



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Aria Rafi Nazari

Prozesskontrolle für ein mittelständiges Unternehmen

*Fakultät Technik und Informatik
Studiendepartment Informatik*

*Faculty of Engineering and Computer Science
Department of Computer Science*

Aria Rafi Nazari

Prozesskontrolle für ein mittelständiges Unternehmen

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Stefan Sarstedt
Zweitgutachter: Prof. Dr. Ulrike Steffens

Eingereicht am: 03.11.2014

Aria Rafi Nazari

Thema der Arbeit

Prozesskontrolle für ein mittelständiges Unternehmen

Stichworte

Prozesskontrolle, Softwareengineering, Prozessautomation, WPF, MVVM, FTP, BPMN

Kurzzusammenfassung

Diese Arbeit befasst sich mit der Prozesskontrolle für ein mittelständiges Unternehmen mit Sitz in Deutschland und Fertigungsdienstleistungen in Schwellenländern. Hierfür wird der Prozess zur Produktionsüberwachung und -erfassung, die in den Schwellenländern erfolgen, analysiert und optimiert. Die Transparenz und die Prozessgeschwindigkeit soll gesteigert werden. Die Mitarbeiter haben keine Möglichkeit, in den Fabriken das ERP-System des Unternehmens zu nutzen. Deswegen wurde eine Anwendung entwickelt, die den Arbeitsaufwand für sie reduziert und den Produktionsstand transparenter darstellt.

Aria Rafi Nazari

Title of the paper

Process controlling for a medium-sized company

Keywords

process controlling, software engineering, process automation, WPF, MVVM, FTP, BPMN

Abstract

This paper is about process controlling for a medium-sized company with registered seat in Germany and manufacturing services in emerging economies. For this purpose the process of production monitoring and recording, which takes place in emerging economies, is analyzed and optimized to enhance the transparency and process speed. Since the employees can not access the ERP system of the company when they are in the factories, an application was developed to reduce the workload and to increase the transparency of the production status.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Schmidt Group GmbH	1
1.2. Problemstellung	2
1.3. Ziel der Arbeit	3
1.4. Aufbau der Arbeit	3
2. Grundlagen	5
2.1. Prozesskontrolle	5
2.2. Business Process Modelling	6
2.2.1. BPM-Kreislauf nach camunda	6
2.2.2. Camunda BPMN-Framework	8
2.2.3. Business Process Modelling Notation	9
2.3. Windows Presentation Foundation	11
2.4. Model-View-Controller	11
2.5. Model-View-ViewModel	12
2.6. Entwicklungsumgebung des ERP-Systems	13
3. Analyse	14
3.1. Prozesserhebung	14
3.1.1. Stakeholder	14
3.1.2. Ablauf des IST-Zustandes	15
3.2. Prozessdokumentation	17
3.3. Prozessanalyse	18
3.4. Prozesskonzeption	20
3.4.1. SOLL-Aufnahme	20
3.4.1.1. Erfassung des SOLL-Zustandes	20
3.4.1.2. Ablauf des SOLL-Zustand	21
3.4.1.3. SOLL-Konzept	22
3.4.2. Anforderung an die Architektur	24
3.4.3. SOLL- Prozessmodell	26
3.5. Anforderungen an das SL-System	27
3.5.1. Funktionale Anforderungen	27
3.5.2. Nichtfunktionale Anforderungen	29
3.5.3. Priorisierung	31

3.6.	Anforderungen an das ERP-System	32
3.6.1.	Funktionale Anforderung	32
3.6.2.	Nichtfunktionale Anforderungen	33
3.7.	Vergleichbare Software und Lösungen	34
3.7.1.	MiniSell	34
3.7.2.	Fazit	34
4.	Spezifikation	35
4.1.	Glossar	35
4.2.	Fachliches Datenmodell	36
4.2.1.	Zustandsgraph für das SL-System	37
4.2.2.	Zustandsgraph für das ERP-System	38
4.3.	Anwendungsfälle	39
4.3.1.	SL-System	39
4.3.1.1.	Formulare aktualisieren	40
4.3.1.2.	Formular Auswahl	41
4.3.1.3.	Produkt Auswahl	42
4.3.1.4.	Produkt erfassen/prüfen	43
4.3.1.5.	Produkt übermitteln	44
4.3.1.6.	Produkt freigeben	45
4.3.1.7.	Produkt Freigabe entfernen	46
4.3.1.8.	User Stories	47
4.3.2.	ERP-System	47
4.3.2.1.	Produkt übermitteln an FTP	48
4.3.2.2.	Produkt übermitteln an ERP	49
4.3.2.3.	Produkt erhalten	50
4.4.	Dialoge	51
4.4.1.	File	51
4.4.2.	FTP	52
4.4.3.	Options	52
4.4.4.	Produkt Auswahl	53
4.4.5.	Formular Auswahl	53
5.	Design und Realisierung	54
5.1.	Allgemeine Designentscheidungen	54
5.1.1.	Verteilungssicht	55
5.1.2.	Client-Server Architektur	56
5.1.3.	Nebenläufigkeit	57
5.1.3.1.	Pessimistische Synchronisation	57
5.1.3.2.	Optimistische Synchronisation	58
5.1.4.	Ordnerstruktur	59
5.1.5.	Datenaustauschformat	61

5.1.6.	Sicherheit	62
5.1.6.1.	File Transfer Protokoll	63
5.1.6.2.	Dateien verschlüsseln	63
5.2.	ERP-System (Server)	63
5.2.1.	Sichten	64
5.2.1.1.	Bausteinsicht	64
5.2.1.2.	Laufzeitsicht	65
5.2.2.	Strukturen	66
5.2.2.1.	XML-Struktur	66
5.2.2.2.	XAML-Struktur (Formular View)	69
5.3.	SL-System (Client)	71
5.3.1.	Architekturmuster	71
5.3.2.	Frameworks, Libraries und Tools	71
5.3.2.1.	DevExpress	71
5.3.2.2.	FTP-Library	72
5.3.2.3.	ftpdmin	72
5.3.3.	Sichten	72
5.3.3.1.	Bausteinsicht	72
5.3.3.2.	Laufzeitsicht	74
5.3.4.	GUI Design	80
5.3.5.	Probleme	81
5.3.5.1.	Nebenläufigkeit	81
5.3.5.2.	Fehlerbehandlung	82
5.3.6.	Installationsprogramm	83
5.3.7.	Tests	84
6.	Evaluierung	85
6.1.	Interview	85
6.2.	Prototyp	86
6.3.	Auswertung	87
6.4.	Fazit	88
7.	Schluss	89
7.1.	Zusammenfassung	89
7.2.	Fazit	91
7.3.	Ausblick	92
	Abkürzungsverzeichnis	93
	Glossar	95
	Abbildungsverzeichnis	97
	Tabellenverzeichnis	99

Auflistungsverzeichnis	100
Literaturverzeichnis	100
A. Anhang	104

1. Einleitung

Die Herstellung vieler Produkte, die in Europa entworfen und vertrieben werden, erfolgt in Schwellenländern. Typisch für diese Schwellenländer sind ein niedriges Lohnniveau und Defizite in der Infrastruktur. Die beauftragenden Unternehmen in Europa stehen vor der Herausforderung, die Produktion in den Schwellenländern mit einem ähnlichen Detaillierungsgrad wie bei einer Inhouse-Produktion zu verfolgen. Nur so lassen sich teure Qualitäts- und Terminabweichungen bei der Lieferung vermeiden.

Eine Branche, in der die Risiken besonders hoch erscheinen, ist die Textil-Branche¹. Produkte lassen sich nur in einer sehr kurzen Zeitspanne am Markt absetzen (zum Beispiel Saisonwaren). Nachlieferungen sind genau wie Qualitätsabweichungen nicht akzeptabel und müssen vermieden werden. Informationstechnologie kann hier helfen, die notwendige Transparenz zu erreichen und damit die Steuerbarkeit der Produktion und der Lieferanten im Sinne des bestellenden Unternehmens sicherzustellen. Im Mittelpunkt steht hierbei die Suche nach der Anpassung an die Gegebenheiten im Produktionsland und die Integration in die IT-Systeme und -Prozesse des europäischen Unternehmens.

1.1. Schmidt Group GmbH

Die Schmidt Group GmbH² wurde 1906 gegründet und ist ein in der vierten Generation inhabergeführtes Unternehmen der Textilindustrie mit Sitz in Norderstedt. Als Besitzer von Marken wie Chiemsee, Kappa und Colorado sowie als Lieferant für große Handelsunternehmen stellt sich die Schmidt Group den Herausforderungen, die sich aus der Produktion in Schwellenländern ergeben.

¹http://www.fmm-magazin.de/weltweite-branchenanalyse-textil-und-bau-mit-dem-hoechsten-risiko-finanzen-mm_kat52_id7530.html. Abgerufen am 30.09.14

²<http://www.fashioncenter.net>. Abgerufen am 30.09.14

1.2. Problemstellung

Am Standort Norderstedt betreibt die Schmidt Group ein **Enterprise-Resource-Planning**-System (ERP-System), welches den Produktionsfortschritt eines Produktes in den Fabriken der Schwellenländer verwalten kann. Diese Fabriken gehören den Produzenten, welche die Schmidt Group beliefern. Deswegen hat die Schmidt Group Mitarbeiter in den Schwellenländer, die den Produktionsfortschritt eines Produktes überwachen und mitteilen. Das ERP-System der Schmidt Group befindet sich in einem privaten Intranet. Es ist von außen somit nur mit einem **Virtual Private Network** (VPN)-Client möglich, auf das ERP-System zuzugreifen. Da den Mitarbeitern in den Fabriken des Öfteren keine Verbindung zum Internet zur Verfügung steht (zum Beispiel durch Unterbrechungen), können diese keinen VPN-Client nutzen, um auf das ERP-System zuzugreifen. Sie sind daher nicht in der Lage, Informationen (zum Beispiel wie viele Schuhe in Kalenderwoche 10 produziert wurden) ohne Umwege an das ERP-System zu übermitteln.

Aktuell geschieht die Erfassung des Produktstandes im ERP-System mehrstufig:

- In den Fabriken werden die Produkte inspiziert und deren Ergebnisse vor Ort (lokal) in Excel und auf Papier erfasst.
- Die während der Inspektionen gesammelten Daten werden von den Regionalverantwortlichen zusammengetragen, logisch geprüft und als Excel-Dateien nach Norderstedt gemailt.
- In Norderstedt erfolgt die Übertragung der Informationen (durch Abtippen der Daten per Hand) aus den Excel-Tabellen in das ERP-System.

Die Daten werden mehrfach nicht-systemunterstützt (zum Beispiel durch Email oder Telefon) weitergegeben³, daher passiert es von Zeit zu Zeit, dass die Daten fehlerhaft (zum Beispiel durch falsches Abtippen) importiert werden. Dazu werden die Informationen der inspizierten Produkte von den Regionalverantwortlichen gesammelt und erst später nach Norderstedt weiter übermittelt. Folglich kann es Tage oder Wochen dauern, bis Informationen einzelner Produkte bekannt sind. Durch diese Verzögerung können wichtige Entscheidungen (zum Beispiel wird die Produktion verschifft oder per Flugzeug transportiert) häufig falsch oder erst verspätet getroffen werden. Dementsprechend bietet der Prozess, also die Erfassung der Daten von den Produkten bis zur Eintragung in das ERP-System, noch einiges an Optimierungspotenzial. Deswegen soll

³Von Inspektoren zu ihrem Leiter, danach an den nächsten Mitarbeiter bis zur Eingabe im ERP-System

dieser Prozess bewertet und verbessert werden. Durch den Einsatz von Systemunterstützungen kann dieser Prozess *schneller* (Prozessgeschwindigkeit) und *transparenter* gestaltet werden. Diese beiden Punkte werden als Optimierungsziele festgelegt.

1.3. Ziel der Arbeit

Ziel dieser Arbeit ist es, den Prozess zur Produktionsüberwachung und -erfassung mittels Prozesskontrolle zu bewerten und zu verbessern. Der Prozess soll transparenter und schneller (Prozessgeschwindigkeit) werden. Hierfür wird er zunächst analysiert, um mögliche Optimierungen zu identifizieren. Anhand der Analyse wird ein Konzept entworfen, mit dem der Prozess verbessert werden soll. Im Rahmen einer Evaluierung erfolgt die Prüfung, ob das realisierte Konzept zur Optimierung beitragen kann.

Das Ergebnis dieser Arbeit soll aufzeigen, wie das entworfene Konzept und die Realisierung die Transparenz und Prozessgeschwindigkeit des Prozesses verbessern könnte.

Die Umsetzung der Anbindung an das ERP-System wird nicht Teil dieser Arbeit sein. Zur Evaluierung werden bestehende Daten aus dem ERP-System genutzt.

1.4. Aufbau der Arbeit

Der Aufbau dieser Arbeit ist in sieben Kapitel unterteilt.

Kenntnisse in Unified Modeling Language (UML) und Client-Server-Architektur werden für diese Arbeit vorausgesetzt.

Zunächst werden im zweiten Kapitel, *Grundlagen*, die Themen beschrieben, die für das Verständnis sowie die Umsetzung von Relevanz sind. Zu diesen Grundlagen gehören Prozesskontrolle, Business Process Modelling, die Windows Presentation Foundation mit Model-View-Controller und Model-View-ViewModel und die Entwicklungsumgebung des ERP-Systems.

Im dritten Kapitel, *Analyse*, wird am Anfang der aktuelle IST-Zustand erhoben und dokumentiert. Dieser wird anschließend analysiert, um die Schwachstellen des IST-Zustandes zu identifizieren. Ausgehend von der Analyse werden mögliche Lösungen beschrieben. Es wird eine Lösung gewählt und auf Basis dieser ein SOLL-Konzept und ein Anforderungsdokument

1. Einleitung

entwickelt. Zuletzt werden die Anforderungen mit den derzeit erhältlichen Softwarelösungen verglichen.

Im vierten Kapitel, *Spezifikation*, werden die Anforderungen spezifiziert, um den Aufbau des Systems näher zu erläutern.

Im fünften Kapitel, *Design und Realisierung*, wird ein Konzept für die Realisierung des Gesamtsystems und dessen Umsetzung beschrieben. Zum Konzept des Gesamtsystems tragen die Analyse und Spezifikation bei.

Mit dem sechsten Kapitel, *Evaluierung*, wird der Prototyp vorgestellt. Aufgrund der Auswertungen aus den Interviews können später zusätzliche Weiterentwicklungsmöglichkeiten vorgenommen werden.

Im siebten Kapitel, *Schluss*, wird die Arbeit zusammengefasst, ein Fazit formuliert und ein kurzer Ausblick gegeben.

2. Grundlagen

In diesem Kapitel werden grundlegende Themen beschrieben, die für das Verständnis dieser Arbeit relevant sind. Zunächst erfolgt die Klärung der Begriffe der Prozesskontrolle. Danach werden die Grundlagen des Business Process Modelling erörtert. Anschließend werden Windows Presentation Foundation mit Model-View-Controller-Entwurfsmuster und Model-View-ViewModel-Entwurfsmuster beschrieben. Zuletzt wird das verwendete ERP-System erklärt.

2.1. Prozesskontrolle

Ein Prozess wird als eine Reihe von Tätigkeiten definiert, die von Menschen oder Maschinen ausgeführt werden, um ein oder mehrere Ziele zu erreichen.

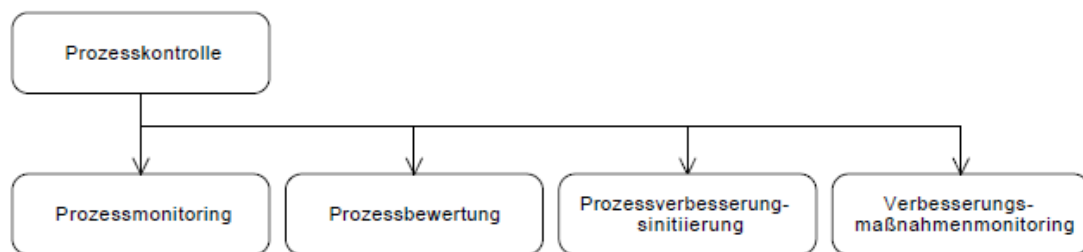


Abbildung 2.1.: Funktionsbaum Prozesskontrolle

Unter Prozesskontrolle (auch Prozesscontrolling genannt) nach Abbildung 2.1⁴ versteht man die Beobachtung, Bewertung und Regelung eines Prozesses (vgl. Hackl, 2004, S.2). Ziel der Prozesskontrolle ist es,

- Prozesse zu überwachen, um Probleme zu erkennen.
- Prozesse zu bewerten, um eine Grundlage für Prozessverbesserungen zu schaffen.

⁴<http://semture.de>. Abgerufen am 30.09.14

- Prozessverbesserungen zu initiieren, um die Prozesse zu optimieren.
- Prozessverbesserungen zu überwachen, um Probleme bei der Umsetzung der Maßnahmen zu erkennen.

2.2. Business Process Modelling

Das Business Process Modelling (BPM), auch Geschäftsprozessmanagement genannt, „ist ein systematischer Ansatz, um sowohl automatisierte als auch nicht-automatisierte Prozesse zu erfassen, zu gestalten, auszuführen, zu dokumentieren, zu messen, zu überwachen und zu steuern und damit nachhaltig die mit der Unternehmensstrategie abgestimmten Ziele zu erreichen“ (Schmidt, 2009).

BPM wird eingesetzt, um bestehende Prozesse zu verbessern, zu dokumentieren oder neue Prozesse einzuführen. Der Haupteinsatzzweck ist die Verbesserung bestehender Prozesse, bei der zum Beispiel das manuelle Abtippen eines Formulars durch eine automatisierte Wertübernahme ersetzt wird. Ferner wird BPM ebenfalls zur Dokumentation von Prozessen eingesetzt. Diese Art der Dokumentation wird zum Beispiel für eine Zertifizierung nach **DIN EN ISO 9000** (und folgende) gefordert.

Seltener wird BPM auch zur Neueinführung von Prozessen verwendet, um zum Beispiel auf veränderte Marktbedingungen zu reagieren (vgl. **Freund und Rücker, 2012**, S.2).

Die Business Process Modelling Notation (BPMN) wurde für die Modellierung von Prozessen entwickelt. Es ist eine grafische Spezifikationsprache und stellt Symbole zur Verfügung, um Prozessabläufe modellieren und dokumentieren zu können. Ausführlichere Informationen zum Business Process Model und Notation finden sich im **Freund und Rücker (2012)**.

2.2.1. BPM-Kreislauf nach camunda

Der camunda BPM-Kreislauf (siehe Abbildung 2.2) ist ein Vorgehensmodell für BPM. Dieses dient zur Analyse von Prozessen. Hierbei wird die Prozessanalyse, -konzeption, -umsetzung und -controlling zyklisch durchgelaufen um eine kontinuierliche Prozessverbesserung zu erreichen.

Das Vorgehensmodell ist zur Anwendung auf separate Prozesse gedacht und kann somit für jeden Prozess einzeln durchlaufen werden. Angewendet wird es typischerweise in zwei

2. Grundlagen

Situationen: entweder soll ein existierender Prozess dokumentiert und/oder verbessert werden, oder es soll ein neuer Prozess eingeführt werden.

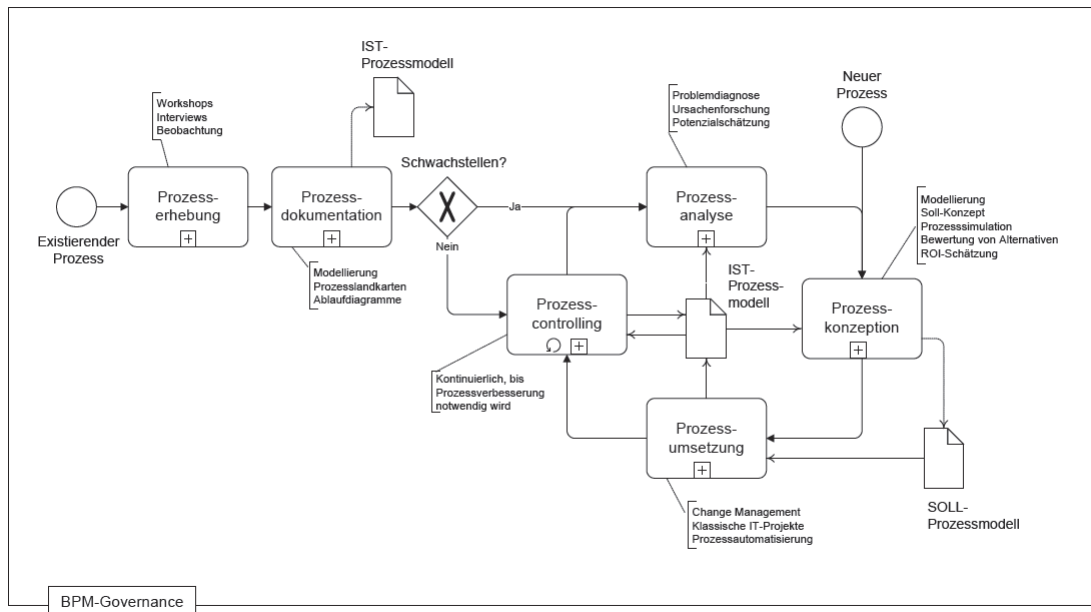


Abbildung 2.2.: Der camunda BPM-Kreislauf (Freund und Rücker, 2012, S.4)

Ein bereits existierender Prozess wird zunächst in der Prozess-erhebung erhoben. Die Erhebungstechniken sind zum Beispiel Workshops oder Einzelinterviews mit den am Prozess beteiligten Personen. Mit den Erkenntnissen aus der Prozess-erhebung wird ein IST-Prozessmodell erstellt. Die Prozessdokumentation kann aus unterschiedlichen Diagrammen und weiterführenden Beschreibungen zusammengesetzt werden. In der Prozessanalyse werden die Schwachstellen eingegrenzt und eindeutig identifiziert. Die Prozessanalyse wird durchgeführt, wenn durch eine erstmalige Dokumentation eines Prozesses oder im Rahmen des Prozesscontrollings Schwachstellen erkannt wurden, die sich nicht durch kleinere Anpassungen beheben lassen (vgl. Freund und Rücker, 2012, S.4). Die in der Prozessanalyse erkannten Ursachen für die Schwachstellen sind die Ausgangspunkte für eine Prozesskonzeption. Eine Prozesskonzeption findet auch bei einem neu eingeführten Prozess statt. In beiden Fällen ist das Ergebnis ein SOLL-Prozessmodell. Anschließend erfolgt die Prozessumsetzung, bei der das SOLL-Prozessmodell umgesetzt wird. Das Ergebnis der Umsetzung ist ein IST-Prozess, der dem SOLL-Prozessmodell entspricht und somit dokumentiert ist (vgl. Freund und Rücker, 2012, S.5).

2.2.2. Camunda BPMN-Framework

Das camunda BPMN-Framework (caBPMN) ist eine Sammlung von Erfahrungen der camunda Company, welche eine Hilfe anbietet, in welcher Situation welche BPMN-Symbole und BPMN-Konstrukte angewendet werden können.

Der Schwerpunkt des caBPMN liegt auf Projekten, bei denen die Prozesse eine verbesserte IT-Unterstützung benötigen und im SOLL-Zustand modelliert werden müssen. Das caBPMN besteht aus vier Ebenen (Freund und Rücker, 2012, S.14).

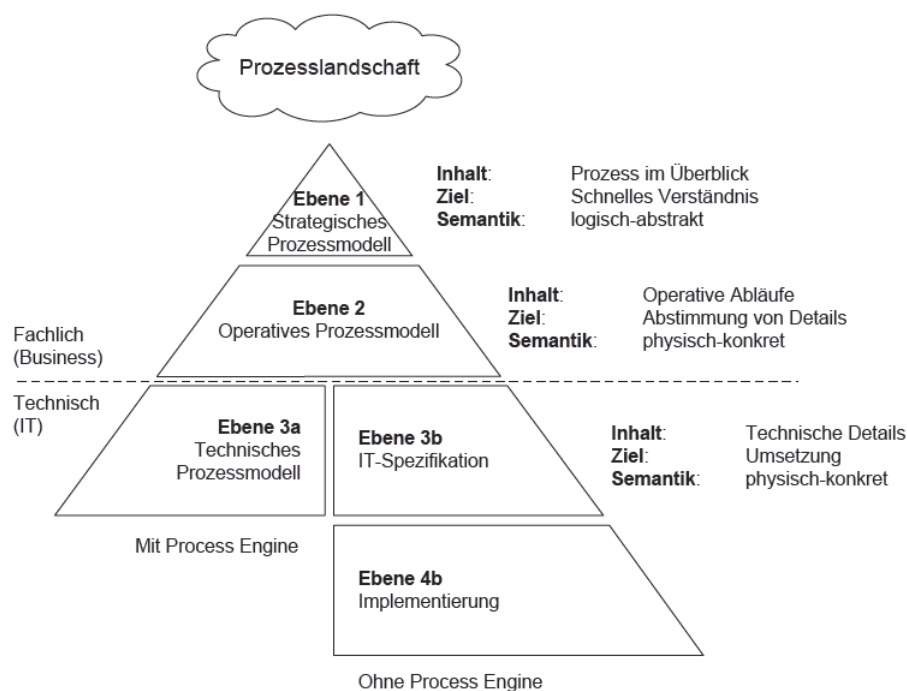


Abbildung 2.3.: camunda BPMN-Framework (Freund und Rücker, 2012, S.15)

Bei der Ebene 1, „Strategisches Prozessmodell“, geht es darum, eine grundsätzliche und ergebnisorientierte Darstellung des Prozesses zu liefern. Der Prozess wird anhand weniger Schritte aus der Vogelperspektive skizziert. Mögliche Varianten oder Fehler werden nicht dargestellt.

