

Dokumentation Praktikum 4

Inhalt

1. Dokumentorganisation.....	1
1.1. Autorenliste	1
1.2. Versionen.....	1
2. Analyse.....	2
3. Requirements/Tests.....	2
3.1. System Requirements.....	2
3.2. System Tests	2
3.3. Software Requirements für neue Komponente	3
3.4. Software Tests für neue Komponente	3
4. Design	3
5. Integrationstests	4
6. Fazit.....	4
7. Anhang.....	4

1. Dokumentorganisation

1.1. Autorenliste

Kürzel	Name
LMN	Prof. Dr. Thomas Lehmann
BEY	Jannik Beyerstedt
DAN	Daniel Friedrich
KRH	Martin Kroh
LIL	Sascha Marc Lilie

1.2. Versionen

Version	Erstellt	Autor	Kommentar
0.1	02.06.14	LMN	Initiale Version des Templates.
0.2	12.11.14	LMN	Überarbeitung des Templates.
0.3	25.11.14	KRH	Erster Entwurf für die Softwarekomponente
0.4	01.12.14	BEY	Ergänzung um die Artefakte aus Labor 3

2. Analyse

Wir haben zuerst aus der Aufgabenstellung versucht die Requirements an die Produktionsanlage als Gesamtsystem zu rekonstruieren. Außerdem haben wir, um einen Überblick über den vorhandenen Code zu erhalten, ein Klassendiagramm erstellt.

Im nächsten Schritt ging es dann darum die neuen Anforderungen in Requirements, Tests und die entsprechenden Code-Strukturen umzusetzen. Die hauptsächliche Frage war hierbei, wie weit wir die Tests führen sollen. Soll es nur ein Unit-Test für die neue Komponente allein sein, oder soll auch die gesamte Anlage in Software getestet werden. Diese Frage hat sich dann aber am Anfang des ersten Laborterminals geklärt.

Daraufhin mussten dann die Höhenlinien mit entsprechenden Toleranzen aus den vorgelegten Messprotokollen herausgearbeitet werden, sowie eine Methode entwickelt werden, wie aus den Höhendaten die verschiedenen Baufehler erkannt werden.

3. Requirements/Tests

3.1. System Requirements

1. Die Produktionsanlage untersucht Fertigteile auf metallische Verunreinigungen und sortiert Teile mit metallischen Verunreinigungen über eine Rutsche aus. Alle anderen Teile werden bis an das Ende des Förderbandes transportiert.
2. Durch Prüfung der Höhenlinie werden Fertigteile mit Baufehlern erkannt und an den Anfang des Förderbandes zurück transportiert. Während des Rücktransportes und bis zum Entnehmen des Teils blinkt die rote Lampe der Ampel der Produktionsanlage.
Es werden dabei drei Arten von Fertigteilen erkannt:
 - 2.1. Gutteil: Bauteil von 4x4 Einheiten Grundfläche und 2 Einheiten Höhe, Massiv
 - 2.2. Baufehler Typ 1: Ein Baustein fehlt
 - 2.3. Baufehler Typ 2: Ein Baustein versetzt angebaut
3. Die Fertigteile werden, um korrekt erkannt zu werden, immer so auf das Förderband gelegt, dass die Seite mit der vollen Bauhöhe senkrecht zur Beförderungsrichtung steht.
4. Wenn ein fehlerhaftes Teil erkannt wurde, wird die Art des Fehlers (siehe Req. 2) in ein Log-File geschrieben, sowie auf der Konsole ausgegeben.
5. Wenn das fehlerhafte Teil zurück transportiert wird, wird dieses bis an den Anfang des Förderbandes transportiert und verweilt dort, bis es manuell entnommen wird. Solange blinkt die rote Lampe der Ampel weiter.
Nach dem Entnehmen geht die Produktionsanlage wieder in den normalen Betrieb über.

