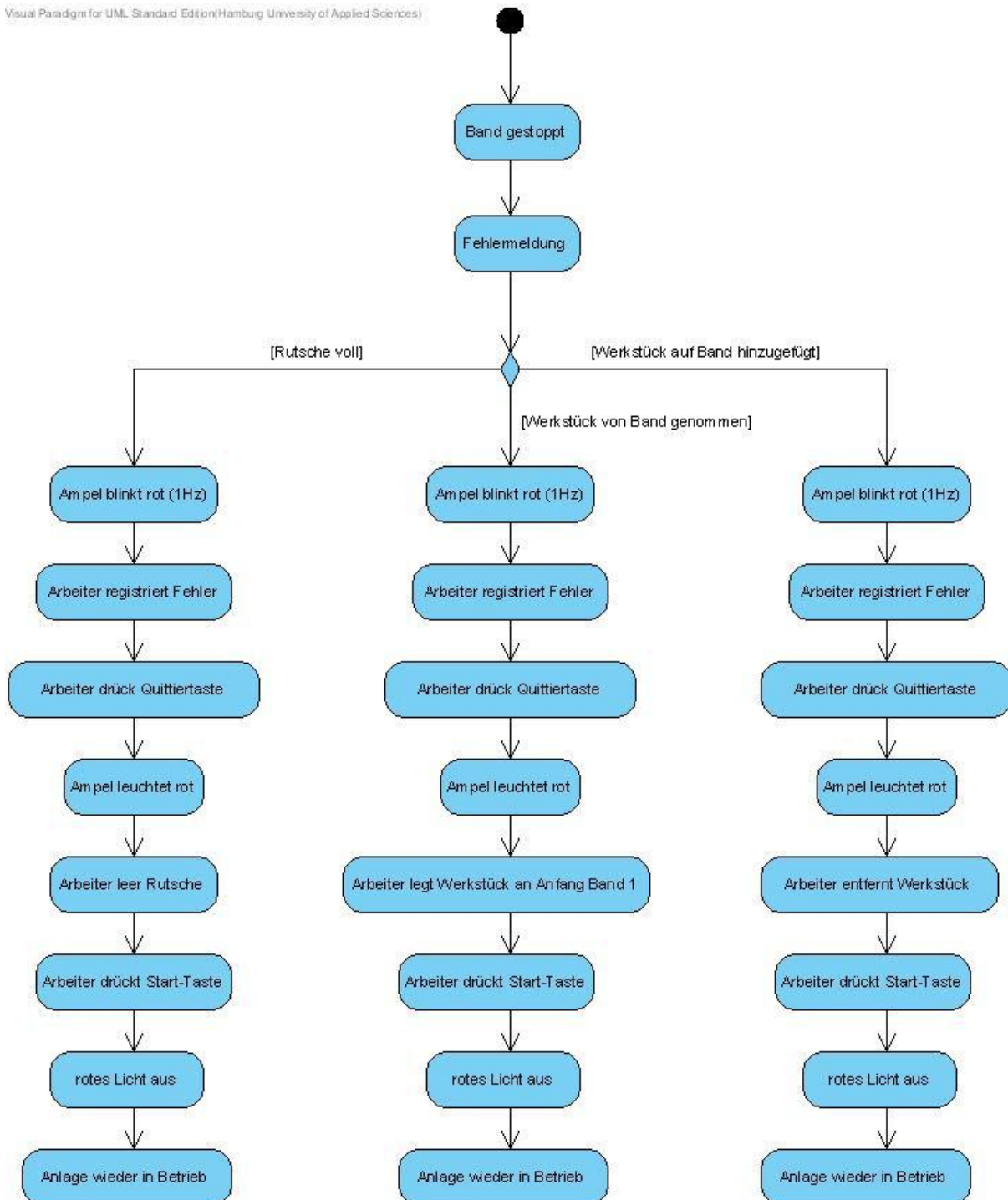


### 3.4.3 Aussortieren von Werkstücken mit Bohrung nach oben mit Metalleinsatz



### 3.4.4 Fehlerbehandlung

Visual Paradigm for UML Standard Edition (Hamburg University of Applied Sciences)