Bei der Ebene 2, „Operatives Prozessmodell“, werden die operativen Details der tatsächlichen Abwicklung betrachtet. Es ist eine detailliertere Beschreibung als Ebene 1. Diese wird später für das technische Prozessmodell benötigt. Wenn ein Prozess nur analysiert und dokumentiert

2. Grundlagen

werden soll, wird aufgrund des schlechten Aufwand-Nutzen-Verhältnisses in der Praxis meist auf die Ebene 2 verzichtet.

Bei der Ebene 3a, „Technisches Prozessmodell“, geht es um die Verfeinerung des aus Ebene 2 erhaltenen Prozessmodells, um es in einer Process Engine ablaufen lassen zu können. Dies funktioniert erst seit BPMN 2.0.

Die Process Engine steuert die Geschäftsprozesse, die durch technische Prozessmodell modelliert sind (vgl. **Freund und Rücker, 2012, S.7**).

Die Ebene 3b, „IT-Spezifikation“, wird nur verwendet, wenn es keine Process Engine gibt und die Prozesslogik in einer herkömmlichen Programmiersprache (zum Beispiel C#) umgesetzt werden soll. In diesen Fall muss die Umsetzung noch technisch spezifiziert (zum Beispiel mit Unified Modeling Language (UML)) werden. Für BPMN spielt die Ebene 3b eine untergeordnete Rolle.

In der Ebene 4b, „Implementierung“, erfolgt die technische Umsetzung des Prozesses auf einer herkömmlichen Plattform (alle vgl. **Freund und Rücker, 2012, S.16**).

Nur die Ebenen 1 bis 3a sind einhundertprozentig BPM-relevant. Für die Ebenen 3b und 4b wird Wissen über UML benötigt.

2.2.3. Business Process Modelling Notation

Die BPMN wurde für die Modellierung von Prozessen entwickelt. In Abbildung 2.4 werden die Basiselemente⁵ für BPMN beschrieben. In einem Prozess werden bestimmte Aufgaben erledigt (Aktivität), manche Ereignisse (Ereignis) passieren nur unter bestimmten Bedingungen (Gateway). Artefakte liefern zusätzliche Informationen zum Prozess, können aber keinen direkten Einfluss auf die Reihenfolge der Flussobjekte haben (vgl. **Freund und Rücker, 2012, S.21**).

⁵Weitere Informationen über BPMN 2.0 Symbole(Poster) unter http://www.bpmb.de/index.php/BPM_NPoster. Abgerufen am 30.09.14

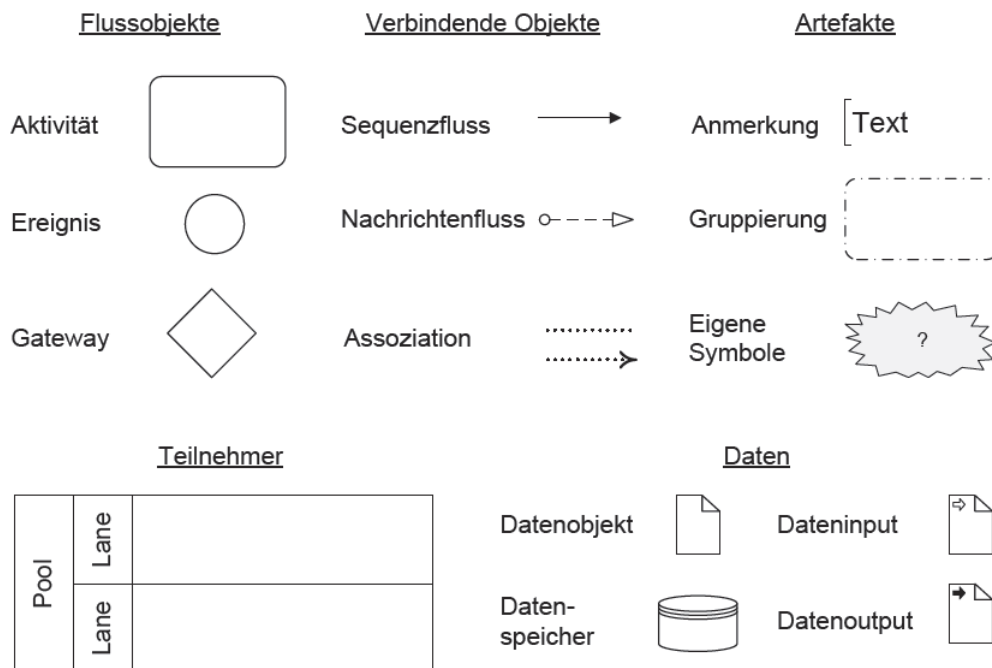


Abbildung 2.4.: Die Basiselemente der BPMN (Freund und Rucker, 2012, S.21)

Ein Diagramm kann einen oder mehrere Teilnehmer haben. Ein Teilnehmer ist für BPMN ein logisches Element, für das bestimmte Regeln gelten (Freund und Rucker, 2012, S.22):

- „Für einen Prozess existiert nur ein einziger Teilnehmer.“
- „Dieser Teilnehmer hat die totale Kontrolle über den Prozessfluss.“
- „Andere Teilnehmer können den Prozess dieses Teilnehmers nicht beeinflussen; unter Umständen wissen sie nicht einmal, wie er funktioniert.“
- „Der Teilnehmer ist dementsprechend für diesen Prozess verantwortlich.“
- „Wenn der Teilnehmer im Rahmen seines Prozesses mit anderen Teilnehmern interagieren möchte, muss er mit ihnen Nachrichten austauschen, was diese wiederum in ihren eigenen Prozessen entsprechend unterstützen müssen.“

Somit kann derselbe Prozess für die jeweiligen Teilnehmer ganz unterschiedlich aussehen, abhängig von der jeweiligen Perspektive. Das BPMN-Symbol für den Prozess eines Teilnehmers ist der Pool. Der Pool entspricht gleichzeitig dem Teilnehmer selbst, ein Teilnehmer kann aber

auch mehr als einen Prozess steuern. Die Flussobjekte werden innerhalb eines Pools bzw. Lane⁶ mit Sequenzfluss miteinander verbunden. Sollte eine Verbindung zwischen Flussobjekten über Pool-Grenzen hinweg erfolgen, dann muss ein Nachrichtenfluss verwendet werden (Freund und Rücker, 2012, S.23).

2.3. Windows Presentation Foundation

Die Windows Presentation Foundation (WPF) ist eine Klassenbibliothek zur Entwicklung von grafischen Benutzeroberflächen (GUI) und Teil des .NET-Framework.

Die Beschreibung der Benutzeroberfläche erfolgt mit der **Extensible Application Markup Language** (XAML), diese basiert auf **Extensible Markup Language** (XML) und unterliegt damit auch denselben Regeln wie XML (Wegener, 2013, S.6-7).

2.4. Model-View-Controller

Das Model-View-Controller-Entwurfsmuster (MVC) wurde in den 1970er vorgestellt. Ziel des MVC ist die Trennung von Logik, Benutzeroberfläche und (Benutzer-)Interaktionen (vgl. Balzert, 2011, S.62-68). Die Abbildung 2.5 von **Microsoft-MVC (2003)** beschreibt die drei Komponenten und deren Interaktionen miteinander.

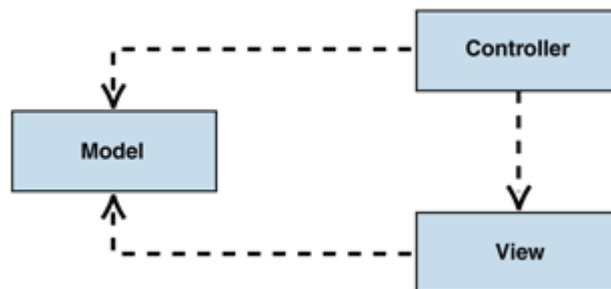


Abbildung 2.5.: Model-View-Controller-Entwurfsmuster

⁶Unterteilung eines Pools

Das *Model* beinhaltet die Logik und Daten der Anwendung.

Die *View* ist die Darstellungsebene und verwaltet die Anzeige der Informationen.

Der *Controller* ist für die Steuerung und Bearbeitung der Benutzerinteraktionen zuständig. Er übermittelt Veränderungen an Model und View (vgl. [Microsoft-MVC \(2003\)](#)).

2.5. Model-View-ViewModel

Das Model-View-ViewModel-Entwurfsmuster (MVVM) wurde von Microsoft entwickelt und ist eine Weiterentwicklung des Model-View-Controller-Entwurfsmusters. Das MVVM wurde speziell für das WPF-Framework entwickelt. Das Entwurfsmuster wurde so konzipiert, dass die Entwicklung der Benutzerschnittstelle, der Präsentations- und der Anwendungslogik vollständig getrennt erfolgen kann und sie in keiner direkten Beziehung zueinander stehen. Somit lassen sich die Komponenten leicht austauschen und erweitern (vgl. [Schmitz, 2014, S.45-46](#)). Die Abbildung 2.6 von [Microsoft-MVVM \(2014\)](#) beschreibt die drei Komponenten und deren Interaktionen miteinander.

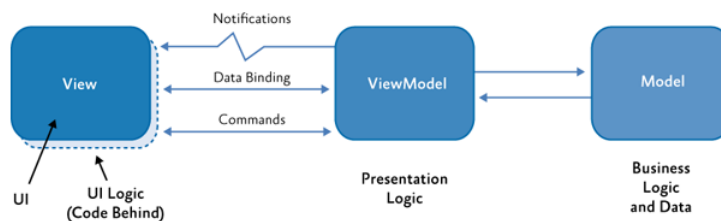


Abbildung 2.6.: Model-View-ViewModel-Entwurfsmuster

Das *Model* enthält die fachlichen Daten für das *ViewModel* und hat keinen Bezug zur *View*. Die Geschäftslogik kann zusätzlich im *Model* enthalten oder davon getrennt sein. Dann dient das *Model* rein als Daten-Container.

Die *View* besteht hauptsächlich aus XAML-Code, beschreibt das Aussehen der Darstellung und soll keine Ausführungslogik enthalten (darf aber UI-Logik enthalten). Ein *View* enthält Steuerelemente, die zur Interaktion zwischen Anwender und Anwendung benötigt werden. Zusätzlich kennt die *View* ihr *ViewModel*, dass durch die *DataContext*-Eigenschaft in der *View* gesetzt wird.

Das *ViewModel* unterstützt die *View* mit Darstellungslogik und bereitet die Daten für die Darstellung in der *View* auf. Es hat keine direkte Referenz zur *View* und kennt es somit nicht (vgl. [Schmitz, 2014, S.46-47](#)).

Durch das Databinding⁷ (deutsch Datenbindungssystem) können Änderungen von der View an das ViewModel weitergegeben werden und umgekehrt. Es unterstützt außerdem Validierungen (Eingabeprüfungen), die im ViewModel oder Model umgesetzt werden können. Das ViewModel stellt ebenso Befehle (Commands) für die View zur Verfügung. Diese kann man den Schaltflächen der View-Oberfläche zuweisen. Beim Klicken eines Buttons zum Beispiel wird der entsprechende Befehl (Command) ausgeführt (vgl. [Schmitz, 2014](#), S.46-48).

Der Vorteil des MVVM ist die Trennung zwischen der Präsentations- und Anwendungslogik. Somit können Modul-Tests geschrieben werden, ohne die Oberfläche anzuzeigen. Es ist auch möglich, die Oberfläche zu ändern, ohne die Programmlogik anzupassen (somit eine many-to-one Beziehung zwischen View und ViewModel) ([Microsoft-MVVM, 2014](#), vgl.).

2.6. Entwicklungsumgebung des ERP-Systems

Für die Entwicklung von Anwendungen für das ERP-System wird Konzept 16 von vectorsoft AG⁸ genutzt. Konzept 16 wurde 1979 entwickelt und ist ein proprietäres Datenbankmanagementsystem (angelehnt an relationale Datenbanksysteme) mit einer integrierten Entwicklungsumgebung. Zusätzlich hat es eine eigene, an C/C++ angelehnte, Programmiersprache mit Compiler und Debugger. Somit ist Konzept 16 eine prozedurale Programmiersprache und bietet keine Objektorientierung⁹.

⁷Informationen über Databinding unter <http://msdn.microsoft.com/en-us/ms752347>

⁸<http://www.vectorsoft.de/>. Abgerufen am 30.09.14

⁹http://de.wikipedia.org/wiki/Concept_16. Abgerufen am 30.09.14

3. Analyse

Dieses Kapitel befasst sich mit der Analyse der Problemstellung, welche in der Einleitung beschrieben wurde. Zunächst wird der Prozess erhoben und daraus ein IST-Prozessmodell erstellt. Mit Hilfe der Dokumentation des IST-Zustandes werden die Probleme analysiert und erfasst. Basierend auf der Prozessanalyse wird ein SOLL-Prozessmodell erstellt und aus diesem die Anforderungen an das System abgeleitet.

Zur Analyse des Prozesses wird der camunda BPM-Kreislauf (Abschnitt 2.2.1) und das camunda BPMN-Framework (Abschnitt 2.2.2) verwendet.

3.1. Prozesserhebung

Die *Produktionsüberwachung und -erfassung* ist ein existierender Prozess, der keine Dokumentation beinhaltet. Daher muss dieser zuerst erhoben werden (IST-Aufnahme). Die im Prozess auszuführenden Aufgaben sowie die beteiligten Akteure werden in mehreren Einzelinterviews identifiziert (vgl. Freund und Rücker, 2012, S.4).

3.1.1. Stakeholder

Als Stakeholder werden „Personen oder Organisationen, die bei der Erstellung des Systems mitwirken, sie beeinflussen oder am entwickelten System ein fachliches Interesse haben“ (Starke, 2014, S.10) bezeichnet.

Inspektor

Zuständig für die Inspektionen der Produkte in den Fabriken. Es gibt mehrere Inspektoren.

Mitarbeiter

Zuständig für den Verkauf der Produkte. Befindet sich in Norderstedt und hat Zugriff auf das ERP-System. Es gibt mehrere Mitarbeiter.

Regionalverantwortlicher

Zuständig für die Zuweisungen der Inspektionen der Produkte an seine Inspektoren. Erhält die erfassten Daten. Befindet sich in einem Landesbüro in dem Land, in dem das Produkt produziert wird. Es gibt mehrere Regionalverantwortliche.

Regionalverantwortlichen-Koordinator

Zuständig für die Leitung der Regionalverantwortlichen. Befindet sich in Norderstedt, um die gesammelten Daten in das ERP-System zu übertragen. Es gibt nur einen Regionalverantwortlichen-Koordinator in der Schmidt Group.

3.1.2. Ablauf des IST-Zustandes

Ein Regionalverantwortlicher wird von einer Fabrik benachrichtigt, dass der Auftrag für ein neues Produkt¹⁰ angenommen wurde und die Produktion begonnen hat. Anschließend weist er einem seiner Inspektoren die Produktinspektion zu und gibt ihm die entsprechenden Informationen¹¹. Der Inspektor inspiziert das Produkt in der Fabrik und erfasst die Daten¹² vor Ort in Excel und auf Papier. Ein Produkt wird bis zur Auslieferung inspiziert. Die Daten werden in unterschiedlichen Zeitabständen in die Landesbüros an die entsprechenden Regionalverantwortlichen weitergegeben. Der Regionalverantwortliche sammelt alle Daten von seinen Inspektoren, prüft die Daten (auf Richtigkeit), fügt diese in einer Excel-Tabelle (Abbildung 3.1, weitere Informationen im Anhang A.3) zusammen und mailt die Tabelle an den Koordinator in Norderstedt.

¹⁰Ein Produkt kann zum Beispiel ein Schuh sein, von dem 100.000 Stück produziert werden sollen.

¹¹Zum Beispiel Standort, Fabrikname usw.

¹²Zum Beispiel, wie viele wurden produziert, wie hoch ist der Output und wie viele Einheiten des Produkts haben die Qualitätskontrolle bestanden usw.

3. Analyse

	A	C	D	E	F	H	J	M	O	P	Q	U	W	Y
1								Date	20.Feb					
2	GENERAL INFORMATION							PRODUCTION AND FRI STATUS				FORECAST AND REMARKS		
3	Style	BC Merch	Agent	Agent Merch	Factory	BC QC	Order Qty.	Production Report Date	Output	Output Balance With 5% Extra	Delivery Date: Ex Ctg Date	Days Left to ship	Productivity/Line/Day	Remarks
4	1234	Max	Sty	Hans	DB	Homer	141750	16. Feb	123456	25382	20. Jan	-31	1500	Booked 1st FRI
5	5678	Peter	Mer	Hans	DB	Homer	298968	10. Feb	234567	79349	21. Feb	1	1800	Full Qty. Shipped.
6	12477	Paul	Sty	Fritz	DB	Homer	332748				10. Apr	49	1900	

Abbildung 3.1.: Im IST-Zustand verwendete (gekürzte) Excel-Tabelle zur Erfassung der Produktdaten. Jede Zeile entspricht einem Produkt. Die Zahlenwerte sind fiktiv.

Eine Übermittlung der bisher erfassten Daten nach Norderstedt zur Schmidt Group geschieht in Abständen von ungefähr zwei Wochen. In Norderstedt erfolgt die Übertragung der Daten von dem Regionalverantwortlichen-Koordinator in das ERP-System. Die Erfassung der Daten in das ERP-System kann je nach Umfang mehrere Tage dauern.

Nachfolgend werden die Prozessschritte noch einmal zusammengefasst:

1. Der Regionalverantwortliche erhält eine Auftragsbestätigung.
2. Der Regionalverantwortliche weist einem Inspektor die Überwachung zu und gibt ihm die Informationen.
3. Der Inspektor inspiziert das Produkt und erfasst den Produktstand bis zur Lieferung.
4. Der Inspektor übergibt den aktuellen Produktstand an den Regionalverantwortlichen.
5. Der Regionalverantwortliche fasst alle Informationen seiner Inspektoren in einer Excel-Tabelle zusammen und prüft diese zugleich.
6. Der Regionalverantwortliche sendet die Excel-Tabelle ungefähr alle zwei Wochen an den Regionalverantwortlichen-Koordinator in Norderstedt.
7. Der Regionalverantwortlichen-Koordinator überträgt die Informationen in das ERP-System.

3.2. Prozessdokumentation

Aus den Erkenntnissen der Prozesserhebung wird nun ein Ebene 1¹³-IST-Prozessmodell modelliert. Das vollständige IST-Prozessmodell (Abbildung 3.2^{14,15}) ist für das einfache Verständnis zu detailliert. Deswegen wurde für die Interviews ein vereinfachtes Modell mit wenigen Symbolen vorgeführt. Dieses findet sich im Anhang A.1 wieder.

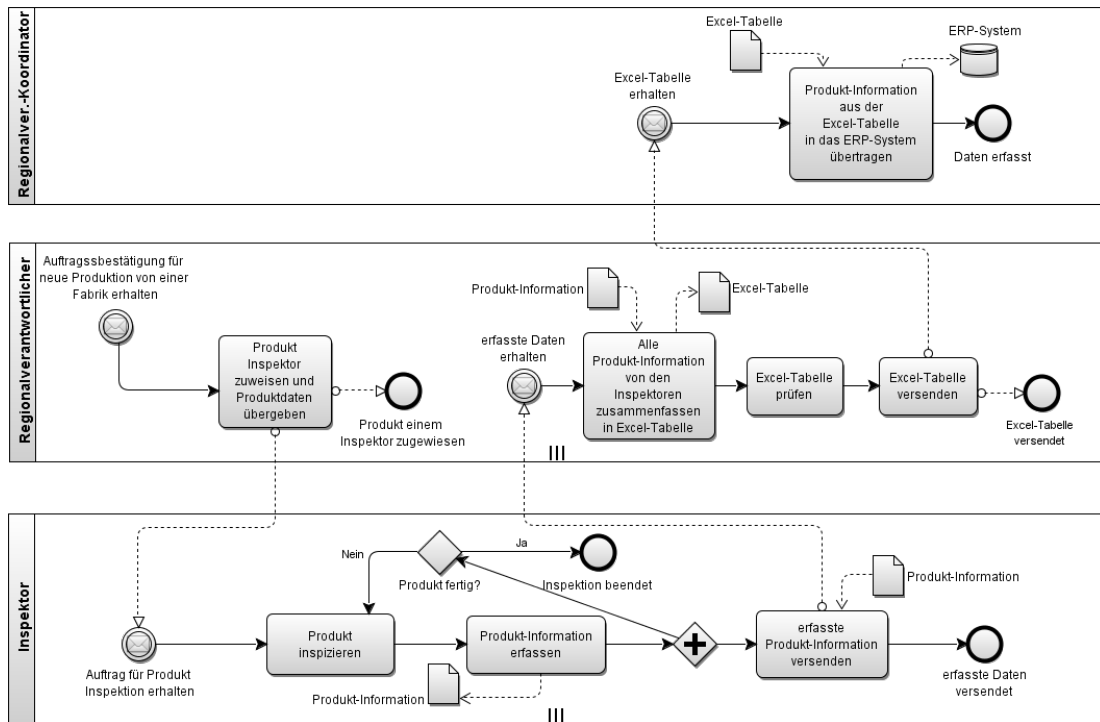


Abbildung 3.2.: IST-Prozessmodell (BPMN-Modell)

In dem Interview wurde geprüft, ob die Erhebung des Prozesses fehlerfrei durchgeführt wurde. Die Schwachstellen und Probleme des Prozesses werden im nächsten Abschnitt analysiert.

¹³nach caBPMN, siehe Abschnitt 2.2.2

¹⁴BPMN 2.0 Sympole(Poster) unter <http://www.bpmb.de/index.php/BPMNPoster>. Abgerufen am 30.09.14

¹⁵Gateway mit "Plus" bedeutet, dass die nachfolgenden Prozessschritte parallel ausgeführt werden.

3.3. Prozessanalyse

Für die Prozessanalyse kann das IST-Prozessmodell von der Ebene 1¹⁶ zur Ebene 2¹⁶ weiter modelliert werden, wenn es für die Zertifizierung oder zur Orientierung im täglichen Ablauf benötigt wird. Zur Analyse und zur Verbesserung des Prozesses reicht die Dokumentation auf Ebene 1¹⁶ aus. Eine Process Engine soll später nicht verwendet werden.

Zur Problemanalyse wurde ein Interview mit der IT-Leitung und dem Regionalverantwortlichen-Koordinator durchgeführt. Dabei wurden drei wichtige Schwachstellen erkannt:

- Der Prozess ist zu intransparent.
Es ist nicht zu erkennen, wie weit der Prozess fortgeschritten ist. Somit ist ein aktueller Stand der Produktüberwachung schwer zu ermitteln. Der jeweilige Produktstand kann nicht erfasst werden.
- Der Prozess ist zu aufwendig.
Es sind zu viele manuelle Tätigkeiten enthalten. Unter anderem ist es zu aufwendig, die erfassten Daten in einer Excel-Tabelle zusammenzufügen und diese anschließend in das ERP-System zu übertragen. In jedem der manuellen Schritte können sich Fehler bei der Datenübernahme einschleichen.
- Die Prozessgeschwindigkeit ist zu langsam bzw. der Prozess dauert zu lange.
Die erfassten Daten werden erst zusammengefasst, somit dauert es, bis Informationen über ein bestimmtes Produkt in Norderstedt vorliegen.

¹⁶nach caBPMN, siehe Abschnitt 2.2.2

3. Analyse

Aus der Analyse der Schwachstellen ergibt sich die Kausalkette der Probleme (Abbildung 3.3). Ausgehend von den drei zentralen Schwächen des Prozesses wurden mehrere Problemstellen identifiziert, die im Rahmen der Arbeit verbessert werden sollen (vgl. Freund und Rücker, 2012, S.141). Darunter:

- Manuelle Tätigkeiten sollen durch Automatisierung und Systemunterstützung verringert werden.
- Der aktuelle Stand zu einzelnen Produkten soll jederzeit einsehbar sein.
- Produktstände sollen nicht mehr zusammengefasst, sondern einzeln bearbeitet werden.