3.2. System Tests

- A. Die Produktionsanlage wird nacheinander mit Gutteilen, Teilen mit metallischen Verunreinigungen, sowie Fertigteilen mit Produktionsfehlern beladen. Es wird dabei immer nur ein Teil auf den Anfang des Förderbandes gelegt, wenn sich kein anderes Teil auf dem Förderband befindet.
Die Teile mit metallischen Verunreinigungen werden auf die Rutsche gelenkt und somit aussortiert.
Die Teile mit Baufehlern werden wieder an den Anfang transportiert. Währenddessen blinkt die rote Lampe der Ampel. Wenn das Teil am Anfang des Förderbandes zum stehen kommt, wird es vom Bediener entnommen und die Anlage geht in den normalen Betrieb über. Auf der Konsole, sowie in der Log-Datei ist der korrekte Fehlertyp zu lesen. Gutteile werden bis an das Ende des Förderbandes transportiert.
Dieser Test muss zu 100% erfolgreich sein.

3.3. Software Requirements für neue Komponente

51. Die neue Software-Komponente wird unabhängig von der Produktionsanlage mit den gegebenen Sensor-Logdaten per Unit-Test getestet.
52. Der Hözensensor liefert für die möglichen Höhenebenen (gem. Req. 2) Werte in den folgenden Wertebereichen:
 - 52.1. volle und korrekte Bauhöhe: Wert 3082 ± 60
 - 52.2. mittlere Bauhöhe (Fehler): Wert 3380 ± 60
 - 52.3. Förderband (kein Teil): Wert 3780 ± 10
53. Aus Req. 52 ergeben sich folgende Höhenlinien für die verschiedenen Arten von Fertigteilen (gem Req. 2):
 - 53.1. Gutteil: Förderband \rightarrow volle Höhe \rightarrow Förderband
 - 53.2. Typ 1: Förderband \rightarrow volle Höhe \rightarrow mittlere Höhe \rightarrow Förderband
 - 53.3. Typ 2: Förderband \rightarrow volle Höhe \rightarrow mittlere Höhe \rightarrow volle Höhe \rightarrow Förderband

3.4. Software Tests für neue Komponente

- N. Die Software-Komponente wird per Unit-Test mit Daten, die die drei unterschiedlichen Fertigteil-Arten nach Req. 53 gespeist. Daraufhin muss über die Schnittstelle result() den Daten entsprechend eine 1 ausgegeben werden, wenn das Teil fehlerhaft ist.
Es wird dabei mit jeweils einem repräsentativen Datensatz getestet.
Außerdem muss in dem Log-File der korrekte Fehlertyp ausgegeben werden, wenn das Teil einen Fehler aufweist.

4. Design

Wir haben zuerst ein Klassendiagramm aus dem vorliegenden Code erstellt. Dieses hat uns erst einmal einen Überblick über die verschiedenen Komponenten des Software verschafft. Welche Klassen gibt es, welche Methoden haben diese, welche Interfaces gibt es und welche Klasse benutzt diese. Alle diese Fragen lassen sich sehr gut aus dem Klassendiagramm ablesen.

Außerdem kann an diesem Diagramm gut erkannt werden, welches Interface die neue Komponente implementieren muss und welche Klassen für den Unit-Test benötigt werden.

Danach wurde der, der Steuerung zugrunde liegende, Zustandsautomat aus dem Code rekonstruiert. Dieser muss nämlich für die neue Funktionalität erweitert werden, damit das fehlerhafte Bauteil wieder an den Anfang des Förderbandes transportiert wird. Dafür sollte der vorhandene Zustandsautomat bekannt sein, wofür dich ein entsprechendes Diagramm arbeitet.

Bei dem Diagrammteil, der das bestehende System abbildet, wurden die selben Begriffe für Transitions und States, wie im Code verwendet. Somit kann schnell eine Verbindung zwischen Code und Diagramm hergestellt werden. Den neuen Teil haben wir hingegen von den Begrifflichkeit allgemeiner gehalten, da dieser noch keine exakte Implementierung vorgeben soll.

Die beiden Diagramme sind im Anhang zu finden.

5. Integrationstests

<Wie haben Sie die modifizierten Teile / vollständige Funktion getestet? Was war das Ergebnis der Tests?>

6. Fazit

<Geben Sie kurz und **ehrlich** an, was Sie an Erkenntnissen aus diesem Praktikum mitgenommen haben.>

<An welchen Stellen hatten Sie die meisten technischen/organisatorischen Probleme?>

7. Anhang

<Quellcode ist elektronisch abzugeben>