## 4 Design

### 4.1 System

### 4.2 Datenmodell

### 4.3 Verhaltensmodell

## 5 Implementierung

### 5.1 Algorithmen

### 5.2 Patterns

### 5.3 Mapping Rules

## 6 Testen

### 6.1 Unit-Test/Komponenten Test

### 6.2 Integration Test/System Test

### 6.3 Regressionstest

Um Fehler in Modifikationen bereits getesteter Software innerhalb dieses Projektes zu finden, ist es unerlässlich, über einen Grundstock an Regressionstests zu verfügen. Es werden folgende Testfälle ausgeführt:

#### 6.3.1 Aktorik HAL Testablauf

Für einen Test der gesamten Aktorik werden folgende Zeilen Code verwendet:

```
39 HALTest halTest;  
40 halTest.testHal();
```

Folgende Ausgabe auf der Konsole wird erwartet, analog zur Ausgabe soll das Verhalten am Festo-System zu beobachten sein:

Debug Hal: New HAL instance created Debug Hal: red light on Debug Hal: red light off Debug Hal: yellow light on Debug Hal: yellow light off Debug Hal: green light on Debug Hal: green light off Debug Hal: green light on Debug Hal: yellow light on Debug Hal: red light on Debug Hal: all lights off Debug Hal: engine right with normal speed Debug Hal: engine right with slow speed Debug Hal: engine stopped Debug Hal: engine revert stop Debug Hal: engine stopped Debug Hal: gate open	rotes Licht an Ampel geht an rotes Licht an Ampel geht aus gelbes Licht an Ampel geht an gelbes Licht an Ampel geht aus grünes Licht an Ampel geht an grünes Licht an Ampel geht aus grünes Licht an Ampel geht an gelbes Licht an Ampel geht an rotes Licht an Ampel geht an alle Lichter an Ampel gehen aus Laufband fährt rechts Laufband fährt langsam rechts Laufband hält an Laufband fährt weiter rechts Laufband hält an Weiche öffnet sich
--	--

Debug Hal: gate closed	Weiche schließt sich
Debug Hal: start led on	LED Start geht an
Debug Hal: reset led on	LED Reset geht an
Debug Hal: Q1 led on	LED Q1 geht an
Debug Hal: Q2 led on	LED Q2 geht an
Debug Hal: start led off	LED Start geht aus
Debug Hal: reset led off	LED Reset geht aus
Debug Hal: Q1 led off	LED Q1 geht aus
Debug Hal: Q2 led off	LED Q2 geht aus

### 6.3.2 RS232 Testablauf

Für einen Test der seriellen Schnittstelle werden folgende Zeilen Code verwendet, weiterhin müssen die beiden COM-Ports des GEME-PC mit einem Null-Modem Kabel verbunden werden:

```
42 RS232Test rs232Test;
43 rs232Test.testRS232();
```

Folgende Ausgabe ist bei korrekter Funktion auf der Konsole zu erwarten:

Debug RS232_1: opening devfile1 SUCCEEDED	COM1 initialisiert und geöffnet
Debug RS232_1: New RS232_1 instance created	
Debug RS232_2: opening devfile2 SUCCEEDED	COM2 initialisiert und geöffnet
Debug RS232_2: New RS232_2 instance created	
Debug RS232_1: Unknown msg recved: b	Lese auf COM1, schreibe auf COM2
Debug RS232_1: Timeout or EAGAIN	Unbekannte Nachricht ,b' empfangen
Debug RS232_1: Timeout recved	
Debug RS232_1: Timeout or EAGAIN	Zyklisch generierter Timeout
Debug RS232_1: Timeout recved	
Testmessage recved on devfile1: a	Gültige Nachricht ,a' empfangen
Debug RS232_1: Timeout or EAGAIN	
Debug RS232_1: Timeout recved	
Debug RS232_1: Timeout or EAGAIN	
Debug RS232_1: Timeout recved	
Debug RS232_2: Unknown msg recved: b	Lese auf COM2, schreibe auf COM1
Debug RS232_2: Timeout or EAGAIN	Unbekannte Nachricht ,b' empfangen
Debug RS232_2: Timeout recved	
Debug RS232_2: Timeout or EAGAIN	Zyklisch generierter Timeout
Debug RS232_2: Timeout recved	
Testmessage recved on devfile2: a	Gültige Nachricht ,a' empfangen
Debug RS232_2: Timeout or EAGAIN	
Debug RS232_2: Timeout recved	
Debug RS232_2: Timeout or EAGAIN	
Debug RS232_2: Timeout recved	