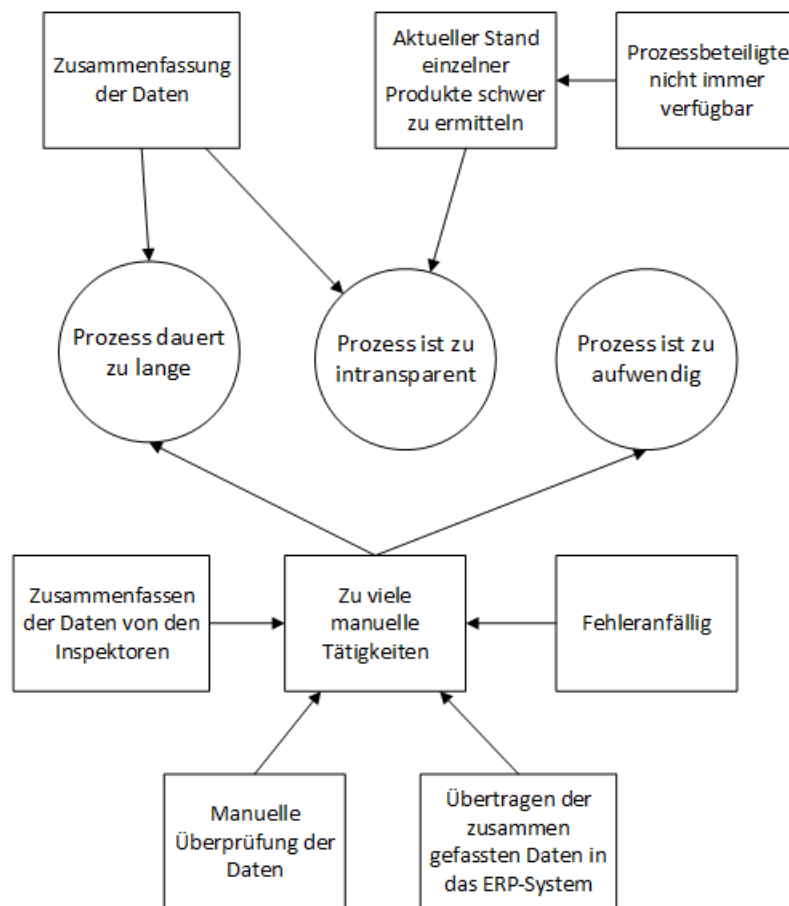


Abbildung 3.3.: Kausalkette zur Darstellung der Prozessschwächen und ihrer Ursachen (nach Freund und Rücker (2012, S.142))

3.4. Prozesskonzeption

Dieser Abschnitt befasst sich mit der Ebene 3b¹⁷. Im vorherigen Abschnitt wurde schon beschrieben, dass keine Process Engine verwendet wird. Deswegen erfolgt die weitere Analyse der Anforderungen und Spezifikationen der Architektur nach [Starke \(2014\)](#) und [Balzert \(2011\)](#).

Zunächst wird eine SOLL-Aufnahme erfasst. In der SOLL-Aufnahme wird der SOLL-Zustand beschrieben. Daraus wird ein SOLL-Konzept erstellt. Danach wird das System analysiert und Anforderungen für die Architektur erstellt. Aus dem Ergebnis entsteht ein SOLL-Prozessmodell.

3.4.1. SOLL-Aufnahme

Neben den Verbesserungsvorschlägen aus der Prozessanalyse wurden bei dem Interview mit der IT-Leitung und dem Regionalverantwortlichen-Koordinator zusätzliche Punkte erfasst, die in dem SOLL-Konzept beschrieben werden sollen.

3.4.1.1. Erfassung des SOLL-Zustandes

Die Optimierungen des Prozesses soll mit möglichst minimalem Aufwand und Kosten gelöst werden. Hierfür wären mehrere Varianten für eine Lösung möglich:

1. Produktüberwachung mit Kameras

In der heutigen Zeit gibt es genügend Techniken, die die Produktion „messen“ und den Stand übermitteln können. Diese Idee wurde aber schnell verworfen, weil die Fabriken nicht der Schmidt Group gehören. Die Kosten und Instandhaltungen wären auch in den Schwellenländer nicht unerheblich.

2. VPN-Client

Die Inspektoren können mit einem VPN-Client auf das ERP-System zugreifen. Eine graphische Oberfläche, um Produktstände zu erfassen, gibt es bereits und könnte somit genutzt werden. Es kann aber nicht sichergestellt werden, dass die Inspektoren in den Fabriken Zugriff auf das Internet haben, somit können sie den Produktstand nicht sofort erfassen. Ein zusätzliches Problem ist die Internetgeschwindigkeit sowie Netzabbrüche, weswegen eine VPN-Verbindung nicht die optimale Lösung darstellt.

¹⁷nach caBPMN, siehe Abschnitt [2.2.2](#)

3. System/Anwendung

Die Inspektoren und Regionalverantwortlichen sollen ein System bzw. eine Anwendung erhalten. Damit sollen die Inspektoren und Regionalverantwortlichen die Produktstände auch offline erfassen können. Die Daten sollen dann, wenn die Inspektoren und Regionalverantwortlichen Internet zu Verfügung haben, an das ERP-System ohne eine VPN-Verbindung übermittelt werden. Das System soll die Funktionalitäten für die Erfassung der Produktstände haben, aber in Zukunft auch um Funktionalitäten erweiterbar sein.

Es wurde die dritte Variante gewählt, weil diese Lösung am ehesten realisierbar erscheint. Das System soll die Bezeichnung *Schwellenland-System* (kurz SL-System) erhalten.

Im nächsten Unterabschnitt wird der Ablauf genauer beschrieben.

3.4.1.2. Ablauf des SOLL-Zustand

Der Regionalverantwortliche bekommt nach wie vor eine Auftragsbestätigung von einer Fabrik, welche mitteilt, dass der Auftrag für eine neue Produktion angenommen wurde und die Produktion beginnen kann. Der Regionalverantwortliche weist einem seiner Inspektoren die Produktionsinspektion zu. Dieser holt sich die Produkt-Information vom ERP-System, anstatt diese vom Regionalverantwortlichen zu erhalten. Die Produkt-Informationen enthalten die Daten, welche in der Abbildung 3.1 (Excel-Tabelle) aufgeführt sind. Hierfür sollen die Produkte jetzt nur noch einzeln erfasst werden und an das ERP-System übermittelt werden, damit der Prozess transparenter wird und der aktuelle Stand einzelner Produktüberwachungen leichter zu ermitteln ist. Die Inspektoren können sich an Orten befinden, an denen kein Netz zur Verfügung steht. Somit sollen die Informationen auch offline erfasst werden können. Bei der Erfassung der Produkt-Informationen soll auf fehlerhafte Eingaben geprüft werden.

Fehlerhafte Eingaben sind zum Beispiel falsche Mengenangaben zu produzierten Einheiten. So ist es unmöglich, dass in einer Woche bei einer Produktionskapazität von 10.000 T-Shirts 100.000 T-Shirts produziert worden sind. Weitere Informationen zu fehlerhaften Eingaben finden sich in Anhang A.2 wieder. Die Erfassung der fehlerhaften Eingaben ist wahrscheinlich nicht vollständig. Es kann Fälle geben, die im Interview nicht beachtet worden sind. Sollten die Fälle im „live“-Betrieb erkannt werden, können diese in der nächsten Problemanalyse besprochen werden. Die Inspektoren können den aktuellen Stand einzelner Produkte übermitteln. Die erfassten Produktstände sollen nicht an das ERP-System, sondern in einen „Sammelbehälter“ übermittelt werden. Dort sollen die Produktstände von dem entsprechenden Regionalverantwortlichen überprüft und dann für das ERP-System freigegeben werden. Wenn es zu keiner Freigabe kommt, muss der Regionalverantwortliche das Problem beheben.

3.4.1.3. SOLL-Konzept

Die Architekturlösung soll mit minimalem Aufwand realisiert werden. Eine Lösung mit den vorhandenen Servern und Systemen ist zu bevorzugen. Ein Interview zum Ermitteln vorhandener System- und Serverkapazitäten wurde mit der IT-Leitung und Administration geführt.

Zur Verfügung stehen:

ERP-System

Das ERP-System enthält die ERP-Software (Apollo) und die Entwicklungsumgebung, um weitere Anwendungen für die ERP-Software zu entwickeln. Im ERP-System befindet sich die C16-Datenbank.

Task-System

Das Task-System steuert automatisierte Prozesse vom ERP-System. Es enthält ein Job-service (Scheduling).

FTP-Server

Der FTP-Server ist öffentlich zugänglich (befindet sich in der **Demilitarisierten Zone** [DMZ]). Dieser dient für den Datentransfer zwischen der Schmidt Group und anderen Firmen.

3. Analyse

Nachfolgend werden die Auswirkungen des SOLL-Konzepts gegenüber dem IST-Modell, mit Bezug auf die jeweiligen Schwachstellen aus der Problemanalyse, zusammengefasst.

- Die Inspektoren erhalten die Produkt-Information über das ERP-System.
- Die Produkte werden einzeln erfasst und nicht mehr zusammengefasst geführt.
Behobene Schwachstelle: Intransparenz und Zeit.
- Die Inspektoren können Produkt-Informationen offline erfassen.
- Beim Erfassen der Produkt-Information wird auf fehlerhafte Eingaben geprüft. Weitere Informationen zu fehlerhaften Eingaben finden sich in Anhang [A.2](#) wieder.
Behobene Schwachstelle: Aufwand.
- Die Inspektoren können den aktuellen Stand der Überwachung jederzeit übermitteln.
Behobene Schwachstelle: Intransparenz und Zeit.
- Die Regionalverantwortlichen prüfen die Produkt-Informationen und fassen diese nicht mehr zusammen.
Behobene Schwachstelle: Aufwand und Zeit.
- Die Produkt-Informationen werden in einem „Sammelbehälter“ zusammengeführt.
- Die Regionalverantwortlichen geben die Produkt-Informationen im „Sammelbehälter“ an das ERP-System frei. Wenn es zu keiner Freigabe kommt, dann muss der Regionalverantwortliche das Problem beheben.
- Der Regionalverantwortlichen-Koordinator muss die Produkt-Informationen nicht mehr an das ERP-System übermitteln.
Behobene Schwachstelle: Aufwand, Intransparenz und Zeit.

3.4.2. Anforderung an die Architektur

Die nachfolgende Abbildung 3.4 zeigt die Kontextsicht des Gesamtsystems.

Um die Verbesserungsvorschläge umzusetzen, wird ein System entwickelt. Das System stellt für die Regionalverantwortlichen und Inspektoren einen Client zur Verfügung, mit dem sie Produkte erhalten, bearbeiten und freigeben können. Die Bezeichnung des Systems ist SL-System. Das SL-System erhält bzw. übermittelt Produkt-Informationen an das ERP-System. Damit das ERP-System Produkt-Informationen an das SL-System sendet, wird das Nachbarsystem Task-System zur Steuerung des ERP-Systems benötigt.

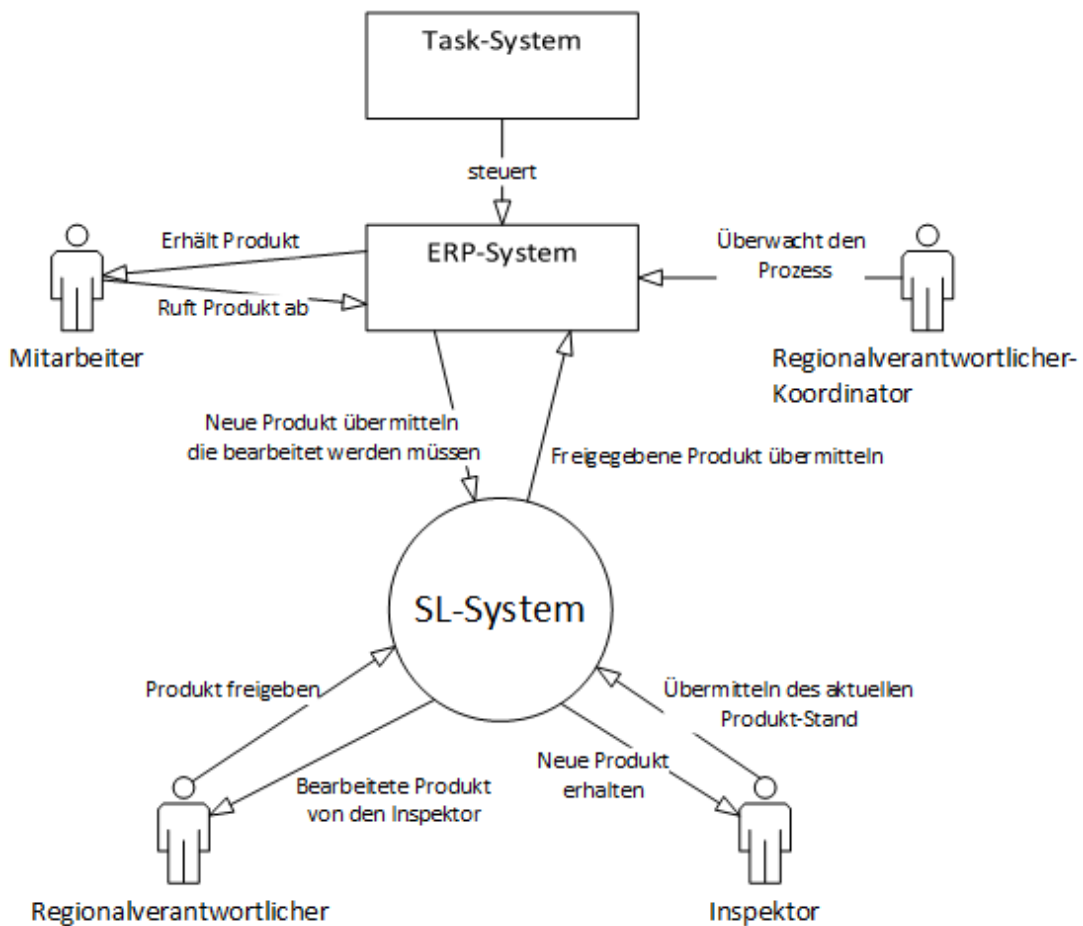


Abbildung 3.4.: Kontextsicht des Gesamtsystems

3. Analyse

Die Mitarbeiter und der Regionalverantwortlichen-Koordinator in Norderstedt greifen direkt auf das ERP-System zu, wenn sie Produkt-Informationen benötigen oder den Produktstand überwachen wollen. Die Inspizierung, Erfassung und Überprüfung der Produkt-Informationen bei den Inspektoren und Regionalverantwortlichen geschieht asynchron, das heißt sie müssen nicht auf eine Antwort warten und können solange weitere Aufgaben erledigen.

Da sich das ERP-System in einem Intranet (geschlossenen Netz) befindet, ist es von außerhalb nicht möglich, direkt auf das ERP-System zuzugreifen und Produkt-Informationen zu übermitteln. Ebenso weiß das ERP-System nicht, wo sich die Regionalverantwortlichen und Inspektoren befinden, um diesen die Produkt-Informationen direkt zu übermitteln. Die Lösung soll mit minimalem Aufwand realisiert werden, deswegen wurde entschieden, dass der FTP-Server als zentrale Stelle für den Datentransfer zwischen ERP-System und SL-System genutzt wird. Als Alternative käme ein Web-Server in Frage, der jedoch zunächst eingerichtet werden müsste. Dadurch würden zusätzliche Kosten entstehen, die vermieden werden sollen. Der FTP-Server kann auch als „Sammelbehälter“ verwendet werden.

3.4.3. SOLL- Prozessmodell

Aus den Ergebnissen des SOLL-Konzepts wird das SOLL-Prozessmodell (Abbildung 3.5¹⁸) erstellt.

Für die Produktionserfassung wird der Regionalverantwortlichen-Koordinator nicht mehr benötigt. Dafür wird das Task-System verwendet, um die manuellen Tätigkeiten zu unterstützen und zu automatisieren. Die Inspektoren erfassen die Produkte im selben „Format“. Somit muss der Regionalverantwortliche diese nicht mehr zusammenfassen und der Prozessaufwand kann gesenkt werden. Die Produkte werden von den Inspektoren einzeln erfasst und an das ERP-System gesendet. Der jeweilige Produktstand ist leichter zu erfassen und der aktuelle Arbeitsschritt leichter zu identifizieren. Damit wird der Prozess transparenter und kann schneller durchlaufen werden.

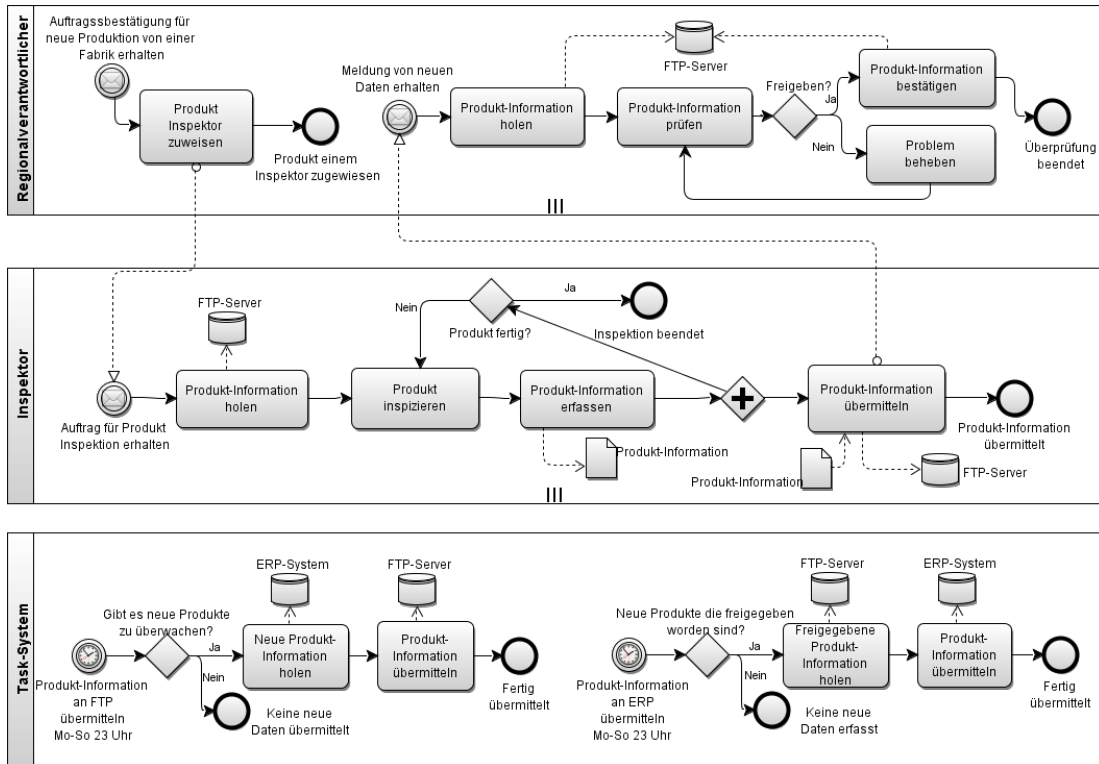


Abbildung 3.5.: SOLL-Prozessmodell (BPMN-Modell)

¹⁸BPMN 2.0 Sympole(Poster) unter <http://www.bpmb.de/index.php/BPMNPoster>. Abgerufen am 30.09.14

3.5. Anforderungen an das SL-System

Das SL-System soll für die Inspektoren und Regionalverantwortlichen erstellt werden, die sich außerhalb des geschlossenen Systems der Schmidt Group befinden. Es soll auf den bereits vorhandenen Rechnern der Mitarbeiter in den Schwellenländer installiert werden. Das SL-System soll die Funktionalitäten für *Produktüberwachung und -erfassung* beinhalten, aber in Zukunft um weitere Funktionalitäten erweiterbar sein. Am Ende wird priorisiert, welche Anforderungen umgesetzt werden sollen und welche nicht.

3.5.1. Funktionale Anforderungen

Funktionale Anforderungen beschreiben, was das System können bzw. leisten muss. Aus dem SOLL-Konzept ergeben sich die folgenden funktionalen Anforderungen (vgl. Balzert, 2011, S.109).

ASL-1 Ein Produkt enthält Produkt-Informationen, die den Informationen, beispielsweise gezeigt in der Excel-Tabelle Abbildung 3.1 (weitere Informationen in Anhang A.3), entsprechen und die bei einer Produktinspizierung erfasst werden müssen. Es soll möglich sein noch weitere Produkt-Informationen erfassen zu können.

ASL-2 Jedes Produkt hat einen Zustand, der beschreibt, in welchem Prozessstand es sich gerade befindet.

- Neu – Produkt, das neu auf dem FTP-Server vom ERP-System erstellt wurde und vom Inspekteur oder dem Regionalverantwortlichen vom FTP-Server geladen worden ist.
- Bearbeitet – Produkt, das vom Inspektor an den FTP-Server übermittelt wird.
- Freigegeben – Produkt, das vom Regionalverantwortlichen explizit auf den Zustand *freigegeben* gesetzt wurde und an den FTP-Server übermittelt worden ist.

ASL-3 Ein Produkt kann nur auf den Zustand *bearbeitet* und *freigegeben* gesetzt werden, wenn die Eingaben valide sind.

ASL-4 Die Produkt-Informationen eines Produktes können sich jederzeit ändern.

3. Analyse

ASL-5 Für die Produktüberwachung und -erfassung soll eine Client-Anwendung entwickelt werden.

ASL-6 Die Anwendung soll eine grafische Benutzeroberfläche (GUI) besitzen.

ASL-7 Die Anwendung soll auch ohne Verbindung bzw. Anmeldung mit dem FTP-Server starten und funktionieren. Somit sollen Produkte offline bearbeitet werden können.

ASL-8 Um ein Produkt auszuwählen, wird zuerst zwischen „lokal vom Rechner“ oder „FTP-Server“ ausgewählt. Anschließend wird eine Auswahlliste erscheinen, in der eine Fabrik ausgewählt werden kann. Daraufhin erscheint eine Auswahlliste, von der ein Produkt ausgewählt werden kann. Es kann nur jeweils ein Produkt zur Zeit bearbeitet und heruntergeladen werden.

ASL-9 Der Inspektor kann Produkte auf den Zustand *bearbeitet* setzen.

ASL-10 Der Regionalverantwortliche kann Produkte auf den Zustand *bearbeitet* und *freigegeben* setzen.

ASL-11 Die Erfassung der Produkt-Informationen soll durch Eingabefelder erfolgen.

ASL-12 Ein Formular beinhaltet Eingabefelder, um bestimmte Produkt-Informationen erfassen zu können. Der Anwender kann nur Formulare auswählen, die für das entsprechende Produkt zur Bearbeitung freigegeben sind.

ASL-13 Die Formulare müssen sich jederzeit ändern lassen können.

ASL-14 Die Eingaben für ein Produkt sollen beim Speichern lokal auf dem Rechner und beim Senden an den FTP-Server auf Validität geprüft werden.

ASL-15 Wenn beim Senden festgestellt wird, dass ein Produkt bereits bearbeitet worden ist, soll der Anwender eine Meldung bzw. Überschreiben-Anfrage erhalten.

ASL-16 Das System soll jeden Zugriff zum FTP-Server protokollieren.

ASL-17 Es soll möglich sein, mehrere Formulare gleichzeitig zu öffnen.

ASL-18 Es können auch mehrere Produkte vom FTP-Server zeitgleich heruntergeladen werden.

3.5.2. Nichtfunktionale Anforderungen

Nichtfunktionale Anforderungen geben Eigenschaften oder Merkmale des zu entwickelnden Systems vor. Sie beinhalten Qualitätsanforderungen an das System (.vgl [Starke, 2014](#), S.39).

Funktionalität

Angemessenheit und Richtigkeit

NFSL-1 Der Anwender muss die Bedeutung der dargestellten Informationsfelder sofort erkennen.

Sicherheit

NFSL-2 Die Dateien dürfen nur verschlüsselt an den FTP-Server gesendet werden.

Zuverlässigkeit

Fehlertoleranz

NFSL-3 Das System soll Fehlermeldungen in ein separates Protokoll schreiben. Dies beinhaltet relevante Informationen, unter anderem die Art des Fehlers und den Namen des Anwenders.

Wiederherstellbarkeit

NFSL-4 Das System soll bei einem Fehlerfall dem Anwender die Optionen geben, entweder die Änderungen rückgängig zu machen, die bei der Aktion geschehen sind (zum Beispiel senden des Produkts) oder die Aktion zu wiederholen.

Benutzbarkeit

Verständlichkeit und Erlernbarkeit

NFSL-5 Die Sprache der Oberflächenelemente ist Englisch. Die Anwendung wird hauptsächlich in den Schwellenländer genutzt.

NFSL-6 Es soll ein Benutzerhandbuch für die Anwendung erstellt werden. Die Sprache des Benutzerhandbuchs ist Englisch.

NFSL-7 Die Anwendung soll auch ohne ein Benutzerhandbuch verständlich sein.

Bedienbarkeit

NFSL-8 Die Zeit vom Start der Anwendung bis zur einsatzbereiten Offline-Datenerfassung soll weniger als eine Minute betragen.

Effizienz

Verbrauchsverhalten

NFSL-9 Der Datentransfer soll asynchron geschehen, damit der Anwender parallel zum Herunterladen der Produkte weiterarbeiten kann.

NFSL-10 Die Anwendung soll unnötige Internetverbindungen vermeiden, damit keine Kosten durch zusätzlichen Datentransfer entstehen. Die Verbindung zum FTP-Server soll nach fünf Minuten geschlossen werden, wenn in der Zeit kein Datentransfer erfolgt ist.

Änderbarkeit

Analysierbarkeit und Modifizierbarkeit

NFSL-11 Für einen Entwickler soll es möglich sein innerhalb eines Tages das Konzept des Systems zu verstehen, um Fehler zu korrigieren und weitere Funktionalitäten implementieren zu können.

NFSL-12 Es soll eine strukturierte Architektur sowie ordentlicher und kommentierter Code eingehalten werden.

Übertragbarkeit

Installierbarkeit

NFSL-13 Es soll eine Installation für die Benutzer zur Verfügung stehen, die sämtliche benötigte Software automatisiert installiert.

NFSL-14 Bei der Installation soll es eine erstmalige Konfiguration geben, bei der das Programm installiert und eine Desktop Verknüpfung erstellt wird.

NFSL-15 Die gesamte Installation darf höchstens 30 Minuten dauern.

Technische Vorgaben

NFSL-16 Der Client soll unter den Betriebssystemen Windows 7 und 8 funktionsfähig / lauffähig sein.

NFSL-17 Es soll das Microsoft .NET Framework in der Version 4.5 eingesetzt werden.

NFSL-18 Als Implementierungssprache soll C# verwendet werden.

NFSL-19 Für die GUI wird das Windows Presentation Foundation Framework (WPF) benutzt.

NFSL-20 Als zusätzliches Softwareentwicklungstool soll DevExpress genutzt werden.

3.5.3. Priorisierung

Aus den Anforderungen erfolgt nun eine Priorisierung der Aufgaben, die in dieser Arbeit erledigt werden müssen und welche nicht. Mit der MoSCoW-Priorisierung wird eine Möglichkeit gegeben, um die Anforderungen eines Projektes zu priorisieren. MoSCoW steht für¹⁹:

Must-have (M) - Ist für das System von zentraler Bedeutung und muss umgesetzt werden.

Should-have (S) - Ist nützlich für das System und soll nach den Must-haves erfüllt werden.

Could-have (C) - Ist „nice to have“ für das System und kann umgesetzt werden, wenn Zeit und Budget reichen.

Won't-have (W) - Wird nicht umgesetzt, ist aber als Idee vorgemerkt für die Zukunft. Kann somit später noch umgesetzt werden.

Priorisierung der funktionalen Anforderungen

ASL-	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
M	X	X	X	X	X	X	X	X	.	X	X	X	X
S	X	X	X	X	.	.
C	X	.
W	X

Priorisierung der nichtfunktionalen Anforderungen

NFSL-	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
M	X	.	X	.	X	.	.	X	X	.	.	X	.	.	.
S	.	.	.	X	.	.	X	.	.	X	X	c	X	X	X
C
W	.	X	.	.	.	X

Die technischen Vorgaben sind Pflichtenanforderungen.

¹⁹<http://www.seo-book.de/projektmanagement/priorisierung-fur-seo-projekt-e-mit-der-moscow-methode>. Abgerufen am 30.09.14

3.6. Anforderungen an das ERP-System

Das ERP-System muss um die Funktionalität *Produktüberwachung und -erfassung* erweitert werden. Die Funktionen des ERP-System werden durch das Nachbarsystem Task-System gesteuert.

Die Anforderungen an das ERP-System werden in dieser Arbeit nicht umgesetzt. Es soll zuerst das SL-System entwickelt werden. Nach der Evaluierung des SL-Systems wird entschieden, ob das SL-System genutzt werden soll, weil ein neues ERP-System geplant ist. Es sollen aber Anforderungen und Designentscheidungen für das jetzige ERP-System beschrieben werden, falls die Anforderungen für das alte ERP-System umgesetzt wird.

3.6.1. Funktionale Anforderung

Funktionale Anforderungen beschreiben, was das System können bzw. leisten muss. Aus dem SOLL-Konzept ergeben sich die folgenden funktionalen Anforderungen (.vgl [Balzert, 2011](#), S.109).

ASL-1 Ein Produkt enthält Produkt-Informationen, die den Informationen, beispielsweise gezeigt in der Excel-Tabelle Abbildung [3.1](#) (weitere Informationen in Anhang [A.3](#)), entsprechen und die bei einer Produktinspizierung erfasst werden müssen. Es soll möglich sein noch weitere Produkt-Informationen erfassen zu können.

AERP-2 Jedes Produkt hat einen Zustand, der beschreibt, in welchem Prozessstand es sich gerade befindet.

- Neu – Produkt, das noch nie an den FTP-Server übermittelt worden ist.
- Gesendet – Produkt, das vom Inspektor an den FTP-Server übermittelt wurde.
- Freigegeben – Produkt, das vom Regionalverantwortlichen explizit auf den Zustand *freigegeben* gesetzt worden ist. Die Produkte werden geladen und an das ERP-System übermittelt.
- Fertig – Produkt, das ausgeliefert ist und nicht mehr an den FTP-Server übermittelt werden muss.

3. Analyse

- NeuSenden – Produkt, welches vom FTP-Server geladen wurde und nicht den Zustand *fertig* erhalten hat. Somit wird das Produkt weiter inspiziert und im Zustand *neuSenden* wieder an FTP-Server übermittelt.

AERP-3 Ein Produkt kann nur auf den Zustand *fertig* und *freigegeben* gesetzt werden, wenn die Eingaben valide sind.

AERP-4 Die Produkt-Information eines Produkt können sich jederzeit ändern.

AERP-5 Das ERP-System soll einmal täglich prüfen, welche Produkte in der Datenbank den Zustand *neu* und *neuSenden* haben.

AERP-6 Das ERP-System soll Produkte, die den Zustand *neu* und *neuSenden* haben, an den FTP-Server übermitteln.

AERP-7 Das ERP-System soll einmal täglich die Produkte auf dem FTP-Server prüfen, ob es neue Produkte gibt, die in den Zustand *freigegeben* gesetzt wurden.

AERP-8 Das ERP-System soll die Produkte, die freigegeben wurden, auswerten und in die Datenbank einfügen. Produkt-Informationen, die freigegeben sind, werden daraufhin auf dem FTP-Server gelöscht.

AERP-9 Das ERP-System soll beim Auswerten prüfen, ob das Produkt geliefert worden ist. Produkte, die geliefert worden sind, erhalten den Zustand *fertig*.

AERP-10 Das ERP-System soll beim Übermitteln des Produkt den letzten Stand mit übergeben.

3.6.2. Nichtfunktionale Anforderungen

Nichtfunktionale Anforderungen geben Eigenschaften oder Merkmale des zu entwickelnden Systems vor. Sie beinhalten Qualitätsanforderungen an das System (.vgl [Starke, 2014](#), S.39).

Funktionalität

Sicherheit

NFERP-1 Die Produkte dürfen nur verschlüsselt an den FTP-Server gesendet werden.

3.7. Vergleichbare Software und Lösungen

In diesem Abschnitt werden vergleichbare Lösungen, die auf dem Markt sind, analysiert. Das Gesamtsystem und dessen Anforderungen wurde speziell für die Schmidt Group konzipiert, deswegen muss geprüft werden, wie aufwendig die Anpassung der vorhandenen Lösungsansätze wäre.

3.7.1. MiniSell

Von der Firma aspin²⁰ gibt es eine kostenpflichtige Softwarelösung zur Bearbeitung von Kunden- und Produktinformationen. Die Lösung ist für Smartphones, Tablets und Laptops, also für Mitarbeiter, die sich nicht im Firmennetz befinden und unterwegs sind. Dafür installiert die Firma aspin einen Databridge-Server²¹ außerhalb des Intranets (wie FTP-Server), auf den die Geräte über FTP zugreifen können. Die Daten, die transferiert werden können, sind zum Beispiel CSV- oder XML-Dateien. Diese Lösungsarchitektur entspricht den technischen Anforderungen, besitzt aber darüber hinaus deutlich mehr Funktionalitäten als benötigt werden. Zusätzlich müssten trotzdem die Dialoge angepasst werden und dementsprechend auch die Logik.

3.7.2. Fazit

Bis auf MiniSell erfüllt keine erhältliche Software die gestellten Anforderungen (deswegen werden auch keine weiteren beschrieben). Die System- Anforderungen benötigen nur einen kleinen Teil von dem, was das Produkt MiniSell anbietet. Es ist zu vermuten, dass Aufwand und Kosten um eine vorhandene Lösung anzupassen, hoch sind. Der Vorteil einer eigenen Software ist, dass sie bei Bedarf um weitere Funktionalitäten erweitert werden kann (was in Zukunft auch geplant ist). Dies soll nicht heißen, dass eine vorhandene Lösung schlecht sei, jedoch sollte der Aufwand den Nutzen nicht übersteigen. Im schlechtesten Fall wird eine teure Software gekauft, die nicht genutzt wird. Deswegen ist es besser, eine Software zu entwickeln, die genau die gestellten Anforderungen erfüllt. Sollte sie genutzt werden, kann sie weiter entwickelt werden.

²⁰<http://www.aspininteractive.com/minisell>. Abgerufen am 30.09.14

²¹Ist ein von aspin selbst entwickelter Server

4. Spezifikation

In diesem Kapitel wird das Gesamtsystem, basierend auf den Anforderungen, spezifiziert. Zur Spezifikation werden des Weiteren ein fachliches Datenmodell, Anwendungsfälle und Dialoge definiert, um den Aufbau des Gesamtsystems näher zu erläutern.

Die Anwendungssprache ist Englisch, in dieser Arbeit werden jedoch deutsche Begriffe verwendet. Im Anhang [A.4](#) befindet sich eine Liste der deutschen Begriffe und ihrer verwendeten englischen Äquivalente.

4.1. Glossar

Nachfolgend werden die im Anwendungsbereich verwendeten Begriffe definiert:

Produkt - Ein Produkt enthält Produkt-Informationen, die bei einer Produktinspektion erfasst werden müssen. Das Produkt enthält auch einen Zustand, damit bei der Überwachung der Stand des Produktes bekannt ist.

Produkt-Information - Produkt-Informationen sind Bestandteil von Produkten. Ein Beispiel für Informationen sind Produktname und Fabrik. Produktinformationen können sich jederzeit ändern.

Formular - Ein Formular beinhaltet Eingabefelder. Um Produkt-Informationen erfassen zu können, muss ein Formular ausgewählt werden. Für ein Produkt können mehrere Formulare ausgewählt und somit verschiedene Produkt-Informationen erfasst werden.

4.2. Fachliches Datenmodell

Das fachliche Datenmodell, welches in der Abbildung 4.1 zu sehen ist, ist für das zu entwickelnde SL-System. Es gibt nur die Klasse: *Produkt*. Die Klasse *Produkt* beinhaltet, neben dem Attribut *Zustand*, noch die Produkt-Informationen, die bei einer Inspizierung zu erfassen sind.

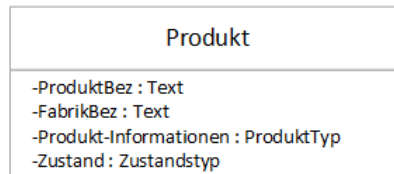


Abbildung 4.1.: Fachliches Datenmodell für das SL-System

Der Datentyp *ProduktTyp* beinhaltet die Informationen des Produktes. Der *ProduktTyp* muss dynamisch anpassbar sein.

Der Datentyp *ZustandsTyp* beschreibt den aktuellen Zustand des Produktes. Der *ZustandsTyp* besteht aus *neu*, *bearbeitet* und *freigegeben* und wird im kommenden Unterabschnitt 4.2.1 weiter beschrieben.

Das ERP-System besitzt schon eine Klasse für das Produkt. Diese wird um das Attribut *Zustand* erweitert, damit im ERP-System auch der Zustand, in dem sich das Produkt befindet, erfasst werden kann.

4.2.1. Zustandsgraph für das SL-System

In diesem Abschnitt wird der Zustandsgraph für das Produkt im SL-System beschrieben (Abbildung 4.2). Ein Produkt kann sich im Zustand *neu*, *bearbeitet* und *freigegeben* befinden.

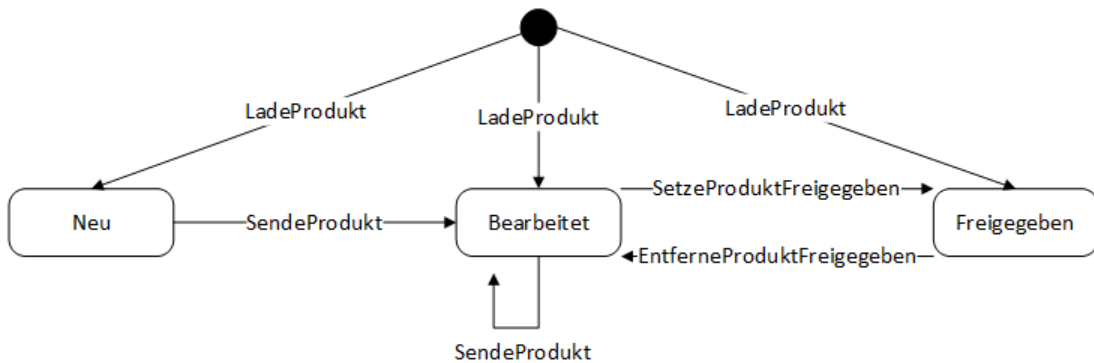


Abbildung 4.2.: Produkt-Zustände im SL-System

Wenn ein Produkt vom ERP-System generiert und auf dem FTP-Server hinzugefügt wird, hat es den Zustand *neu*. Der Inspektor oder der Regionalverantwortliche können ein Produkt mit den Zuständen *neu*, *bearbeitet* und *freigegeben* laden. Wenn der Zustand *neu* ist, ändert er sich mit dem ersten Senden an den FTP in den Zustand *bearbeitet*. Ist der Zustand *bearbeitet*, dann bleibt er das bis der Regionalverantwortliche das Produkt überprüft und freigegeben hat. Mit der Freigabe wechselt der Zustand in *freigegeben* und das ERP-System kann sich das Produkt holen. Der Regionalverantwortliche kann auch den Zustand *freigegeben* entfernen, dann ist der Zustand wieder *bearbeitet*.

4.2.2. Zustandsgraph für das ERP-System

In diesem Abschnitt wird der Zustandsgraph für das Produkt im ERP-System beschrieben (Abbildung 4.3). Ein Produkt kann sich im Zustand *neu*, *gesendet*, *freigegeben*, *neuSenden* und *fertig* befinden.

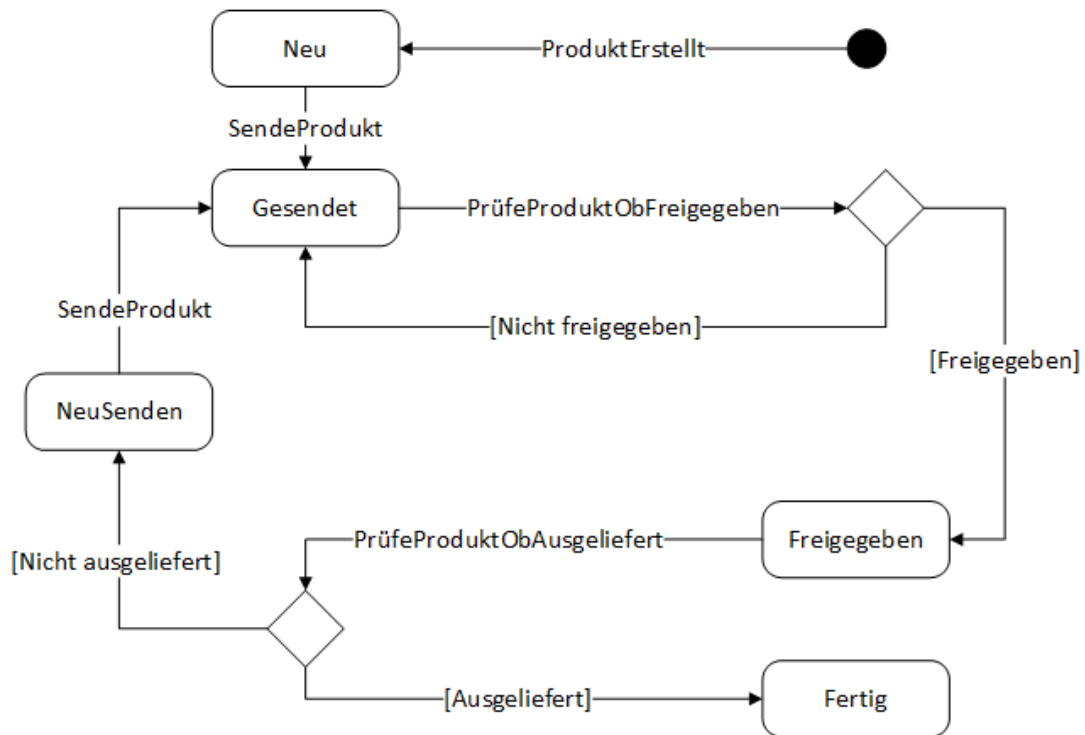


Abbildung 4.3.: Produkt-Zustände im ERP-System

Sobald ein Produkt in die Datenbank eingetragen wird, hat es den Zustand *neu*. Wenn das Produkt an den FTP-Server gesendet worden ist, hat es den Zustand *gesendet*. Es wird dann geprüft, ob sich der Zustand auf *freigegeben* geändert hat. Sobald es den Zustand *freigegeben* hat, wird das Produkt vom FTP-Server heruntergeladen. Dann wird geprüft, ob die Auslieferung erfolgt ist. Wenn das Produkt ausgeliefert worden ist, erhält es den Zustand *fertig* und es muss nicht mehr inspiziert werden. Wenn es nicht ausgeliefert worden ist, dann erhält es den Zustand *neuSenden*. Das ERP-System sendet das Produkt mit den neuen Informationen an den FTP-Server und es erhält wieder den Zustand *gesendet*. Das Produkt wird solange inspiziert, bis es ausgeliefert ist und sich im Zustand *fertig* befindet.

Sollte sich der Zustand in *gesendet* befinden und die Produkt-Informationen haben sich geändert, dann muss der Zustand manuell geändert bzw. das Senden des Produktes an FTP-Server neu angestoßen werden.

4.3. Anwendungsfälle

Die folgenden Anwendungsfälle (Use Cases) beschreiben das Verhalten der Anwendung. Zuerst werden die Anwendungsfälle des SL-Systems im Abschnitt 4.3.1 beschrieben, dann die des ERP-Systems in Abschnitt 4.3.2.

4.3.1. SL-System

Dieser Unterabschnitt beschreibt die wichtigsten Anwendungsfälle des SL-Systems. Die beteiligten Akteure für das System sind die Inspektoren und Regionalverantwortlichen. In diesen Anwendungsfällen wird die Aktualisierung und Auswahl von Formularen beschrieben. Ebenso die Auswahl, Erfassung und Übermittlung von Produkten sowie das Hinzufügen und Entfernen der Freigabe. Am Ende dieses Abschnitts werden die User Stories beschrieben.

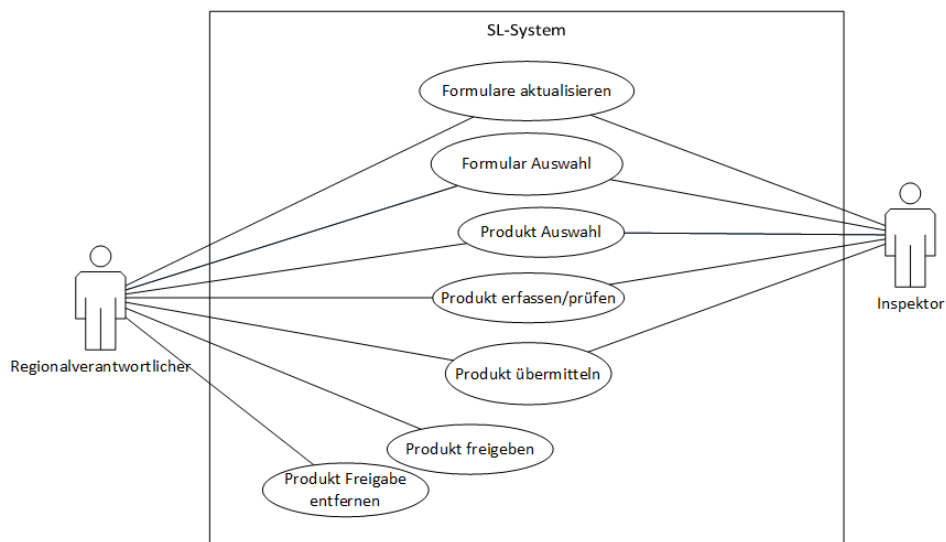


Abbildung 4.4.: Use Cases des SL-Systems

4. Spezifikation

4.3.1.1. Formulare aktualisieren

Dieser Anwendungsfall beschreibt, wie die Akteure die Formulare aktualisieren bzw. neue Formulare erhalten.

Titel	Formulare aktualisieren
Akteur	Inspektor, Regionalverantwortlicher
Ziel	Das System aktualisiert die Formulare.
Auslöser	Die Formulare sollen aktualisiert werden.
Vorbedingungen	Die Anwendung ist gestartet und der Akteur hat sich mit dem FTP-Server verbunden.
Nachbedingungen	Die Formulare sind auf dem aktuellen Stand.
Erfolgsszenario	<ol style="list-style-type: none">1. Der Akteur wählt „Formulare aktualisieren“ oder es wurde das erste Mal am Tag eine Verbindung zum FTP-Server geöffnet.2. Das System prüft, ob die Formulare auf dem Rechner auf demselben Stand sind, wie auf dem FTP-Server.3. Das System startet die Aktualisierung.4. Der Akteur erhält eine Meldung, dass die Aktualisierung der Formulare abgeschlossen worden ist.5. Das System erzeugt ein Protokoll auf dem FTP-Server.
Erweiterungen	-
Fehlerfälle	3.a. Fehler beim Herunterladen der Formulare. System gibt eine Meldung aus und fordert zur nochmaligen Eingabe auf.
Häufigkeit	Mindestens einmal, wenn eine Verbindung zum FTP-Server geöffnet wird.
Anforderungen	ASL-13, ASL-16

Tabelle 4.1.: Anwendungsfall „Formulare aktualisieren“

4.3.1.2. Formular Auswahl

Der Anwendungsfall beschreibt die Auswahl eines Formulars für ein Produkt.

Titel	Formular Auswahl
Akteur	Inspektor, Regionalverantwortlicher
Ziel	Der Akteur wählt ein Formular aus.
Auslöser	Der Akteur entscheidet sich ein (anderes) Formular auszuwählen.
Vorbedingungen	Die Anwendung ist gestartet und der Akteur hat ein Produkt geöffnet.
Nachbedingungen	Der Akteur hat das Formular geöffnet.
Erfolgsszenario	
<ol style="list-style-type: none"> 1. Der Akteur wählt „Formular Auswahl“ aus. 2. Das System prüft, welche Formulare für das Produkt ausgewählt werden können. 3. Der Akteur erhält eine Liste von Formularen, die ausgewählt werden können. 4. Der Akteur wählt ein Formular aus der Liste. 5. Das System gibt das Formular auf dem GUI aus. 	
Erweiterungen	
-	
Fehlerfälle	
<ol style="list-style-type: none"> 2.a Das System findet keine Formulare. <ol style="list-style-type: none"> 2.a.1 Der Akteur muss sich mit dem FTP-Server verbinden. Ausführung des Anwendungsfalls „FTP-Server verbinden“. 2.a.2 Ausführung des Anwendungsfalls „Formulare aktualisieren“. 	
Häufigkeit	Mindestens einmal wöchentlich
Anforderungen	ASL-11, ASL-12, ASL-17

Tabelle 4.2.: Anwendungsfall „Formulare Auswahl“

4. Spezifikation

4.3.1.3. Produkt Auswahl

Der Anwendungsfall beschreibt die Auswahl von Produkten auf dem FTP-Server. Die Auswahl lokal vom Rechner ist analog zum FTP-Server, aber ohne eine FTP- Verbindung.

Titel	Produkt Auswahl
Akteur	Inspektor, Regionalverantwortlicher
Ziel	Der Akteur holt sich Produkte vom FTP-Server (bzw. lokal) .
Auslöser	Der Akteur entscheidet sich ein Produkt vom FTP-Server (bzw. lokal) zu laden.
Vorbedingungen	Die Anwendung ist gestartet (und der Akteur hat sich mit dem FTP-Server verbunden).
Nachbedingungen	Der Akteur ist im Besitz des/der Produkt(s/e).
Erfolgsszenario	
<ol style="list-style-type: none"> 1. Der Akteur wählt „Produkt Auswahl“ und wählt zwischen <i>neu</i>, <i>bearbeitet</i> und <i>freigegeben</i> aus . 2. Das System holt sich die aktuelle Liste der Fabriken, die zur Auswahl stehen. 3. Der Akteur sieht eine Liste von Fabriken und wählt eine Fabrik aus. 4. Das System holt sich die aktuelle Liste der Produkte. 5. Der Akteur sieht eine Liste von Produkten und wählt ein oder mehrere Produkte von der entsprechenden Fabrik aus. 6. (Betrifft FTP-Server) Das System lädt die Produkte vom FTP-Server herunter. 7. (Betrifft FTP-Server) Der Akteur erhält eine Meldung, dass das Herunterladen der Produkte abgeschlossen ist. 8. (Betrifft FTP-Server) Das System erzeugt ein Protokoll auf dem FTP-Server. 9. Das System öffnet das Produkt. 	
Erweiterungen	
Fehlerfälle (Betrifft FTP-Server)	
<ol style="list-style-type: none"> 2.a.1 Fehler beim Herunterladen der Liste. System gibt eine Meldung aus und fordert zur nochmaligen Eingabe auf. 4.a.1 Fehler beim Herunterladen der Liste. System gibt eine Meldung aus und fordert zur nochmaligen Eingabe auf. 6.a.1 Fehler beim Herunterladen der Produkte. System gibt eine Meldung aus und fordert zur nochmaligen Eingabe auf. Nur fehlerhafte Datentransfers werden erneut ausgeführt. 	
Häufigkeit	Mindestens einmal wöchentlich.
Anforderungen	ASL-2, ASL-8, ASL-16, ASL-18

Tabelle 4.3.: Anwendungsfall „Produkt Auswahl“

4. Spezifikation

4.3.1.4. Produkt erfassen/prüfen

Der Anwendungsfall beschreibt das Bearbeiten bzw. Prüfen von einem Produkt.