### 6.3.3 Ampelkontrollthread Testablauf

Für einen Test der verschiedenen Funktionen der Ampel werden folgende Zeilen Code benötigt:

```
46 LightControllerTest lctest;
47 lctest.testLightController();
```

Bei korrekter Funktion ist folgender Ablauf auf der Konsole zu beobachten, analog dazu sind die beschriebenen einzelnen Zustände auch an der Ampel zu sehen:

Debug Hal: New HAL instance created	
Debug Hal: all lights off	Falls noch Lampen an sind, aus machen
Debug Hal: green light on	Normaler Betrieb
Debug Hal: all lights off	Alle Lichter löschen
Debug Hal: red light on	Anstehend unquittiert
Debug Hal: red light off	
Debug Hal: red light on	
Debug Hal: red light off	
Debug Hal: red light on	
Debug Hal: red light off	
Debug Hal: red light on	
Debug Hal: red light off	
Debug Hal: all lights off	Alle Lichter löschen
Debug Hal: red light on	Anstehend quittiert
Debug Hal: all lights off	Alle Lichter löschen
Debug Hal: red light on	Gegangen unquittiert
Debug Hal: red light off	
Debug Hal: red light on	
Debug Hal: red light off	
Debug Hal: all lights off	Alle Lichter löschen
Debug Hal: yellow light on	Manuelle Drehung
Debug Hal: yellow light off	
Debug Hal: yellow light on	
Debug Hal: yellow light off	
Debug Hal: all lights off	Alle Lichter löschen

### 6.3.4 Sensorik HAL Testablauf

Für einen Test der Sensorik mit Interrupts müssen folgende Zeilen Code verwendet werden:

```
54     ISRTTest isrtest;
55     isrtest.start(0);
```

Beim folgenden Testablauf wurden der Reihe nach folgende Werkstücke auf das Band gelegt:

1. Zu kleines Werkstück (landet auf der Rutsche)
2. Akzeptiertes Werkstück, Bohrung nach oben (Band stoppt, wenn WS das Ende erreicht)
3. Akzeptiertes Werkstück, Bohrung nach unten (Band stoppt, wenn WS das Ende erreicht, Ampel blinkt gelb)
4. Werkstück mit Metall (WS landet auf der Rutsche)

Diese Reihenfolge führt zu folgender Ausgabe auf der Konsole:

Debug SensorHAL: New SensorHAL instance created	
Debug Hal: New HAL instance created	
ISR status: aca	Werkstück vorne auf Band gelegt
Werkstueck im Einlauf	
Debug Hal: engine right with normal speed	Band fährt los
Debug Hal: engine revert stop	Lief Band bereits, stopp Bit löschen
ISR status: acb	Werkstück hat erste Schranke verlassen
Kein Werkstueck im Einlauf	
ISR status: ac9	Werkstück in Höhenmessung
Werkstueck in Hoehenmessung	
Werkstueck Hoehe: 2701	Messung der Höhe
Debug Hal: engine stopped	
Debug Hal: engine revert stop	