Titel	Produkt erfassen/prüfen
Akteur	Inspektor, Regionalverantwortlicher
Ziel	Der Akteur hat ein Produkt bearbeitet bzw. geprüft.
Auslöser	Der Inspektor entscheidet sich, ein Produkt zu bearbeiten bzw. zu prüfen.
Vorbedingungen	Die Anwendung ist gestartet und der Akteur hat ein Produkt geöffnet.
Nachbedingungen	Die Eingaben des Produkts sind gültig.
Erfolgsszenario	
1. Ausführung des Anwendungsfalls „Formular Auswahl“. 2. Der Akteur bearbeitet ein Eingabefeld. 3. Das System prüft anhand der Kriterien, ob die Eingabe gültig ist. 4. Das System speichert das Produkt.	
Erweiterungen	
-	
Fehlerfälle	
3.a.1 Falls die Eingabe ungültig ist, fordert das System den Akteur zur Korrektur auf.	
Häufigkeit	Mindestens einmal wöchentlich.
Anforderungen	ASL-2, ASL-14

Tabelle 4.4.: Anwendungsfall „Produkt erfassen/prüfen“

4.3.1.5. Produkt übermitteln

Der Anwendungsfall beschreibt das Übermitteln eines Produktes zum FTP-Server.

Titel	Produkt übermitteln
Akteur	Inspektor, Regionalverantwortlicher
Ziel	Der Akteur sendet das Produkt zum FTP-Server.
Auslöser	Der Akteur entscheidet sich, das Produkt zu übermitteln.
Vorbedingungen	Die Anwendung ist gestartet, der Akteur hat ein Produkt geöffnet und hat sich mit dem FTP-Server verbunden.
Nachbedingungen	Das Produkt befindet sich auf dem FTP-Server und ist im Zustand <i>bearbeitet</i> .
Erfolgsszenario	
<ol style="list-style-type: none"> 1. Der Akteur wählt „Produkt übermitteln“. 2. Das System prüft, ob die eingegebenen Informationen valide sind. 3. Das System prüft auf dem FTP-Server, ob das Produkt den Zustand <i>neu</i> hat. Wenn ja, dann setzt das System den Zustand auf <i>bearbeitet</i>. 4. Das System sendet das Produkt an den FTP-Server. 5. Das System löscht das Produkt lokal auf dem Rechner. 6. Der Akteur erhält eine Meldung, dass das Senden des Produkts an den FTP-Server abgeschlossen worden ist. 7. Das System erzeugt ein Protokoll auf dem FTP-Server. 	
Erweiterungen	
-	
Fehlerfälle	
<ol style="list-style-type: none"> 2.a.1 Falls eine Eingabe ungültig ist, fordert das System den Akteur zur Korrektur auf. 4.a.1 Falls das Produkt von einem anderen Akteur bearbeitet worden ist, fragt das System den Akteur, ob die Datei überschrieben werden soll. 4.b.1 Fehler beim Senden des Produkts. System gibt eine Meldung aus und fordert zur nochmaligen Eingabe auf. 	
Häufigkeit	Mindestens einmal wöchentlich.
Anforderungen	ASL-2, ASL-9, ASL-10, ASL-14, ASL-15, ASL-16

Tabelle 4.5.: Anwendungsfall „Produkt übermitteln“

4. Spezifikation

4.3.1.6. Produkt freigeben

Der Anwendungsfall beschreibt das Freigeben eines Produkts an das ERP-System.

Titel	Produkt freigeben
Akteur	Regionalverantwortlicher
Ziel	Der Akteur setzt den Zustand auf <i>freigegeben</i> .
Auslöser	Der Akteur entscheidet sich das Produkt an das ERP-System freizugeben.
Vorbedingungen	Die Anwendung ist gestartet und der Akteur hat ein Produkt geöffnet, das in dem Zustand <i>bearbeitet</i> ist. Der Akteur hat sich mit dem FTP-Server verbunden.
Nachbedingungen	das Produkt ist im Zustand <i>freigegeben</i> .
Erfolgsszenario	
<ol style="list-style-type: none"> 1. Der Akteur wählt „Produkt freigeben“. 2. Das System prüft, ob die eingegebenen Informationen valide sind. 3. Das System sendet das Produkt mit Zustand <i>freigegeben</i> an den FTP-Server. 4. Das System löscht das Produkt lokal auf dem Rechner. 5. Der Akteur erhält eine Meldung, dass das Senden des Produkts an den FTP-Server abgeschlossen worden ist. 6. Das System erzeugt ein Protokoll auf dem FTP-Server. 	
Erweiterungen	
-	
Fehlerfälle	
<ol style="list-style-type: none"> 2.a.1 Falls eine Eingabe ungültig ist, fordert das System den Akteur zur Korrektur auf. 3.a.1 Fehler beim Senden des Produkts. System gibt eine Meldung aus und fordert zur nochmaligen Eingabe auf. 3.b.1 Falls das Produkt von einem anderen Akteur bearbeitet worden ist, fragt das System den Akteur, ob die Datei überschrieben werden soll. 	
Häufigkeit	Mindestens einmal wöchentlich.
Anforderungen	ASL-2, ASL-10, ASL-14, ASL-15, ASL-16

Tabelle 4.6.: Anwendungsfall „Produkt freigeben“

4.3.1.7. Produkt Freigabe entfernen

Der Anwendungsfall beschreibt das Entfernen der Freigaben eines Produkts.

Titel	Produkt Freigabe entfernen
Akteur	Regionalverantwortlicher
Ziel	Der Akteur setzt den Zustand auf <i>bearbeitet</i> .
Auslöser	Der Akteur entscheidet sich, die Freigabe des Produkts zu entfernen.
Vorbedingungen	Die Anwendung ist gestartet und der Akteur hat ein Produkt geöffnet, das in dem Zustand <i>freigegeben</i> ist. Der Akteur hat sich mit dem FTP-Server verbunden.
Nachbedingungen	Das Produkt ist im Zustand <i>bearbeitet</i> .
Erfolgsszenario	
<ol style="list-style-type: none"> 1. Der Akteur wählt „Produkt Freigabe entfernen“. 2. Das System prüft, ob die eingegebenen Informationen valide sind. 3. Das System sendet das Produkt mit dem Zustand <i>bearbeitet</i> an den FTP-Server. 4. Der Akteur erhält eine Meldung, dass das Senden des Produkts an den FTP-Server abgeschlossen worden ist. 5. Das System erzeugt ein Protokoll auf dem FTP-Server. 	
Erweiterungen	
-	
Fehlerfälle	
<ol style="list-style-type: none"> 2.a.1 Falls eine Eingabe ungültig ist, fordert das System den Akteur zur Korrektur auf. 3.a.1 Fehler beim Senden des Produkts. System gibt eine Meldung aus und fordert zur nochmaligen Eingabe auf. 3.b.1 Falls das Produkt von einem anderen Akteur bearbeitet worden ist, fragt das System den Akteur, ob die Datei überschrieben werden soll. 	
Häufigkeit	Mindestens einmal wöchentlich.
Anforderungen	ASL-2, ASL-10, ASL-14, ASL-15, ASL-16

Tabelle 4.7.: Anwendungsfall „Produkt freigabe entfernen“

4.3.1.8. User Stories

Diese User Stories beschreiben die Systemfunktionalitäten für das System.

FTP-Server Login - Der Akteur kann eine Verbindung zum FTP-Server öffnen.

FTP-Server Disconnect - Der Akteur kann eine bestehende Verbindung zum FTP-Server schließen.

4.3.2. ERP-System

Dieser Unterabschnitt beschreibt die Anwendungsfälle des ERP-Systems. Die beteiligten Akteure für das ERP-System sind die Mitarbeiter, der Regionalverantwortlichen-Koordinator und das Task-System. In diesen Anwendungsfällen wird die Übermittlung der Produkte zum FTP bzw. ERP beschrieben, um die Produkte auswerten zu können.

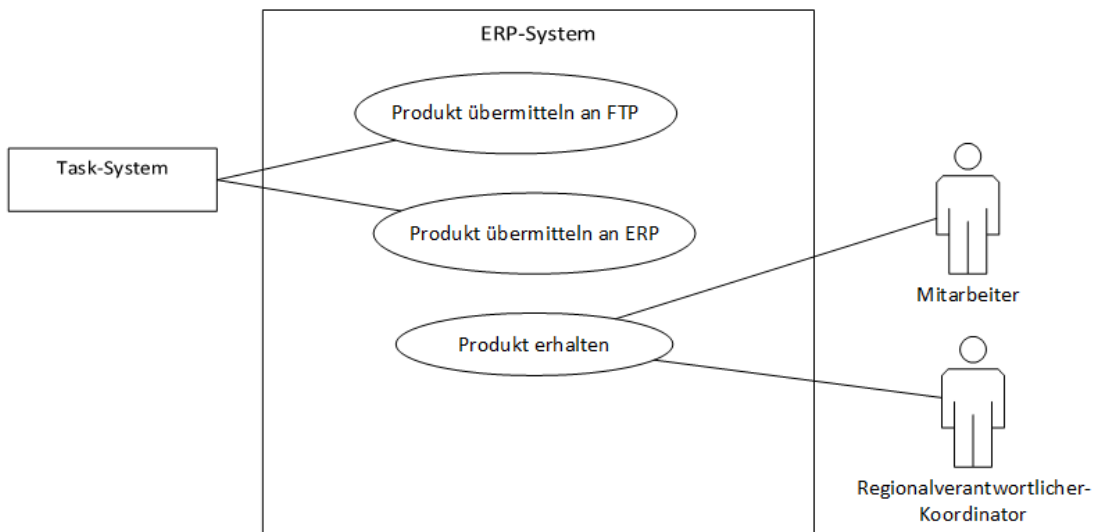


Abbildung 4.5.: Use Cases des ERP-Systems

4.3.2.1. Produkt übermitteln an FTP

Der Anwendungsfall beschreibt die Übermittlung der Produkte an den FTP-Server.

Titel	Produkt übermitteln an FTP
Akteur	Task-System
Ziel	Produkte an den FTP-Server übermittelt.
Auslöser	Automatische Auslösung: einmal täglich.
Vorbedingungen	Das Task-System muss Zugriff auf das ERP-Systems haben.
Nachbedingungen	Das Task-System hat die Produkte an den FTP-Server übermittelt.
Erfolgsszenario	
<ol style="list-style-type: none"> 1. Das Task-System startet und ruft im ERP-System die Funktion „Produkt übermitteln an FTP“ auf. 2. Das ERP-System prüft in der Datenbank, ob es Produkte mit dem Zustand <i>neu</i> oder <i>neuSenden</i> gibt. 3. Das ERP-System öffnet eine Verbindung zum FTP-Server. 4. Das ERP-System sendet die Produkte zum FTP-Server. 5. Das ERP-System schließt die Verbindung zum FTP-Server. 6. Das ERP-System setzt die entsprechende Produkte auf Zustand <i>gesendet</i>. 7. Das ERP-System erzeugt ein Protokoll. 	
Erweiterungen	
-	
Fehlerfälle	
<ol style="list-style-type: none"> 3.a.1 Es konnte keine Verbindung zum FTP-Server hergestellt werden. Das ERP-System versucht es dreimal erneut und bricht anschließend ab. Das ERP-System erzeugt ein Fehlerprotokoll. 4.a.1 Fehler beim Senden des Produkts. Das ERP-System versucht es dreimal erneut und bricht anschließend ab. Das ERP-System erzeugt ein Fehlerprotokoll. 	
Häufigkeit	Einmal täglich um 23 Uhr.
Anforderungen	AERP-2, AERP-5, AERP-6, AERP-10

Tabelle 4.8.: Anwendungsfall „Produkt übermitteln an FTP“

4.3.2.2. Produkt übermitteln an ERP

Der Anwendungsfall beschreibt die Übermittlung der Produkte an das ERP-System.

Titel	Produkt übermitteln an ERP
Akteur	Task-System
Ziel	Produkte an das ERP-System übermittelt.
Auslöser	Automatische Auslösung: einmal täglich oder manuelle Auslösung.
Vorbedingungen	Das Task-System muss Zugriff auf das ERP-System haben.
Nachbedingungen	Das Task-System hat das/die Produkt(e) an das ERP-System übermittelt.
Erfolgsszenario	
<p>1. Das Task-System startet und ruft im ERP-System die Funktion „Produkt übermitteln an ERP“ auf.</p> <p>2. Das ERP-System öffnet eine Verbindung zum FTP-Server.</p> <p>3. Das ERP-System prüft beim FTP-Server, ob es Produkte mit dem Zustand <i>freigegeben</i> gibt.</p> <p>4. Das ERP-System lädt die Produkte vom FTP-Server und erfasst diese in der Datenbank des ERP-Systems.</p> <p>5. Das ERP-System schließt die Verbindung zum FTP-Server.</p> <p>6. Das ERP-System prüft die Produkte, ob diese den Zustand <i>fertig</i> oder <i>neuSenden</i> erhalten und erfasst diese in der Datenbank des ERP-Systems.</p> <p>7. Das ERP-System erzeugt ein Protokoll.</p>	
Erweiterungen	
-	
Fehlerfälle	
<p>2.a.1 Es konnte keine Verbindung zum FTP-Server hergestellt werden. Das ERP-System versucht es dreimal erneut und bricht anschließend ab. Das ERP-System erzeugt ein Fehlerprotokoll.</p> <p>4.a.1 Fehler beim Laden des Produkts. Das ERP-System versucht es dreimal erneut und bricht anschließend ab. Das ERP-System erzeugt ein Fehlerprotokoll.</p>	
Häufigkeit	Einmal täglich um 23 Uhr.
Anforderungen	AERP-2, AERP-3, AERP-7, AERP-8, AERP-9

Tabelle 4.9.: Anwendungsfall „Produkt übermitteln an ERP“

4. Spezifikation

4.3.2.3. Produkt erhalten

Der Anwendungsfall beschreibt die Auswahl eines Produktes, um es zu erhalten.

Titel	Produkt erhalten
Akteur	Mitarbeiter, Regionalverantwortlicher-Koordinator
Ziel	Der Akteur holt sich ein Produkt vom ERP-Server.
Auslöser	Der Akteur entscheidet sich, ein Produkt zu holen.
Vorbedingungen	Der Akteur hat Zugriff auf das ERP-System.
Nachbedingungen	Der Akteur hat das Produkt.
Erfolgsszenario	1. Der Akteur ruft im ERP-System die Graphische Oberfläche „Produkt Überwachung“ auf.
Erweiterungen	-
Fehlerfälle	-
Häufigkeit	Mindestens einmal wöchentlich
Anforderungen	-

Tabelle 4.10.: Anwendungsfall „Produkt erhalten“

4.4. Dialoge

In diesem Abschnitt werden die Benutzeroberflächen beschrieben. Die **Mockups** sollen als Richtlinie für die Benutzeroberflächen dienen. Der Ribbon soll die Tabs *File*, *FTP* und *Options* beinhalten. In den nachfolgenden Unterabschnitten werden die Oberflächen beschrieben.

Die Mockups sind eine grobe Skizze, das heißt, die Produkt-Informationen sind nicht enthalten. Diese sind aber weitere Eingabefelder.

4.4.1. File

Beim Start der Anwendung befindet sich die Oberfläche auf dem Tab *File*. In diesem Tab kann der Anwender die Produkte lokal auf seinem Rechner öffnen (dabei wird der Dialog *Produkt Auswahl* geöffnet), speichern und löschen. Ebenso kann der Anwender die Formulare auswählen, die er bearbeiten will. Im unteren Teil befindet sich das ausgewählte Formular, um die Produkt-Informationen zu erfassen (diese sind aus Platzgründen weggelassen worden).

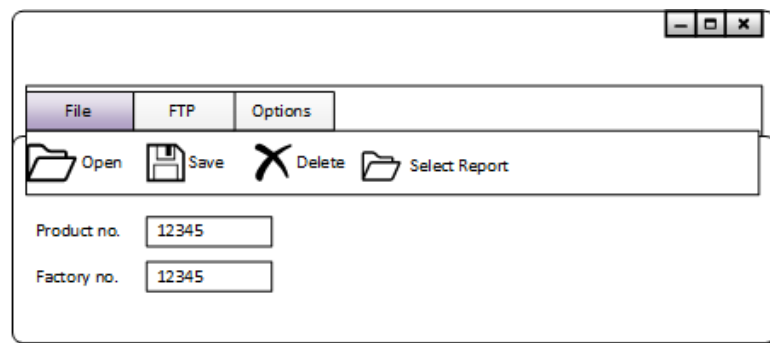


Abbildung 4.6.: GUI Mockup File

4.4.2. FTP

Im Tab *FTP* können neue Produkte geladen (dabei wird der Dialog *Produkt Auswahl* geöffnet), das geöffnete Produkt übermittelt (dabei wird der Zustand auf *bearbeitet* gesetzt), das geöffnete Produkt freigegeben bzw. die Freigabe entfernt und die Formulare aktualisiert werden. Um die Buttons zu nutzen, muss sich der Anwender vorher mit dem FTP verbinden (Login). Der Anwender kann auch manuell die Verbindung beenden (Disconnect).

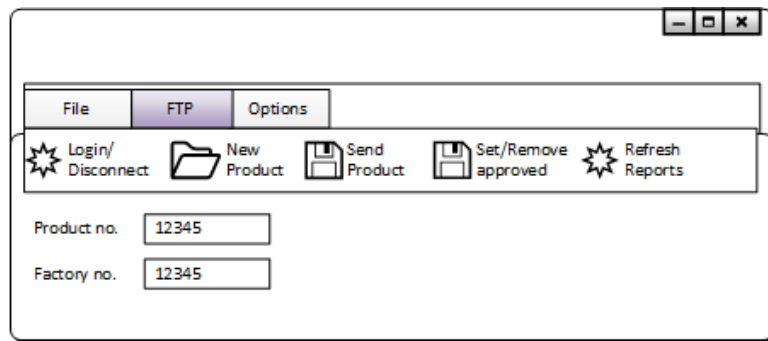


Abbildung 4.7.: GUI Mockup FTP

4.4.3. Options

Im Tab *Options* hat der Anwender die Möglichkeit, die Anwendung zu konfigurieren. Dieser Tab ist als Platzhalter gedacht, falls später Konfigurationsmöglichkeiten implementiert werden sollen.

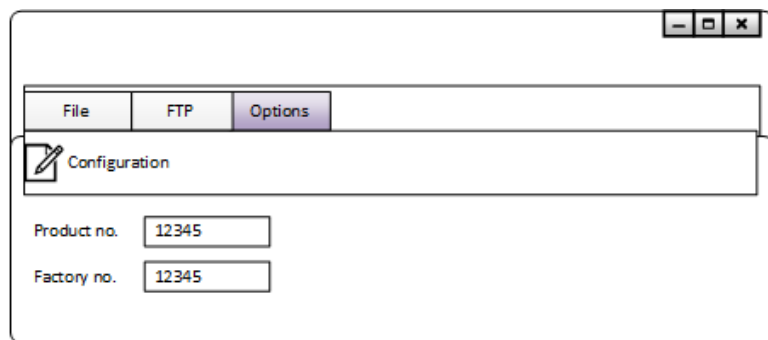


Abbildung 4.8.: GUI Mockup Options

4.4.4. Produkt Auswahl

Wenn der Anwender ein Produkt öffnen bzw. vom FTP-Server herunterladen will, dann öffnet sich dieser Dialog. Der Anwender kann auswählen, welchen Zustand er öffnen bzw. herunterladen will. Er wählt dann eine Fabrik und dazu ein Produkt aus.



Abbildung 4.9.: GUI Mockup Produkt Auswahl

4.4.5. Formular Auswahl

Wenn der Anwender ein Formular auswählen will, dann öffnet sich dieser Dialog. Er wählt dann ein Formular aus.

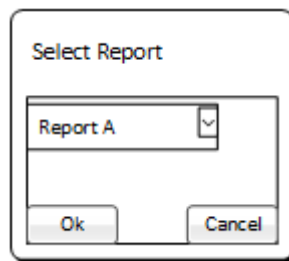


Abbildung 4.10.: GUI Mockup Formular Auswahl

5. Design und Realisierung

In diesem Kapitel wird ein Konzept für die Realisierung des Gesamtsystems und dessen Umsetzung beschrieben. Zum Konzept des Gesamtsystems tragen die Analyse und die Spezifikation bei. Zunächst werden die allgemeinen Designentscheidungen beschrieben, die relevant für das SL-System (Client) und das ERP-System (Server) sind. Anschließend werden die Designentscheidungen für den Server und für den Client beschrieben. Zusätzlich wird die Realisierung des Clients aufgeführt.

5.1. Allgemeine Designentscheidungen

In diesem Abschnitt werden die allgemeinen Designentscheidungen beschrieben, die relevant für den Client und den Server sind. Zuerst wird die Verteilungssicht beschrieben, dann die Client-Server Architektur. Dazu muss entschieden werden, welches Synchronisationsverfahren angewendet wird, um das Problem der Nebenläufigkeit zu vermeiden. Für die Zustände des Produktes wird eine Ordnerstruktur vorgestellt, um die Zustände der Produkte zu erkennen. Anschließend wird das Datenaustauschformat erklärt. Zuletzt wird die Sicherheitsproblematik beschrieben, unter anderem, welche Probleme das File Transfer Protokoll birgt, die unter anderem durch die Verwendung des FTP-Protokolls entsteht.

5.1.1. Verteilungssicht

In der Abbildung 5.1 werden die Anbindungen der Server und der Clients sichtbar. Auf dem Schmidt Group-Server befindet sich das ERP-System und das Task-System. Das SL-System (Client) hat keine direkte Verbindung zum ERP-System (Server). Beide Systeme haben eine Verbindung zum DMZ-Server, auf dem sich der FTP-Server befindet. Damit wird ein Datentransfer zwischen den beiden Systemen ermöglicht.

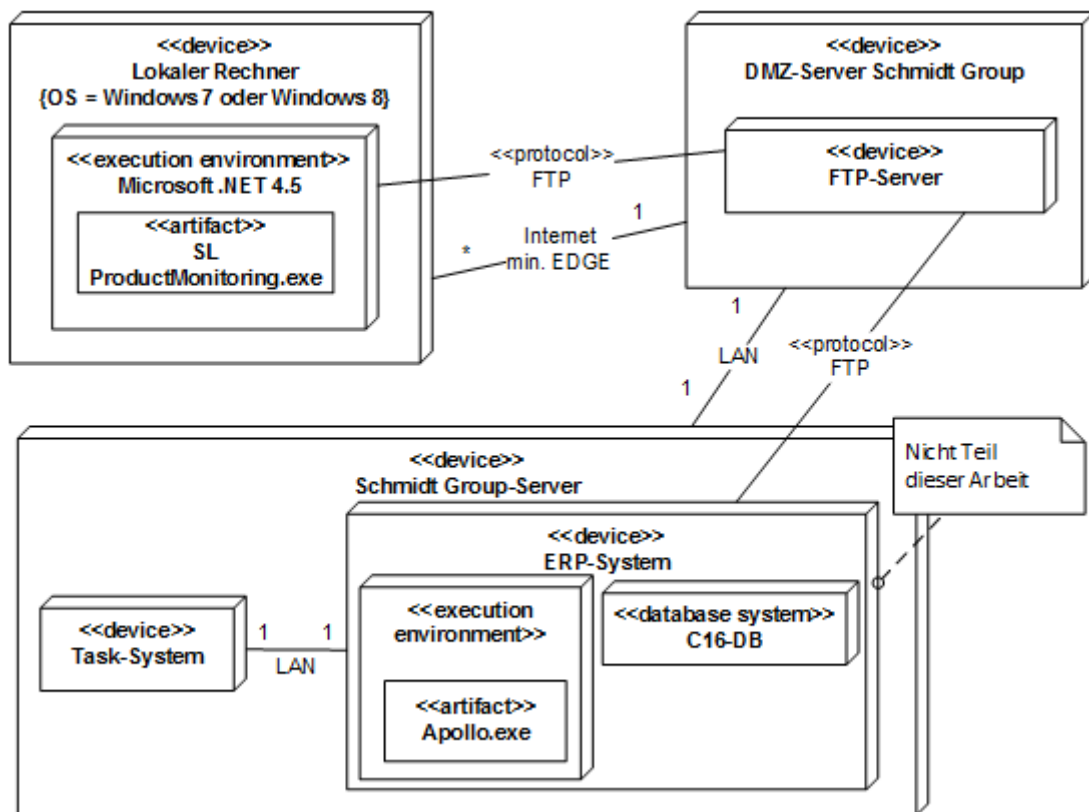


Abbildung 5.1.: Verteilungsdiagramm

Je nachdem, was die beide Systeme benötigen, laden sie die Daten auf den FTP-Server hoch oder von ihm herunter. Die Mitarbeiter, die den Client nutzen, benötigen einen Benutzeraccount, um auf den FTP-Server Zugriff zu erhalten. Deshalb muss für jeden Mitarbeiter ein Benutzeraccount eingerichtet werden. Die Kommunikation der beiden Systeme zum FTP-Server wird über das File Transfer Protokoll (FTP) ermöglicht. Die Daten werden beim Client auf den Datenträger (zum Beispiel Festplatte) gespeichert. Das Task-System enthält ein Jobservice der Funktionen im ERP-System aufruft.