ISR status: acb Kein Werkstueck in Hoehenmessung ISR status: ac3 Werkstueck in Weiche ISR status: acb Kein Werkstueck in Weiche ISR status: a8b Rutsche voll ISR status: acb Rutsche nicht voll Debug Hal: engine stopped ISR status: aca Werkstueck im Einlauf Debug Hal: engine right with normal speed Debug Hal: engine revert stop ISR status: acb Kein Werkstueck im Einlauf ISR status: acf Werkstueck im Toleranzbereich: 3798 ISR status: acb Werkstueck zu klein/gross: 2708 ISR status: ac9 Werkstueck in Hoehenmessung Werkstueck Hoehe: 3518 Debug Hal: engine stopped Debug Hal: engine revert stop ISR status: acd Werkstueck im Toleranzbereich: 3825 ISR status: ac9 Werkstueck zu klein/gross: 2726 ISR status: acb Kein Werkstueck in Hoehenmessung ISR status: ac3 Werkstueck in Weiche Debug Hal: gate open ISR status: ae3 Weiche offen ISR status: aeb Kein Werkstueck in Weiche ISR status: a6b Debug Hal: gate closed Debug Hal: engine stopped Werkstueck im Auslauf ISR status: a4b Weiche geshlossen ISR status: acb kein Werkstueck im Auslauf Debug Hal: all lights off ISR status: aca Werkstueck im Einlauf Debug Hal: engine right with normal speed Debug Hal: engine revert stop ISR status: acb Kein Werkstueck im Einlauf ISR status: acf Werkstueck im Toleranzbereich: 3787 ISR status: acd Werkstueck in Hoehenmessung Werkstueck Hoehe: 2464 Debug Hal: engine stopped	Werkstück hat Höhemessung verlassen  Werkstück ist an Weiche angekommen  Werkstück hat Weiche passiert  Werkstück kommt auf Rutsche  Werkstück ist aussortiert  Testlauf mit WS1 beendet Nächstes Werkstück auf Band gelegt          IRQ von Höhensensor  IRQ von Höhensensor   Eigene Messung ohne IRQ          Werkstück verlässt Höhenmessung Weiche wird geöffnet          Werkstücke am Ende angekommen Weiche schließen          Werkstück vom Ende entnommen  Nächstes Werkstück auf Band gelegt
--	---

<p> Debug Hal: engine revert stop  ISR status: ac9  Werkstueck zu klein/gross: 2735  ISR status: acb  Kein Werkstueck in Hoehenmessung  ISR status: ac3  Werkstueck in Weiche  Debug Hal: gate open  ISR status: ae3  Weiche offen  ISR status: aeb  Kein Werkstueck in Weiche  ISR status: a6b  Debug Hal: gate closed  Debug Hal: engine stopped  manualTurnover: 1  Debug Hal: all lights off  Werkstueck im Auslauf  Debug Hal: yellow light on  ISR status: a4b  Weiche geshlossen  Debug Hal: yellow light off  Debug Hal: yellow light on  ISR status: acb  kein Werkstueck im Auslauf  Debug Hal: all lights off  ISR status: aca  Werkstueck im Einlauf  Debug Hal: engine right with normal speed  Debug Hal: engine revert stop  ISR status: acb  Kein Werkstueck im Einlauf  ISR status: acf  Werkstueck im Toleranzbereich: 2609  ISR status: acd  Werkstueck in Hoehenmessung  Werkstueck Hoehe: 2437  Debug Hal: engine stopped  Debug Hal: engine revert stop  ISR status: ac9  Werkstueck zu klein/gross: 3128  ISR status: acd  Werkstueck im Toleranzbereich: 3595  ISR status: ac9  Werkstueck zu klein/gross: 2739  ISR status: acb  Kein Werkstueck in Hoehenmessung  ISR status: adb  Werkstueck Metall  ISR status: ad3  Werkstueck in Weiche  ISR status: ac3  Werkstueck kein Metall  ISR status: acb  Kein Werkstueck in Weiche  ISR status: a8b  Rutsche voll  ISR status: acb  Rutsche nicht voll  Debug Hal: engine stopped </p>	<p>Werkstück muss gewendet werden</p> <p>Nächstes Werkstück auf Band gelegt</p> <p>Werkstück enthält Metall</p> <p>Weiche nicht öffnen</p>
--	--



## **6.4 Abnahmetest**

## **6.5 Testplan**

## **6.6 Testprotokolle und Auswertungen**

# **7 Projektplan**

## **7.1 Verantwortlichkeiten**

Innerhalb der Gesprächssitzung zur Organisation des Teams wurde sich darauf verständigt, eine demokratische Grundordnung zu verfolgen. Entscheidungen werden gemeinsam im Team getroffen.

## **7.2 Projektstrukturplan**

# **8 Lessons Learned**

## **Glossar**

## **Abkürzungen**

## **Anhänge**