5.1.2. Client-Server Architektur

Für das Gesamtsystem wird eine Client-Server Architektur entwickelt. Das SL-System entspricht einem Client und das ERP-System dem Server. Die Abbildung 5.2 zeigt die möglichen Aufteilungen einer verteilten Anwendung nach Tanenbaum und van Steen (2007).

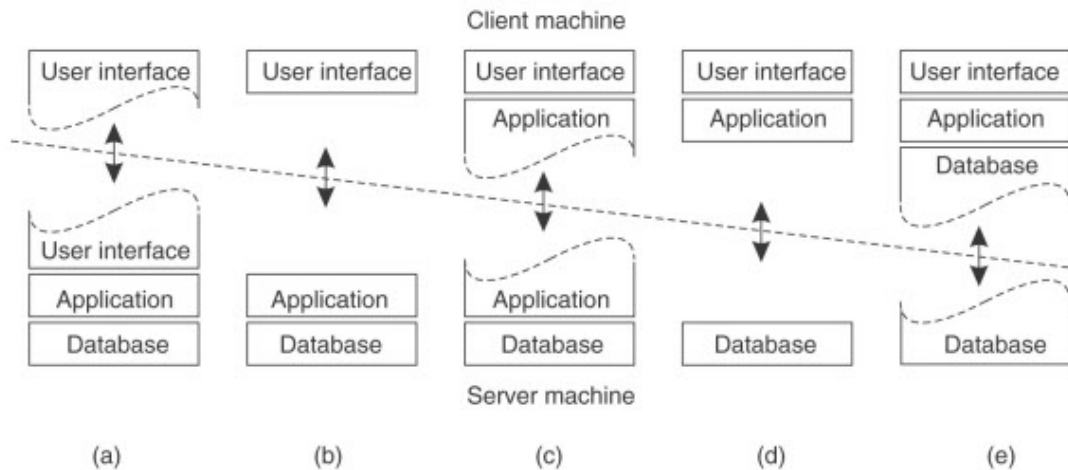


Abbildung 5.2.: Client-Server Architektur (Tanenbaum und van Steen, 2007)

Der Server stellt die Daten für den Client zur Verfügung. Der Client kann nicht direkt auf den Server zugreifen, deswegen muss der Server die Daten für den Client vorbereiten und diese auf den FTP-Server hochladen. Dies wird im Abschnitt 5.2.2 beschrieben. Der Server muss die bearbeiteten Daten vom FTP-Server herunterladen und diese der Datenbank hinzufügen. Die Überprüfungen, ob die Daten korrekt sind, geschieht auf der Client Seite. Der Client besitzt somit auch Logik, um die Daten bearbeiten zu können. Die Client-Anwendung muss offline funktionieren, deswegen werden die Daten auf den Datenträger des Clients gespeichert. Sobald der Client Internetzugriff hat bzw. online ist, muss der Client die Daten mit dem FTP-Server synchronisieren. Weil es mehrere Clients gibt und es passieren kann, dass mehrere das selbe Produkt (zeitgleich) inspizieren, muss auf die Nebenläufigkeit geachtet werden. Dies wird im Abschnitt 5.1.3 genau beschrieben.

Die Architektur für den Client wird gemäß Variante e umgesetzt, weil die Anwendung offline funktionieren muss und deshalb eine eigene Database benötigt wird.

5.1.3. Nebenläufigkeit

Wie in Abbildung 5.1 zu erkennen ist, hat jedes System seinen eigenen Datenträger, das ERP-System seine eigene Datenbank. Hierbei kann es zu Problemen in der Konsistenzprüfung führen, wenn ein Inspektor oder Regionalverantwortlicher die Daten ändert. Falls ein anderer Inspektor oder Regionalverantwortlicher das selbe Produkt bearbeitet, entstünde eine Dateninkonsistenz. Ebenso kann es geschehen, dass ein Inspektor ein Produkt bearbeitet und das ERP-System die Produktinformationen, die es bereitstellt, geändert hat und diese an den FTP-Server gesendet hat. Die Wahrscheinlichkeit, dass ein Konflikt zwischen Inspektoren entstehen kann, ist gering, weil Produkte meistens jeweils nur einem Inspektor zugewiesen werden. Der Regionalverantwortliche schaut sich auch nur bearbeitete Produkte an. Hier könnte es aber schon passieren, dass ein Regionalverantwortlicher die Produkte freigibt, bei denen ein Inspektor den Produktstand nachträglich geändert hat. Falls eine Änderung vom ERP-System kommen sollte, wird dies von einem Mitarbeiter der Schmidt Group manuell ausgelöst. Für diesen Fall wurde entschieden, dass der Inspektor auf diese Änderungen hingewiesen wird (zum Beispiel durch eine Email).

Für die Synchronisation wird das pessimistische und das optimistische Verfahren betrachtet, um zu bewerten, welches dieser Verfahren besser geeignet ist.

5.1.3.1. Pessimistische Synchronisation

Das pessimistische Synchronisationsverfahren zielt darauf ab, dass die Konflikte bereits während der Bearbeitung vermieden werden. Bei der pessimistische Synchronisation wird ein Write-Lock (Schreibsperre) auf die Daten gesetzt, die bearbeitet werden. Deshalb kann verhindert werden, dass eine andere Person oder ein anderes System die Daten bearbeitet (Coulouris, Dollimore und Kindberg, 2002, vgl.). Dieses Verfahren scheitert aber schon daran, dass die Daten offline bearbeitet werden können. Es ist somit nicht möglich, ein Write-Lock zu setzen. Deswegen wird das pessimistische Synchronisationsverfahren nicht weiter betrachtet.

5.1.3.2. Optimistische Synchronisation

Das optimistische Synchronisationsverfahren erkennt einen Konflikt erst bei der Ausführung. Dieses Verfahren teilt sich in drei Phasen (siehe Abbildung 5.3²²): Lesephase (Arbeitsphase), Validationsphase und Schreibphase. In der Lesephase werden die Daten auf eine privaten Kopie gelesen und geschrieben. Bei der Validationsphase wird geprüft, ob die Daten sich geändert haben, also persistiert werden können. Zuletzt wird bei der Schreibphase geprüft, ob die Validierung positiv oder negativ ist. Bei positiver Validierung werden die Daten überschrieben bzw. geändert (Coulouris, Dollimore und Kindberg, 2002, vgl.).

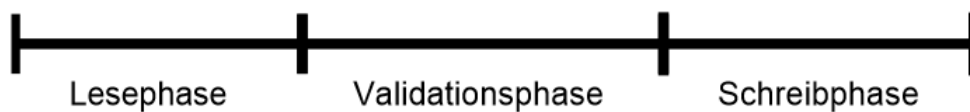


Abbildung 5.3.: Drei Phasen der optimistische Synchronisationsverfahren

Das optimistische Synchronisationsverfahren ist die bessere Wahl, weil die Daten offline erfasst werden und erst beim Übertragen geprüft werden kann, ob sich die Daten geändert haben. Dabei wird sich am Änderungsdatum orientiert. Das heißt, wenn sich das Änderungsdatum der Datei auf dem FTP-Server geändert hat, nachdem die Datei heruntergeladen wurde, hat ein zweite Person/System die Datei bearbeitet. Sollte sich die Datei geändert haben, muss der Konflikt entsprechend geregelt werden.

Es wurde in einem Gespräch mit der IT-Leitung über zwei mögliche Varianten zur Lösung dieses Konfliktes gesprochen. Die möglichen Lösungen orientieren sich an dem EntityFramework Concurrency Pattern von Microsoft²³.

Client-Wins Dieses Verfahren bedeutet, dass die letzte Datei, die speichert, gewinnt (der letzte Speichervorgang die vorherige Datei überschreibt). Wenn also eine Person speichert und dann eine andere Person ebenso speichert, wird die Datei ohne eine Meldung überschrieben.

²² von http://www.informatik.uni-ulm.de/dbis/papers/vdb-buch/vdb99_08.pdf. Abgerufen am 30.09.14

²³ <http://msdn.microsoft.com/en-us/data/jj592904.aspx>. Abgerufen am 30.09.14

Store-Wins Dieses Verfahren bedeutet, dass die Person, die speichert, eine Meldung bekommt, wenn die Datei vorher von einer anderen Person geändert worden ist. In diesem Fall gibt es mehrere Lösungsansätze. Der gängigste Ansatz ist das Überschreiben oder das Zusammenführen der Daten.

Es wurde die Store-Wins-Variante gewählt, weil eine Überschreibung ohne Meldung nicht sinnvoll ist. Das Zusammenführen der Daten wäre mit dem Microsoft Sync Framework²⁴ möglich. Das Sync Framework kann nicht direkt Ordner auf einem FTP-Server synchronisieren und Daten zusammenführen. Entweder werden die Daten heruntergeladen und geprüft oder es muss auf dem FTP-Server ein Service geschrieben werden, der dies ermöglicht. Deswegen wurde entschieden, dass der Mitarbeiter eine Meldung erhält, ob er die Datei überschreiben möchte oder nicht. Ein Konflikt wird wahrscheinlich nur selten entstehen, weil die Produkte jeweils nur einem Inspektoren zugewiesen werden und diese sich in Kontakt mit den Regionalverantwortlichen befinden. Deshalb wird in Kauf genommen, dass der Mitarbeiter Daten möglicherweise unnötig bearbeitet hat bzw. mit zusätzlichem Aufwand erneuert bearbeiten muss.

Die Realisierung des SL-System wird im Abschnitt 5.3.5.1 beschrieben.

5.1.4. Ordnerstruktur

Für die Ordner im Datenträger des Clients und des FTP-Servers muss eine Struktur festgelegt werden, die den Zustand eines Produktes widerspiegelt. Die Anwendung wird in Englisch gehalten sein, aber in dieser Arbeit werden weiterhin die deutschen Begriffe verwendet. Im Anhang A.4 befindet sich eine Übersicht über die Übersetzung der Begriffe.

Variante 1

Zuerst wird in drei Ordner aufgeteilt: In Dialog, Produktstand und Protokoll (siehe Abbildung 5.4). Der Ordner *Dialog* ist für die Formulare. Dieser beinhaltet die aktuellen Formulare als XAML-Dateien. Die alten Formulare sollen sich nicht mehr auf dem FTP-Server befinden. Der Ordner *Protokoll* beinhaltet die Protokolle, die vom Client erzeugt werden. Dieser wird sich nur auf dem FTP-Server befinden. Dann gibt es den Ordner *Produktstand*. Er beinhaltet die Produkte, die inspiziert und überwacht werden müssen. Die Aufteilung erfolgt nach Fabriken, anschließend nach den Produkten. Die Ordner für die Produkte beinhalten die Dateien, in denen

²⁴<http://msdn.microsoft.com/de-de/sync>. Abgerufen am 30.09.14

die Produkt-Informationen enthalten sind. Die Produkte befinden sich in eigene Ordnern, falls die Produkt-Informationen in Zukunft in mehr als eine Datei aufgeteilt werden oder weitere Produkt-Informationen, wie zum Beispiel Bilder, in zusätzlichen Dateien gespeichert werden. Um herauszufinden, in welchem Zustand sich das Produkt befindet, muss entweder in den Produkt-Informationen der Zustand stehen oder in einer zusätzlichen Datei.

Der Vorteil dieser Variante ist, dass sichergestellt wird, dass es ein Produkt nur einmal gibt (es also in *Produktstand* nur einmal existiert). Der Nachteil ist, dass, wenn zum Beispiel alle Produkte im Zustand *freigegeben* geprüft werden sollen, für jedes Produkt in allen Fabriken geprüft werden muss, in welchem Zustand es sich befindet.

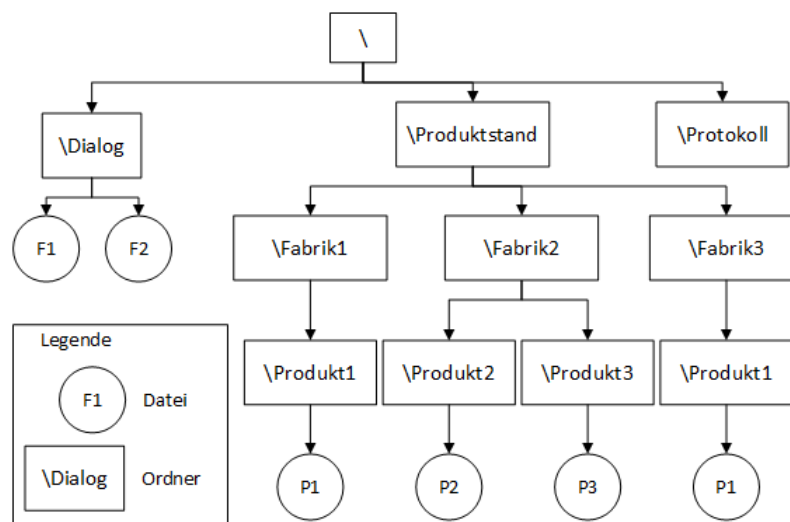


Abbildung 5.4.: Ordnerstruktur für FTP-Server Variante 1

Variante 2

Wie bei Variante 1 wird am Anfang in drei Ordner aufgeteilt: In Dialog, Produktstand und Protokoll (siehe Abbildung 5.5). Bis auf die Ordnerstufe nach *Produktstand*, entspricht Variante 2 der Variante 1. In *Produktstand* erfolgt die Aufteilung zunächst in die Zustände *neu*, *bearbeitet* und *freigegeben*. Nach den Zuständen kommen die Ordner für Fabriken und anschließend die für Produkte. Ein Produkt darf sich jeweils nur in einem der Ordner für die Zustände befinden. Der Vorteil dieser Variante ist, dass besser zu erkennen ist, in welchem Zustand sich welche Produkte befinden. Wenn beispielsweise nur nach Produkten im Zustand *freigegeben* gesucht wird, dann erhält der Anwender nur die Fabriken, in denen sich dieses Produkt im Zustand *freigegeben* befindet. Der Nachteil ist, dass es passieren kann, dass ein Produkt zweimal existiert, zum Beispiel im Zustand *neu* und im Zustand *bearbeitet*. Es muss also sichergestellt werden, dass sich ein Produkt nicht zweimal im Ordner *Produktstand* befindet.

Eine Überprüfung aller Produkte, um herauszufinden in welchem Zustand sich das jeweilige Produkt befindet, wie es bei Variante 1 nötig wäre, kann zeitaufwendig sein. Bei mehreren hundert Produkten ist dies nicht sinnvoll, wenn jede Datei überprüft werden müsste, ob sie freigegeben ist. Deswegen fiel die Entscheidung auf Variante 2.

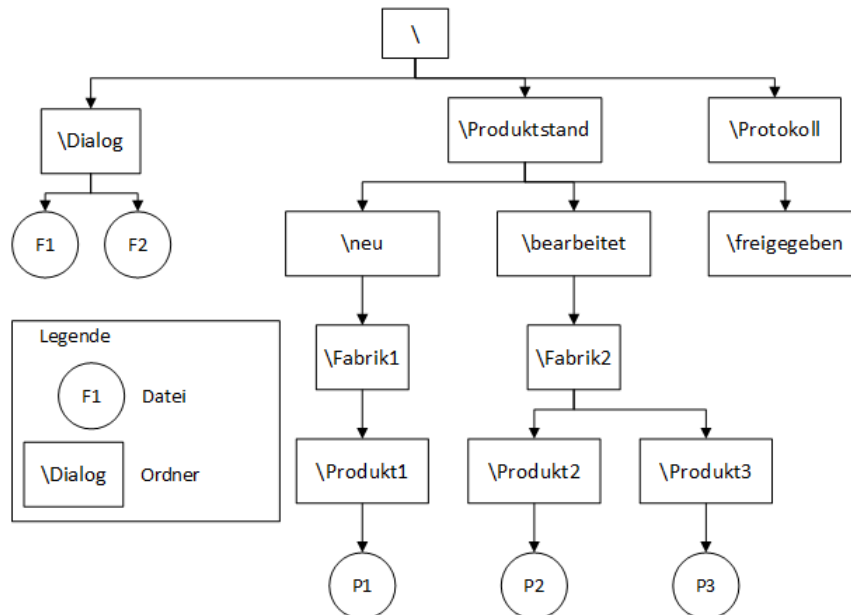


Abbildung 5.5.: Ordnerstruktur für FTP-Server Variante 2

5.1.5. Datenaustauschformat

Ein Produkt wird als eine Datei gespeichert. Deswegen muss sich für ein Datenaustauschformat entschieden werden, in dem die Produkte gespeichert und an den FTP-Server übertragen werden können. Hierfür wurde XML und **JavaScript Object Notation** (JSON) in Bezug auf die möglichen Nutzungsmöglichkeiten in WPF verglichen.

Der Vorteil bei XML ist, dass die Daten in einer XML-Datei als eine „Dateninsel“ verwendet werden können. Durch **XPath** kann im XAML-Code ein XML-Element oder ein -Attribut ausgewählt und angezeigt werden. Deswegen kann für eine XML-Datei ein oder mehrere XAML-Code (Views) erstellt werden, die verschiedene Elemente der XML-Datei darstellen. Die Daten von der XML-Datei können eingelesen, angezeigt, geändert und anschließend wieder

direkt in die XML-Datei gespeichert werden²⁵. Wenn zum Beispiel ein Eingabefeld (ein neues Informationsfeld, das erfasst werden muss) zur XAML-Datei hinzugefügt wird, dann muss die neue Information nur noch in der XML-Datei ergänzt werden. Damit muss keine Logik angepasst werden (vgl. Wegener, 2013, S.269-272).

Diesen Vorteil gibt es bei JSON in Vergleich zum XML nicht. Bei JSON müsste die Model-Klasse angepasst werden, wenn ein Informationsfeld hinzugefügt wird. Hierfür wäre es zum Beispiel möglich, die Model-Klasse zur Laufzeit zu erstellen oder XML in JSON umzuwandeln und an den FTP-Server zu übertragen. Die XML-Datei wird nur den letzten Stand (Anforderung an das ERP-System) und den aktuellen Stand beinhalten, deswegen wird die Dateigröße klein sein. Bei Tests bezüglich der Dateigröße, wenn nur der aktuelle und der letzte Stand enthalten sind, beträgt die Größe der XML-Datei 2KB. Wenn die letzten neun Stände und der aktuelle Stand in der XML-Datei enthalten sind, dann ist sie 12-14KB groß. Hier gibt es keinen wesentlichen Unterschied der Dateigröße zwischen XML und JSON. Dieser entstünde erst, wenn eine deutlich größere Menge an Ständen gespeichert werden würde. Es erfolgt aber nur eine Übergabe des jeweils letzten Standes.

Deswegen wird sich für die Übertragung der Dateien für das Datenaustauschformat XML entschieden.

5.1.6. Sicherheit

Die Daten werden nicht innerhalb des Intranets, sondern über das Internet gesendet. Deswegen muss entschieden werden, wie die Sicherheitsproblematik gehandhabt wird. Es muss verhindert werden, dass Unbefugte Zugriff auf die Daten erhalten oder diese manipulieren können. Wichtig ist, welches Protokoll zum Datentransfer genutzt wird. Deswegen wird im nächsten Unterabschnitt beschrieben, warum das Protokoll zum Datentransfer von File Transfer Protocol zu Secure File Transfer Protocol (SFTP) geändert werden muss. Anschließend, wieso die Dateien verschlüsselt werden müssen, insbesondere die XAML-Dateien und die XML-Dateien. Die Realisierung der Verschlüsselung der Dateien ist nicht Teil diese Arbeit.

²⁵Beispiel unter <http://www.codeproject.com/Articles/144820/WPF-XmlDataProvider-Two-Way-Data-Binding-Enhanced>. Abgerufen am 30.09.14

5.1.6.1. File Transfer Protokoll

FTP ist ein altes Protokoll, das in den 1970er Jahren entwickelt wurde. Die Nutzung des FTP-Protokolls wird als sehr unsicher eingestuft, weil die Übertragung der Daten (auch Kennwort) unverschlüsselt über das Netz erfolgt. Es wäre möglich, Daten einzusehen und zu manipulieren und diese dann an den Zielrechner oder -server weiterzugeben. Das Kennwort wird in Klartext übertragen, so kann ein unbefugter Zugang zum Netz erhalten.

SFTP wurde in den 1990er Jahren entwickelt. Die Nutzung des SFTP-Protokolls ist sicherer als die des FTP-Protokolls, weil die Daten über eine sicheren Verbindung übertragen werden.

SFTP verwendet das SSH2 Protokoll, damit werden die Daten und das Kennwort nicht im Klartext übertragen (vgl. [Technologies \(2013\)](#) und [WISE-FTP \(2014\)](#)).

Im Gespräch mit der Systemadministration wurde entschieden, dass der FTP-Server auf das SFTP-Protokoll geändert werden muss. Dies wird aber erst gegen Ende dieses Jahres (2014) erfolgen.

5.1.6.2. Dateien verschlüsseln

Die Realisierung der Verschlüsselung von Dateien ist nicht Teil dieser Arbeit, deswegen werden hier nur kurz die Gründe genannt, warum eine Verschlüsselung sinnvoll ist. Ein Zugriff auf den FTP-Server ist vom Internet aus möglich und folglich ist er angreifbar, deshalb müssen die Daten verschlüsselt werden. Insbesondere bei XAML ist es möglich, Inline Code²⁶ zu schreiben, der beim Laden des XAML ausgeführt wird. Das gleiche gilt für XML-Dateien, die zusätzlich noch sensible Produkt-Informationen beinhalten.

5.2. ERP-System (Server)

Das ERP-System stellt für den Client die benötigten Produkt-Informationen über ein Produkt auf dem FTP-Server zu Verfügung. Ebenso lädt sich das ERP-System die freigegebenen Produkte vom FTP-Server und trägt diese in die Datenbank ein. Es benötigt keine Informationen über den Client und funktioniert somit unabhängig vom diesem. Darüber hinaus stellt es keine Schnittstellen für den Client zur Verfügung, weil es von außerhalb nicht erreichbar ist. Das

²⁶C#-Code, der im XAML geschrieben ist.

ERP-System gibt die Vorgabe, wie die XML hierarchisch aufgebaut sein muss, um die Daten später wieder in die ERP-Datenbank einzutragen zu können. Deswegen muss auch die XAML (View) vom ERP-System vorgegeben und auf dem FTP-Server geladen werden.

Für das ERP-System soll ein Entwurf erstellt, aber in diese Arbeit nicht umgesetzt werden.

5.2.1. Sichten

Die nachfolgenden Sichten stellen einen detaillierten Blick auf das ERP-System und dessen Ablauf dar.

5.2.1.1. Bausteinsicht

Zum ERP-System werden zwei weitere Komponenten hinzugefügt (siehe Abbildung 5.6). Es werden nur die Komponenten beschrieben, die für den Client relevant sind. Die existierenden Komponenten, um Produkte zu erstellen oder zu bearbeiten, sind in Abbildung 5.6 nicht enthalten. Wie im Kapitel *Grundlagen* beschrieben, ist Konzept 16 keine objektorientierte Programmiersprache.

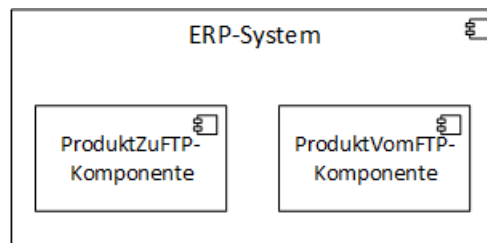


Abbildung 5.6.: Bausteinsicht des ERP-System

ProduktZuFTP-Komponente

Diese Komponente enthält die Funktionen, mit denen überprüft wird, welche Produkte an den FTP-Server gesendet werden müssen. Dafür werden die XML-Dateien erzeugt und dann gesendet. Zusätzlich werden die Funktionalitäten für XML und FTP inkludiert.

ProduktVomFTP-Komponente

Diese Komponente enthält die Funktionen, mit denen überprüft wird, welche Produkte vom FTP-Server geladen werden müssen. Diese XML-Dateien werden dann heruntergeladen und in die Datenbank eingefügt. Zusätzlich werden die Funktionalitäten für XML und FTP inkludiert.

5.2.1.2. Laufzeitsicht

Der dargestellte Ablauf beschreibt, wie die Produkte vom FTP-Server geladen und zum FTP-Server gesendet werden (siehe Abbildung 5.7).

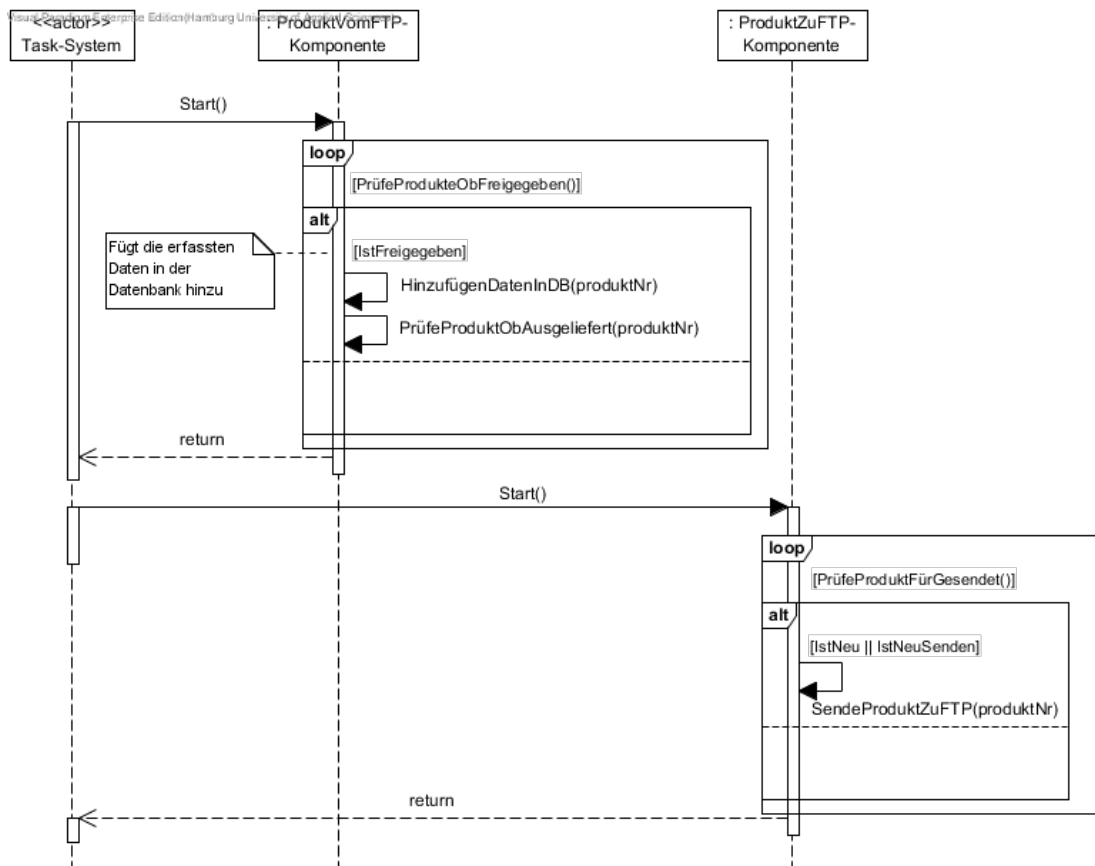


Abbildung 5.7.: Laufzeitsicht des ERP-System

1. Das Task-System startet die Funktion, mit der die freigegebene Produkte vom FTP-Server geladen werden.
2. Die ProduktVomFTP-Komponente durchsucht nun solange den Ordner *freigegeben*, bis es keine freigegebenen Produkte mehr auf dem FTP-Server gibt.
3. Ist ein Produkt freigegeben, wird es heruntergeladen, in die Datenbank mit den Zustand *freigegeben* eingefügt und dann vom FTP-Server gelöscht.

4. Anschließen wird geprüft, ob das Produkt ausgeliefert worden ist. Wenn ja, erhält es den Zustand *fertig*, sonst den Zustand *neuSenden*. Dies wird zusätzlich in einem Protokoll erfasst und geloggt.
5. Das Task-System startet nun die Funktion, mit der die zu sendenden (Zustand *neu* und *neuSenden*) Produkte vom ERP-System zum FTP-Server gesendet werden.
6. Die ProduktZuFTP-Komponente prüft alle Produkte mit den Zustand *neu* und *neuSenden*. Für diese wird jeweils eine XML-Datei erzeugt und an den FTP-Server gesendet. Dies wird zusätzlich in einem Protokoll erfasst und geloggt.

5.2.2. Strukturen

Dieser Abschnitt beschreibt, wie die XML und die XAML für den Server und die Clients aufgebaut sein müssen.

5.2.2.1. XML-Struktur

Die XML-Struktur ist ein wichtiger Punkt für das Gesamtsystem. Es muss eine Möglichkeit ausgewählt werden, die den Clients für ihre Aufgabe genügend Informationen zu Verfügung stellt. Die XML-Datei enthält die Produkt-Informationen, die erfasst werden müssen. Die Produkt-Informationen sind dynamisch und können somit für jedes Produkt unterschiedlich sein.

Erst einmal die grundlegende Struktur (siehe Auflistung 5.1). Hierbei wird sich an den drei Formularen orientiert, die bekannt sind (siehe Anhang A.3). Es ist auch möglich, dass es XML-Elemente gibt, für die es kein Eingabefeld im Formular gibt, um sie zu erfassen.

```
1 <Product>
2   <GENERAL_INFORMATION>
3     <Style></Style>
4     <Knit_Woven></Knit_Woven>
5     <BC_Merch></BC_Merch>
6     <Agent></Agent>
7     <Agent_Merch></Agent_Merch>
8     <Factory></Factory>
```

```
9     <Total_Lines_of_the_Factory></Total_Lines_of_the_Factory>
10    <BC_QC></BC_QC>
11    <Order_Placement></Order_Placement>
12    <Order_Qty></Order_Qty>
13    <Packs></Packs>
14    <Order_Qty_in_pcs></Order_Qty_in_pcs>
15  </GENERAL_INFORMATION>
16  <PRODUCTION_AND_FRI_STATUS>
17    <Production_Report_Date></Production_Report_Date>
18    <Cut></Cut>
19    <Output></Output>
20    <Output_Balance_With_5P_Extra></Output_Balance_With_5P_Extra>
21    <Delivery_Date_Ex_Ctg_Date></Delivery_Date_Ex_Ctg_Date>
22    <DDP></DDP>
23    <FRI_Done_Qty></FRI_Done_Qty>
24    <balance_Qty></balance_Qty>
25  </PRODUCTION_AND_FRI_STATUS>
26  <FORECAST_AND_REMARKS>
27    <Days_Left_To_Ship></Days_Left_To_Ship>
28    <Running_Lines></Running_Lines>
29    <Productivity_Line_Day></Productivity_Line_Day>
30    <Days_Required_to_Stitch_Balance_Qnty></
      Days_Required_to_Stitch_Balance_Qnty>
31    <Remarks></Remarks>
32  </FORECAST_AND_REMARKS>
33 </Product>
```

Auflistung 5.1: Grundlegende XML-Struktur eines Produktes.

Es muss möglich sein, den letzten Stand mitzugeben. Hierfür wurden Überlegungen zu zwei verschiedenen Lösungsansätzen angestellt.

Der erste Lösungsansatz ist, dass zwei Baumstrukturen aufgebaut werden (siehe Auflistung 5.2). Eine enthält die alten Daten, die als letztes erfasst und bereits an das ERP-System gesendet wurden. Die andere soll die neuen Daten enthalten, die erfasst werden müssen und an das ERP-System gesendet werden sollen. Dadurch wären alle Elemente doppelt enthalten, auch wenn nur für bestimmte Elemente alte Daten benötigt werden, was zu unnötigem Overhead führen würde.

```
1 <LastProduct>
2 ...
3   <PRODUCTION_AND_FRI_STATUS>
4 ...
5     <Output>100</Output>
6 ...
7   </PRODUCTION_AND_FRI_STATUS>
8 ...
9 </LastProduct>
10 <Product>
11 ...
12   <PRODUCTION_AND_FRI_STATUS>
13 ...
14     <Output></Output>
15 ...
16   </PRODUCTION_AND_FRI_STATUS>
17 ...
18 </Product>
```

Auflistung 5.2: Zwei Baumstrukturen eines Produktes, um den letzten und den aktuellen Informationsstand zu erfassen.

Der zweite Lösungsansatz ist, dass die alten Daten, die bereits an das ERP-System gesendet wurden, nur als Attribute übergeben werden (siehe Auflistung 5.3). Deshalb wird nur eine Baumstruktur benötigt, in der die Daten erfasst werden, die an das ERP-System gesendet werden sollen. Nur die Elemente, die alten Daten benötigen, erhalten ein Attribut. Dadurch entsteht kein unnötiger Overhead wie im ersten Lösungsansatz.

```
1 <Product>
2 ...
3   <PRODUCTION_AND_FRI_STATUS>
4 ...
5     <Output lastProduct="100"></Output>
6 ...
7   </PRODUCTION_AND_FRI_STATUS>
8 ...
9 </Product>
```

Auflistung 5.3: Der letzte Informationsstand für ein XML-Element als Attribute.

Es wurde sich für die zweite Variante entschieden. Jede Information, die ein XML-Element benötigt, wird als Attribut geschrieben. Dies ermöglicht, dass auch die Summenformel (im Folgenden beschrieben) als Attribut geschrieben werden kann.

Das XML-Element *Output_Balance_With_5P_Extra* wird zusammengerechnet aus $Order_Qty_in_pcs * 1,05$. Um eine Summenformel zu berechnen, wird die Bezeichnung *function* hinzugefügt als Attribut. Damit unterschieden werden kann, welche der Werte ein XML-Element und welcher ein Attribut darstellt, wurde entschieden, dass ein XML-Element in [] geschrieben wird und Attribut am Anfang ein @ enthält. Der Sinn des @ ist, dass so bestimmte Werte (wie zum Beispiel 1,05) Variablen sind, die geändert werden können. In Auflistung 5.4 ist ein Beispiel, wie dies aufgebaut sein kann. [*Output_Balance_With_5P_Extra/@Prozent*] bedeutet, dass sich das Attribut *Prozent* im XML-Element *Output_Balance_With_5P_Extra* befindet.

```
1 <Product>
2 ...
3   <PRODUCTION_AND_FRI_STATUS>
4 ...
5     <Output lastProduct="100"></Output>
6     <Output_Balance_With_5P_Extra Prozent="5" function="[
7       Order_Qty_in_pcs] * (1+ ([Output_Balance_With_5P_Extra/@Prozent
8         ]/100)) - [Output]"></Output_Balance_With_5P_Extra>
9     ...
10  </PRODUCTION_AND_FRI_STATUS>
11  ...
12 </Product>
```

Auflistung 5.4: Summenformel in einem XML-Element.

5.2.2.2. XAML-Struktur (Formular View)

Von der Server-Seite muss neben der XML- auch die XAML-Struktur vorgegeben werden. In Auflistung 5.5 wird eine beispielhafte Formular View dargestellt. Diese können den Inhalt von XML-Elementen anzeigen. Dafür muss die XML zuerst geladen werden, zum Beispiel in ein `XmlDocument`²⁷. Das `XmlDocument` befindet sich im `ViewModel`.

²⁷[http://msdn.microsoft.com/de-de/library/system.xml.xmldocument\(v=vs.110\).aspx](http://msdn.microsoft.com/de-de/library/system.xml.xmldocument(v=vs.110).aspx). Abgerufen am 30.09.14

```

1 <UserControl ...>
2 .....
3 <Grid Margin="0" DataContext="{Binding XmlDoc}">
4 ...
5     <StackPanel Grid.Row="0" Margin="5" DataContext="{Binding
6         Mode=TwoWay, XPath=Product/PRODUCTION_AND_FRI_STATUS,
7         UpdateSourceTrigger=PropertyChanged}">
8         <StackPanel Orientation="Horizontal" Style="{
9             StaticResource fieldsetStyle}">
10            <Label Style="{StaticResource labelStyle}">
11                Production_Report_Date:</Label>
12            <dx:DateEdit Style="{StaticResource dateeditStyle}"
13                Text="{Binding XPath=Production_Report_Date,
14                UpdateSourceTrigger=PropertyChanged}"/>
15        </StackPanel>
16        .....
17        <StackPanel Orientation="Horizontal" Style="{
18            StaticResource fieldsetStyle}">
19            <Label Style="{StaticResource labelStyle}">
20                balance_Qty:</Label>
21            <TextBox Style="{StaticResource textboxStyle}" Text=
22                "{Binding XPath=balance_Qty, UpdateSourceTrigger
23                =PropertyChanged}" />
24        </StackPanel>
25    </StackPanel>
26 </Grid>
27 </UserControl>

```

Auflistung 5.5: Beispielhafte Formular View, um XML Inhalte darzustellen.

Der DataContext wird auf das XmlDocument gesetzt. Anschließend kann die XPath gesetzt werden, sodass nur die Elemente und Kinder von beispielsweise *PRODUCTION_AND_FRI_STATUS* in diesem View relevant sind. Dies führt dazu, dass beim Binden der Textfelder nur noch *XPath=balance_Qty* anstatt *XPath=Product/PRODUCTION_AND_FRI_STATUS/balance_Qty* anzugeben ist. Wenn der Anwender etwas an den Daten ändert, dann geschieht dies direkt im XmlDocument. Am Ende kann die Änderungen direkt in die XML-Datei gespeichert werden.

5.3. SL-System (Client)

Der Client wird lokal auf den Rechnern der Mitarbeiter installiert und enthält die Darstellungs- und die Anwendungslogik. Der Server gibt die XML- und XAML-Struktur vor. Der Server stellt dem Client keine Schnittstellen zu Verfügung. Die Daten werden persistent auf einen Datenträger (zum Beispiel Festplatte) gespeichert.

Die Anwendung wird in C# geschrieben und die grafische Benutzeroberfläche wird mit der WPF realisiert.

5.3.1. Architekturmuster

Der Client wird nach dem MVVM-Entwurfsmuster aufgebaut. Einerseits, weil das MVVM speziell für das WPF konzipiert ist, im Vergleich zu MVC, und andererseits, weil es mehrere Views gibt (wegen den verschiedenen Formularen), die auf das gleiche ViewModel zugreifen. Deshalb muss nur die Oberfläche geändert werden, anstatt auch noch die Anwendungslogik.

5.3.2. Frameworks, Libraries und Tools

In diesem Abschnitt werden die verwendeten Frameworks, Libraries und Tools vorgestellt.

5.3.2.1. DevExpress

DevExpress²⁸ ist ein .NET Application Framework und bietet zahlreiche Funktionalitäten und wiederverwendbare Codebestandteile für WinForms, WPF, ASP.Net, Silverlight und Windows 8 XAML (Touch UI). Die Verwendung von DevExpress für Entwickler kostet monatlich Geld.

Bei der Umsetzung werden die Funktionalitäten für die Views (zum Beispiel Textbox, Dock-Window usw.), Notifikation, Kommandos (Commands), Messagerivce und Dialogservice (um Modal Dialoge zu nutzen) von DevExpress genutzt.

²⁸<https://www.devexpress.com/>. Abgerufen am 30.09.14

5.3.2.2. FTP-Library

Das .NET-Framework stellt wenige Funktionalitäten für FTP zur Verfügung, deswegen wird die Library WinSCP²⁹ genutzt. Diese bietet neben den Funktionalitäten des FTP-Protokolls noch die des SFTP-Protokolls an.

WinSCP ist eine Open-Source .NET Library und bietet die Protokolle FTP, SFTP, **Web-based Distributed Authoring and Versioning** (WebDAV) und **Secure Copy Protocol** (SCP) an, um Verbindung mit einem FTP-Server herstellen zu können und Daten zu senden bzw. zu laden.

Leider ist WinSCP für GUI Applikationen nicht optimal geeignet, aber im Vergleich gibt es keine Open-Source .NET Library, die FTP und SFTP unterstützt und die weiter entwickelt wird.

5.3.2.3. ftpdmin

ftpdmin³⁰ ist ein Programm und muss nicht installiert werden. Es stellt ein minimalen FTP-Server dar und kann zum Beispiel für Tests genutzt werden, bei denen ein FTP-Server benötigt wird. Beim Ausführen des ftpdmin muss ein Verzeichnis ausgewählt werden, das als Stammverzeichnis (engl. root directory) dient. Zum Verbinden mit dem FTP-Server wird kein Benutzername und Passwort benötigt.

5.3.3. Sichten

Die nachfolgenden Sichten stellen einen detaillierten Blick auf das SL-System und dessen Ablauf dar. Sie werden in Englisch beschrieben.

5.3.3.1. Bausteinsicht

In der Bausteinsicht wird die Struktur des Systems dargestellt. Die Darstellung beinhaltet die Aufteilung des Systems in die einzelnen Bausteine und deren Schnittstellen. Die Ansicht kann durch die Darstellung der einzelnen Bausteine verfeinert werden (vgl. **Starke, 2014**, S.75).

In Abbildung 5.8 wird der Aufbau des SL-System dargestellt.

²⁹<http://winscp.net/>. Abgerufen am 30.09.14

³⁰<http://www.sentex.net/~mwandel/ftpdmin/>. Abgerufen am 30.09.14

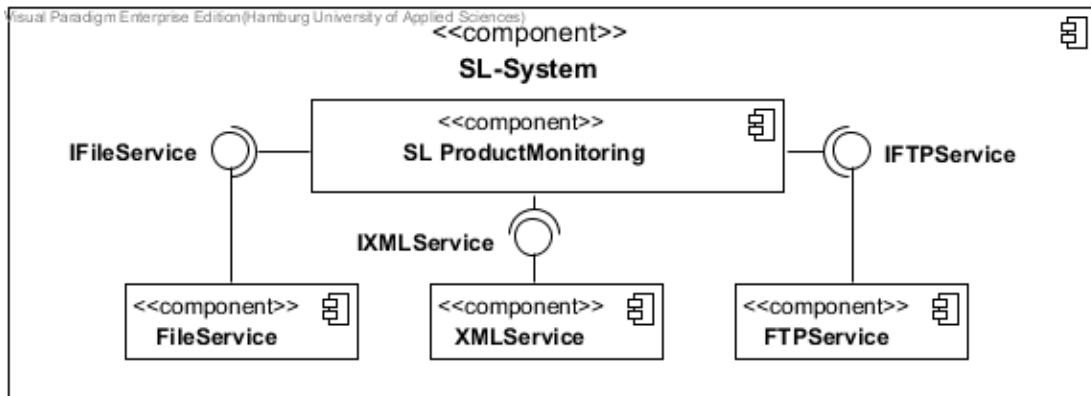


Abbildung 5.8.: Bausteinsicht des SL-System

SL ProductMonitoring

Das SL ProductMonitoring ist für die Überwachung und Erfassung von Produkten zuständig. Es enthält View, ViewModel, Model und die Business Logik für die Anwendung.

FileService

Der FileService enthält die Logik für die Bearbeitung von lokale Daten.

FTPService

Der FTPService enthält die Logik für die Bearbeitung von Daten, die den FTP-Server betreffen.

XMLService

Der XMLService enthält die Logik für die Bearbeitung von XML.

In Abbildung 5.9 wird die Innensicht des SL ProductMonitoring dargestellt.

View

In der Schicht View befinden sich die Dialoge für Main (MainWindow), FTP Login (LoginFTPUserController), Produkt Auswahl (SelectProductUserController), Formular Auswahl (SelectReportUserController) und später für die Konfigurationen. Der Dialog MainWindow ist das Hauptfenster, in dem die Produkte erfasst werden können. Die Formulare werden zur Laufzeit gelesen, deswegen enthält das MainWindow eine UI-Logik, um die Formulare anzeigen zu können. Das MainWindow benötigt dementsprechend den FileService, um eine Liste von XAML-Dateien zu erhalten. MainWindow wird als **Window** erstellt. FTP Login, Produkt Auswahl und Formular Auswahl werden als **UserControl** erstellt, dadurch können sie wiederverwendet werden.

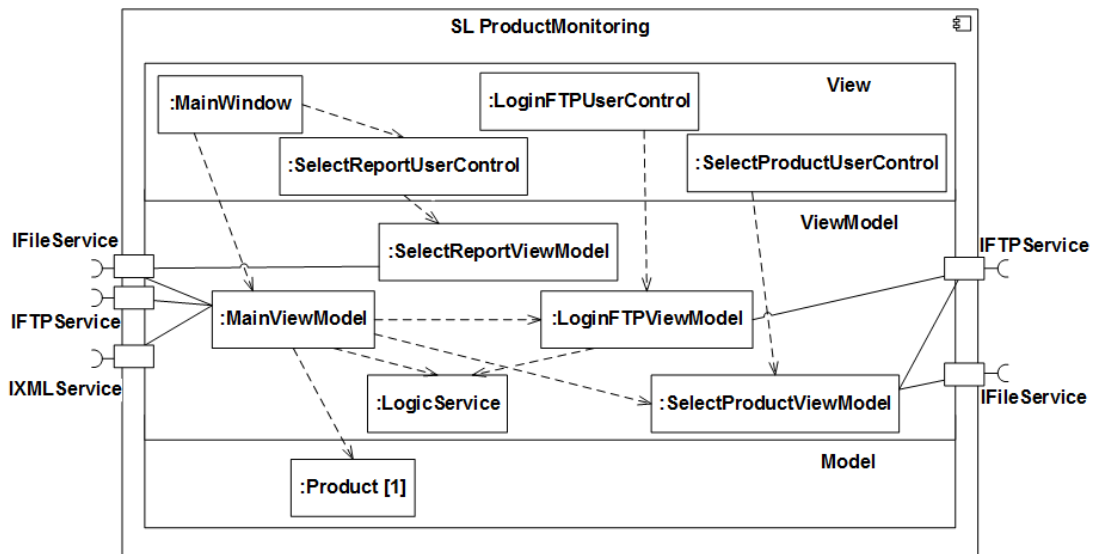


Abbildung 5.9.: Innensicht für das SL ProductMonitoring

ViewModel

In der Schicht ViewModel befinden sich ViewModels für alle Views. Diese stellen die Informationen für die zugehörigen Views bereit. Ein FTPService wird erst erzeugt, wenn eine Verbindung mit dem FTP-Server hergestellt werden soll. Der LogicService enthält die Business Logik der ViewModels.

Model

In der Schicht Model befindet sich die Klasse für das Produkt, welches hauptsächlich als Daten-Container und für die Validierung dient.

5.3.3.2. Laufzeitsicht

Die dargestellten Abläufe beschreiben, wie ein Produkt ausgewählt, vom FTP-Server geladen und an den FTP-Server gesendet wird. Die Diagramme zeigen keine exakten Abläufe, um möglichst Platz zu sparen. Genauere Abläufe von Methoden können zum Beispiel mit Visual Studio 2013 automatisiert generiert werden.

Inspektor/Regionalverantwortlicher wählt einen Produkt aus, um es anschließend herunterzuladen.

Der Inspektor/Regionalverantwortliche will Produkt-Informationen eines Produktes erfassen. Dafür muss er zuerst ein Produkt auswählen. Er verbindet sich mit dem FTP-Server und

wählt ein Produkt aus (siehe Abbildung 5.10). Anschließend wird das Produkt heruntergeladen (siehe Abbildung 5.11). Der Ablauf, ein Produkt lokal auszuwählen, funktioniert nach einem ähnlichen Schema wie in Abbildung 5.10. Der Unterschied ist, dass anstatt das *FTPService* der *FileService* für die Methode *UpdateFolderCollection* genutzt wird. In diesem Fall wird lokal geprüft. In diesem Ablauf (Produkt auswählen und vom FTP-Server herunterladen) wird davon ausgegangen, dass der Akteur sich bereits mit dem FTP-Server verbunden hat.

Die Abbildung 5.10 zeigt den Ablauf, bei dem ein Inspektor/Regionalverantwortlicher ein Produkt auf dem FTP-Server auswählt.

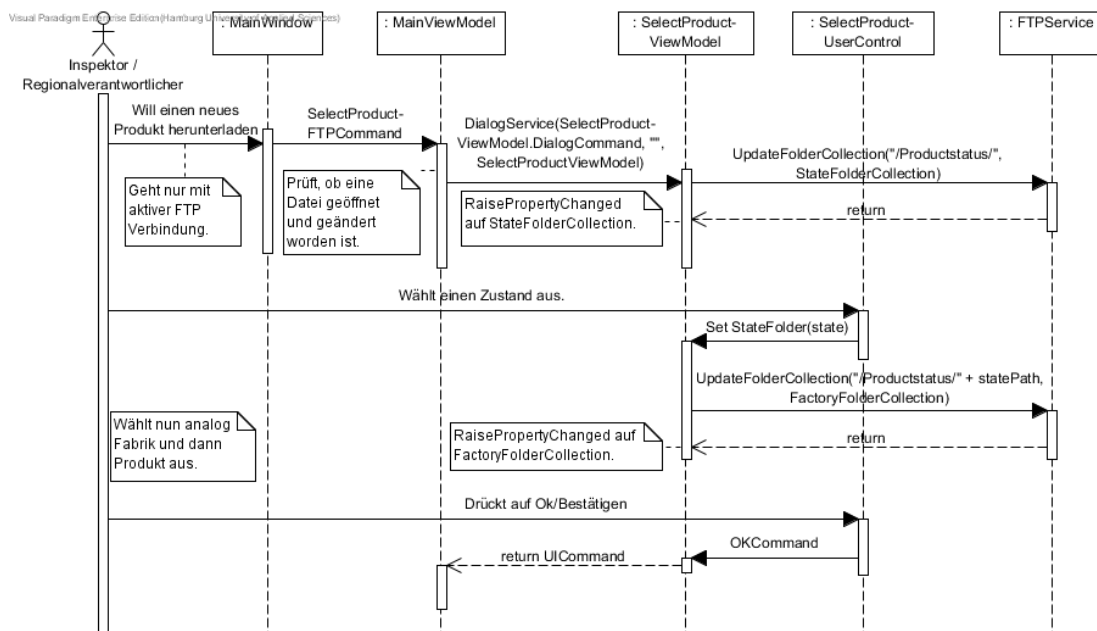


Abbildung 5.10.: Laufzeitsicht: Produkt auf dem FTP-Server auswählen.

1. Wenn der Akteur ein Produkt herunterladen will, klickt er auf den Button für die Produktauswahl im FTP-Tab. Wenn er ein Produkt lokal öffnen will, klickt er im File-Tab auf Öffnen.
2. Das *MainViewModel* öffnet über den *DialogService* einen Modal Dialog für Produkt Auswahl.
3. Das *SelectProductViewModel* füllt die Liste (engl. collection) zur Auswahl von Zuständen.
4. Der Akteur wählt nun einen Zustand aus.

5. Das SelectProductViewModel füllt die Liste (engl. collection) zur Auswahl von Fabriken.
6. Der Akteur wählt nun eine Fabrik aus.
7. Das SelectProductViewModel füllt die Liste (engl. collection) zur Auswahl von Produkten.
8. Der Akteur wählt nun ein Produkt aus.
9. Der Akteur drückt auf den Ok-Button und der Produkt Auswahl Dialog schließt sich.
10. Wenn das Produkt lokal ausgewählt worden ist, dann wird es nun geöffnet. Wenn über den FTP-Server ein Produkt ausgewählt worden ist, dann geht der Ablauf weiter (siehe Abbildung 5.11).

Die Abbildung 5.11 zeigt den Ablauf, bei dem ein Inspektor/Regionalverantwortlicher ein Produkt ausgewählt hat, das anschließend heruntergeladen wird.

1. Das MainViewModel prüft zuerst, ob die Datei lokal existiert. Falls sie existiert, wird geprüft, ob die XML-Datei lokal neuer ist. Wenn ja, wird abgefragt, ob die Datei überschrieben werden soll. Falls sie nicht existiert, wird geprüft, ob die Datei bereits in einem anderen Zustand lokal existiert. Wenn ja, wird abgefragt, ob die Datei überschrieben werden soll.
2. Falls die Datei nicht existiert oder auf Überschreiben geklickt worden ist, wird geprüft, ob die Ordner für Zustand, Fabrik und Produkt existieren. Wenn nein, dann werden sie erstellt.
3. Der FTPService lädt anschließend den Inhalt des Ordners Produkt vom FTP-Server herunter.
4. Der XMLService lädt die XML-Datei und gibt deren Inhalt an das MainViewModel zurück.
5. Das MainViewModel macht zuletzt einen RaisePropertyChange, damit die Views erfahren, dass es eine Änderung gab.

5. Design und Realisierung

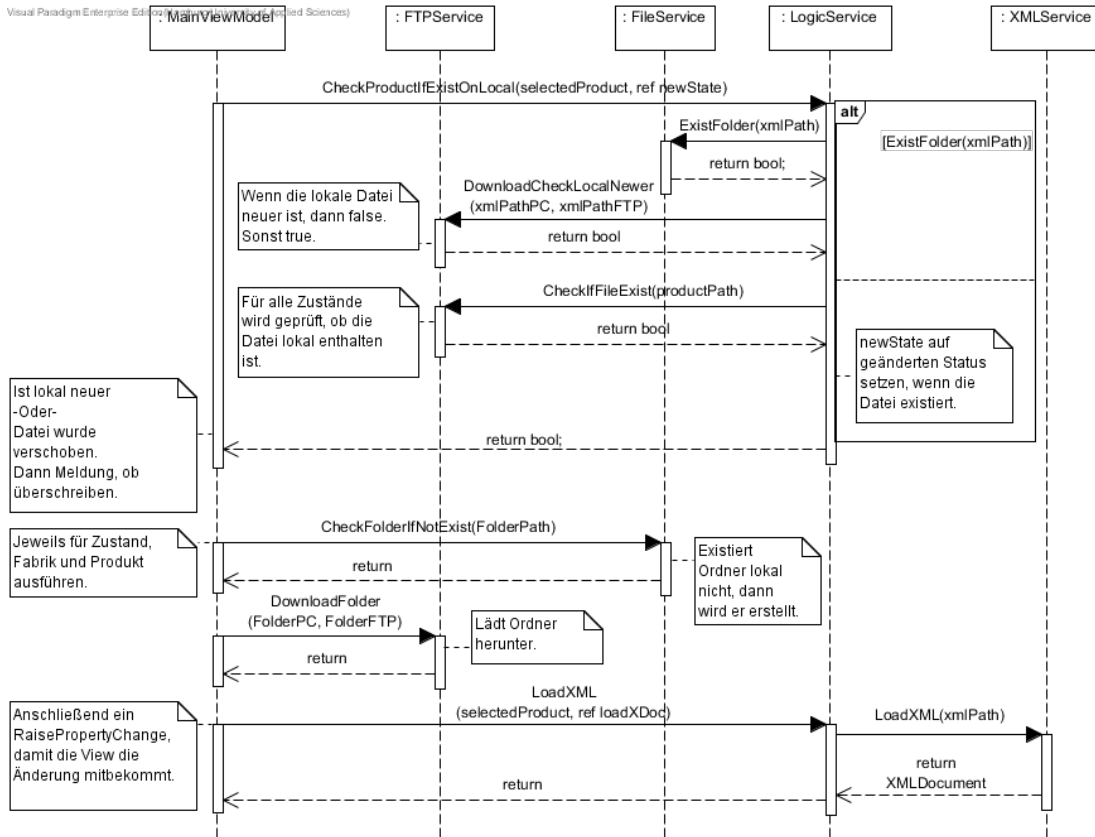


Abbildung 5.11.: Laufzeitsicht: Produkt vom FTP-Server herunterladen.

Inspektor/Regionalverantwortlicher sendet ein Produkt an den FTP-Server.

Der Inspektor/Regionalverantwortliche will ein Produkt an den FTP-Server senden. Er verbindet sich mit dem FTP-Server und hat ein Produkt ausgewählt bzw. geöffnet. Anschließend drückt er auf Produkt senden (siehe Abbildung 5.11). Der Ablauf, ein Produkt freizugeben bzw. die Freigabe zu entfernen, funktioniert nach dem selben Schema, nur dass es an einen anderen Zustand erhält.

Die Abbildung 5.12 zeigt den Ablauf, bei dem ein Inspektor/Regionalverantwortlicher ein Produkt an den FTP-Server sendet.

1. Der Akteur will ein Produkt senden und klickt im FTP-Tab den Button, um ein Produkt zu senden.

2. Der LogicService prüft zuerst, ob die Datei in dem ausgewählten Pfad auf dem FTP-Server existiert.
3. Falls die Datei existiert, wird geprüft, ob die Datei nach dem Herunterladen durch ein anderen Akteur geändert wurde. Wenn ja, wird abgefragt, ob die Datei überschrieben werden soll.
4. Falls die Datei nicht existiert, wird geprüft, ob die Datei bereits in einem anderen Zustand auf dem FTP-Server existiert. Es kann passieren, dass die Datei freigegeben worden ist und der Inspektor die Datei nicht sieht. Es kann aber auch passieren, dass das Produkt ausgeliefert worden ist und somit fertig ist. Deshalb wird es nicht mehr neu vom ERP-System generiert und im Zustand neu an den FTP-Server gesendet. Sollte dieser Fall geschehen, dann ist es nicht möglich die Datei hochzuladen. Falls die Datei in einem anderen Zustand existiert und der Akteur die Datei sehen kann, wird abgefragt, ob die Datei überschrieben werden soll.
5. Das MainViewModel will das Produkt an den FTP-Server senden.
6. Der FTPService prüft, ob sich der Zustand ändert. Wenn ja, wird der Ordner auf dem FTP-Server verschoben. Anschließend wird die Datei hochgeladen. Falls sich der Zustand geändert hat, werden die leeren Ordnern gelöscht (außer die Zustände).
7. Wenn deleteFolderPC gesetzt ist, wird die lokale Datei gelöscht. Sonst wird die Datei in den Ordner im neuen Zustand verschoben.

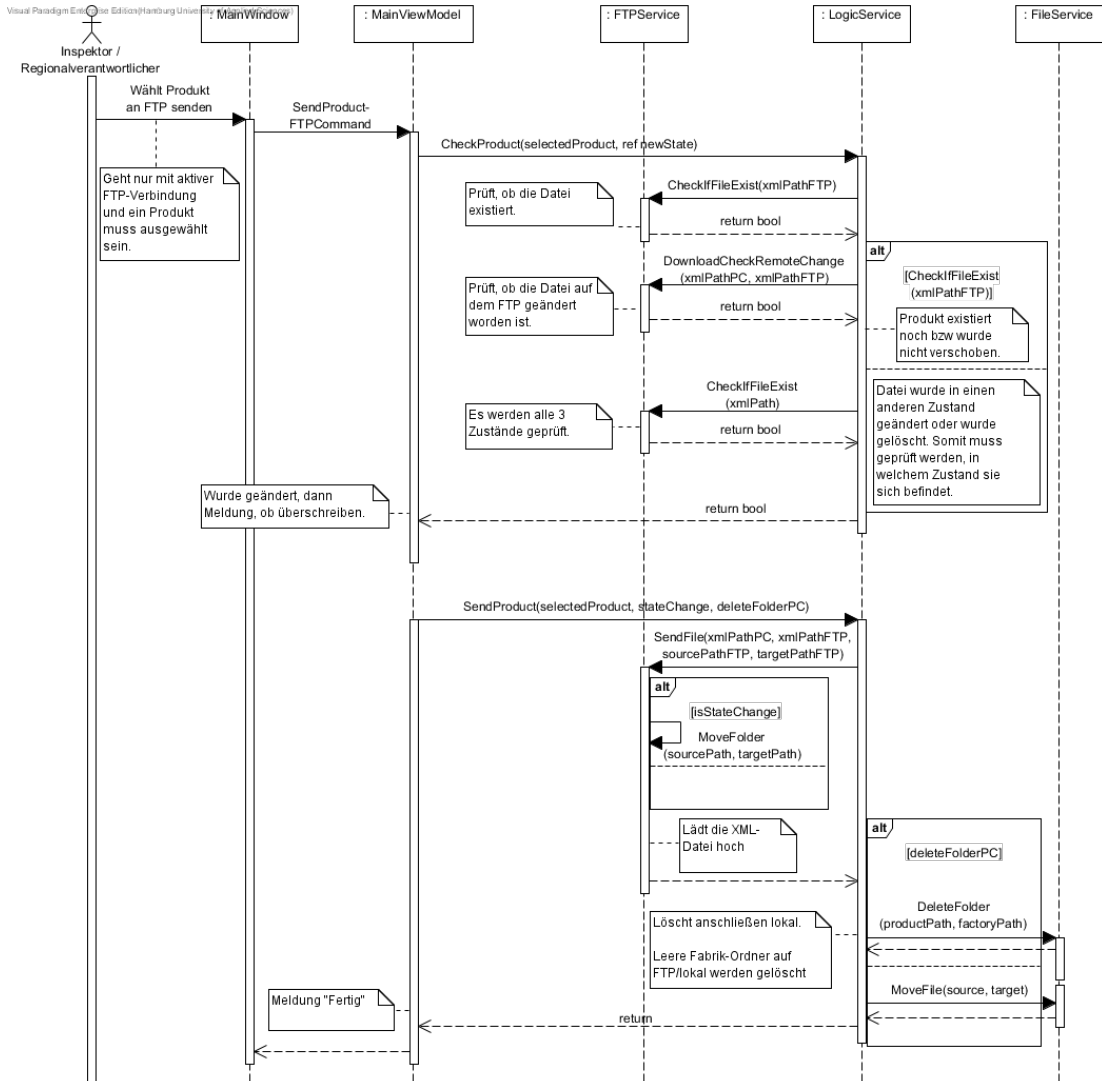


Abbildung 5.12.: Laufzeitsicht: Produkt an FTP-Server senden.

5.3.4. GUI Design

Für das Design des Views bietet DevExpress viele Funktionalitäten. Eine dieser Funktionalitäten ist das Dock-Window³¹. Es ist somit möglich, für jedes Formular ein eigenständig Fenster (View) zu öffnen. Diese Fenster können je nach belieben auf dem Desktop verteilt, in einem Fenster zusammengetragen als Tabs oder Nebeneinander gesetzt werden (siehe Abbildung 5.13). Die Views für die Formulare verwenden das selbe ViewModel, deswegen können alle Änderungen ohne Probleme jedem View mitgeteilt werden.

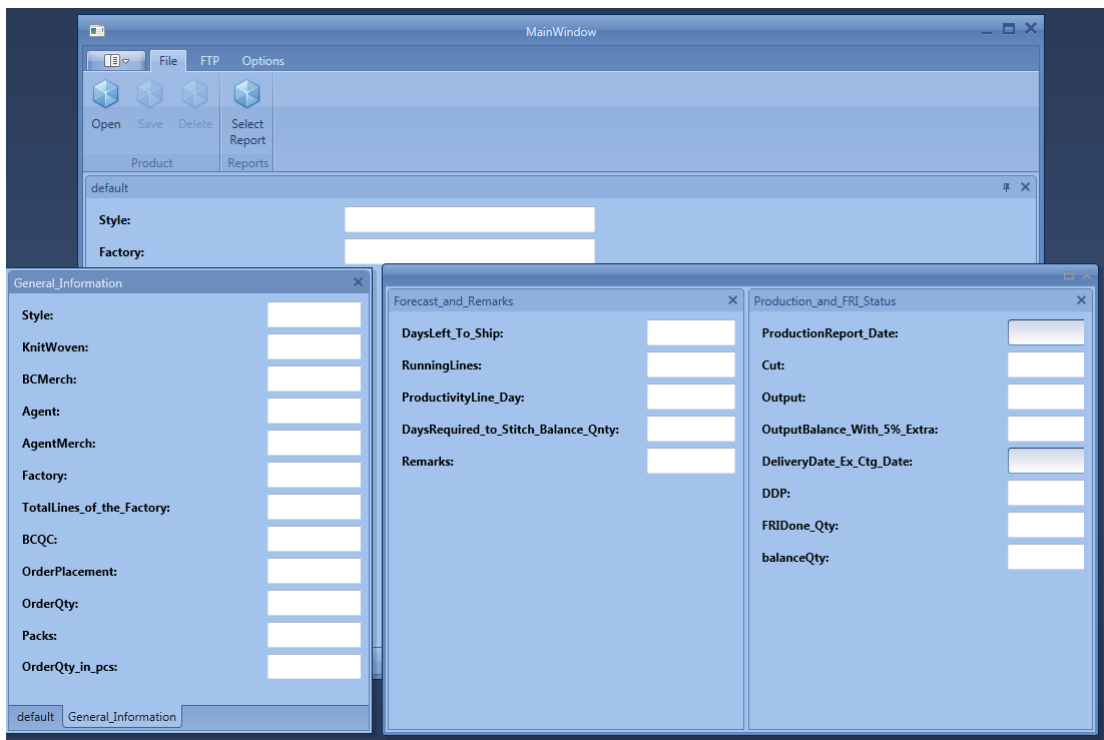


Abbildung 5.13.: GUI-Design mit verschiedene geöffnete Formulare

³¹<https://documentation.devexpress.com/#WPF/CustomDocument6191>. Abgerufen am 30.09.14

5.3.5. Probleme

In diesem Abschnitt werden die Probleme der Nebenläufigkeit und Fehlerbehandlung erläutert und die dazugehörigen Konzepte, um diese zu lösen.

5.3.5.1. Nebenläufigkeit

Wie im Abschnitt 5.1.3 beschrieben, wurde das optimistische Synchronisationsverfahren gewählt. Der Anwender soll eine Meldung zur Überschreibung der Datei erhalten, wenn sie von einem anderen Anwender zwischenzeitlich geändert worden ist. Bei der Ordnerstruktur, die gewählt worden ist, gibt es zwei Probleme. Das erste Problem tritt auf, wenn das Produkt ohne Zustandsänderung geändert bzw. gespeichert worden ist. Das zweite Problem tritt auf, wenn das Produkt mit einer Zustandsänderung geändert bzw. gespeichert worden ist.

Für das erste Problem wurden drei Ideen in Betracht gezogen. Die erste Idee war, das Änderungsdatum in der XML-Datei zu speichern. Dies würde aber dazu führen, dass jede Datei erst heruntergeladen werden müsste und dann überprüft wird, ob es eine Änderung gab. Deswegen wurde die erste Idee verworfen.

Die zweite Idee war, dass neben jeder XML-Datei noch eine zweite Datei existiert, die die Version und das Datum als Namen enthält. Das würde aber wegen der unterschiedlichen Zeitzonen zu Problemen führen. Deshalb müsste eine Zeitzone gewählt werden, mit der verglichen werden könnte. Deswegen fiel die Entscheidung auf die dritte Idee.

Die dritte Idee ist, dass das Änderungsdatum der Datei, die sich auf dem FTP-Server befindet, geprüft wird. Hierdurch würden alle Benutzer/Systeme für die selbe Zeitzone (die des FTP-Servers) prüfen, ob sich die Datei geändert hat³². Das einzige Problem beim FTP-Server ist, wenn die Uhrzeit nicht stimmt³³. Beispiel: Wenn Inspektor A um 10 Uhr in Deutschland die Datei vom FTP-Server herunterlädt und die umgerechnete Zeit des FTP-Servers 20 Minuten nachgeht (im Sinne von falsch), hat die Datei lokal trotzdem ein Erstellungsdatum von 10 Uhr anstatt 9:40 Uhr. Sollte nun ein anderer Inspektor B (auch in Deutschland) die Datei um 10:10 Uhr ändern, dann wäre das Änderungsdatum auf dem FTP-Server um 9:50 Uhr. Sollte nun Inspektor A um 10:15 Uhr die Datei an den FTP-Server senden, dann wird geprüft ob das Erstellungsdatum

³²Es lässt sich nur die Zeitzone prüfen, nicht die exakte Uhrzeit. Die lokale Uhrzeit wird beim Herunterladen und Senden von Dateien automatisch umgerechnet.

³³Auch allgemeines Problem verteilter Systeme

älter ist als das Änderungsdatum, der Datei, die auf dem FTP-Server liegt. Sollte die Datei älter sein, dann wurde die Datei zwischenzeitlich geändert. Da aber durch die Verschiebung das Änderungsdatum auf dem FTP-Server 9:50 Uhr ist, ist somit das Änderungsdatum älter als das lokale Erstellungsdatum, obwohl die Datei schon einmal geändert worden ist. Falls die FTP-Server-Zeit noch nicht angepasst wurde, müssen die Anwender eine Möglichkeit haben, die Differenz manuell auszugleichen. Für diesen Zweck muss es eine Konfigurationsdatei geben.

Das zweite Problem ist komplizierter. Ein Problem ist hier, wenn der Zustand auf *freigegeben* gesetzt ist und der Inspektor es nicht mehr sehen kann. Die Idee ist, dass zuerst überprüft wird, ob das Produkt existiert. Wenn nicht, befindet sich das Produkt im Zustand *freigegeben* oder wurde ausgeliefert und vom ERP-System nicht neu generiert. Der Inspektor kriegt eine Meldung, dass das Produkt nicht mehr existiert und nicht an den FTP-Server gesendet werden kann. Ein anderes Problem tritt bei den Regionalverantwortlichen auf. Wenn zum Beispiel ein Regionalverantwortlicher ein Produkt im Zustand *bearbeitet* speichern will, aber ein anderer es auf *freigegeben* gesetzt hat, dann erhält der Regionalverantwortliche eine Meldung zum Überschreiben, mit dem Hinweis, dass es eine Zustandsänderung gab. Sollte der Regionalverantwortliche überschreiben, dann muss das Produkt in den Zustand geändert werden, den dieser setzt. Dieses Problemschema existiert allgemein, wenn es zwischenzeitlich eine Zustandsänderung gab. Es muss sichergestellt werden, dass ein Produkt nicht mehrmals im Ordner *Produktstand* enthalten ist.

Bei diesem Problem wurde bisher nur die Idee umgesetzt, dass der Inspektor die Datei nicht an den FTP-Server senden kann, wenn er das Produkt nicht mehr sehen kann.

5.3.5.2. Fehlerbehandlung

Für die Fehlerbehandlung, zum Beispiel beim Datentransfer zum FTP-Server, gibt es noch kein zufriedenstellendes Konzept. Die Idee ist, dass, wenn eine Interaktion vom lokalen Rechner zum FTP-Server stattfindet (Hochladen), zuerst die Aktion auf dem FTP-Server ausgeführt wird. Ein FTP-Fehlerfall wird wahrscheinlich eher eintreten, als ein lokaler Fehlerfall. Jede Aktion mit dem FTP-Server wird temporär gespeichert und im Fehlerfall wird versucht, diese Aktionen wieder rückgängig zu machen. Es muss aber versucht werden, den Datentransfer zwischen dem Client und dem FTP-Server möglichst gering zu halten bzw. unnötige Datentransfers zu vermeiden. Es wurde festgelegt, mit der Entwicklung eines Konzeptes zu warten, bis nach der der Evaluierung entschieden ist, ob die Anwendung genutzt werden soll.

5.3.6. Installationsprogramm

Eine der Anforderungen an das SL-System ist ein Installationsprogramm. Hierfür wurden Installshield Limited Edition³⁴, Innosetup³⁵ und WiX Toolset³⁶ verglichen.

Installshield LE von Microsoft ist kostenlos und bietet eine Integration in Visual Studio. Das Setup kann als .exe oder .msi³⁷ erstellt werden. Bietet jedoch nur eine limitierte Funktionalität. Für eine einfache Installation reicht dies.

Innosetup ist ein kostenloses Installationsprogramm-Tool und Open Source. Es bietet als Setup die Datei als .exe an. Innosetup bietet simple(Wizard um einen Installer zu erstellen) und komplexe Konfigurationsmöglichkeiten für eine Installation an.

WiX Toolset ist ein kostenloses Installationsprogramm-Tool und Open Source. Es bietet als Setup die Datei als .msi an. WiX Toolset selbst bietet keinen Wizard an, um mit dessen Hilfe ein Setup zu erstellen.

Die Anwendung wird von den "einfachen Anwendern" installiert und nicht auf allen Rechnern des Unternehmen. Deswegen reicht eine .exe-Installation aus. Die Installationsprogramm Installshield LE und Innosetup bieten dies an. Die Entscheidung viel auf Innosetup, weil dieser mehr Funktionalitäten bietet als Installshield LE.

³⁴<http://msdn.microsoft.com/en-us/library/dn531020.aspx>. Abgerufen am 30.09.14

³⁵<http://www.jrsoftware.org/isinfo.php>. Abgerufen am 30.09.14

³⁶<http://wixtoolset.org/>. Abgerufen am 30.09.14

³⁷Microsoft Installer. Hat den Vorteil, dass es über den Windows-Systemdienst installiert werden kann.

5.3.7. Tests

Um die Funktionalitäten des Systems und Qualität des Codes zu prüfen, müssen Tests geschrieben werden. Hierfür werden Komponententests³⁸ und Systemtests³⁹ entwickelt. Sobald die Funktionalitäten für das ERP-System umgesetzt wurden, sollte das SL-System um einen Verbundtest⁴⁰ erweitert werden.

Für den FTPService wird zum Testen das Programm *ftpdmin* genutzt. Dies wird beim Start automatisch ausgeführt und anschließend am Ende des Tests wieder beendet.

Das MVVM bietet den Vorteil, dass die grafische Benutzeroberfläche nicht manuell getestet werden muss. Deswegen können die ViewModels und die Business Logik automatisiert getestet werden.

³⁸Um das Verhalten einer einzelnen Komponente isoliert zu testen.

³⁹Um das Gesamtsystem zu testen.

⁴⁰Um das Gesamtsystem zusammen mit den realen Nachbarsystemen zu testen.

6. Evaluierung

In diesem Kapitel wird die Evaluierung der entworfenen Architektur und Anwendung dokumentiert. Zuerst wird beschrieben, welche Funktionalitäten implementiert sind und welche fehlen. Anschließend wird der Prototyp vorgestellt. Hierbei wird geprüft, wie die IT-Leitung und der Regionalverantwortlichen-Koordinator mit der Anwendung zurecht kommen. Die Bewertung der einzelnen Bereiche des Prototypen erfolgte durch drei Fragestellungen: „Was hat gefallen? Was hat gefehlt? Was hat gestört?“. Die jeweiligen Antworten wurden protokolliert, um eine anschließende Bewertung und Verbesserung zu ermöglichen.

6.1. Interview

Es wurde ein Interview mit der IT-Leitung und dem Regionalverantwortlichen-Koordinator geführt, in dem der Prototyp vorgestellt worden ist. Die Evaluierung erfolgt nach dem Verfahren, das bei der Schmidt Group üblich ist. Die Anwendung bzw. das Projekt wird vorgestellt und anschließend werden typische Alltagsaufgaben mit der Anwendung erledigt. Dabei werden die drei oben aufgeführten Fragen gestellt und die Antworten protokolliert. Die für eine umfassende Befragung mittels eines Fragebogens notwendige Zeit stand der IT-Leitung und dem Regionalverantwortlichen-Koordinator nicht zur Verfügung, so dass nur eine direkte, zielgerichtete Befragung durchgeführt werden konnte.

Die von der IT-Leitung und dem Regionalverantwortlichen-Koordinator gewünschten Veränderungen an die Anforderungen, die sich aus der Auswertung ergeben haben, müssen bei der folgenden Prozessanalyse berücksichtigt werden. Sie sind somit nicht mehr Teil dieser Arbeit.

6.2. Prototyp

In den Abbildungen 6.1, 6.2 und 6.3 wird beschrieben, welche der funktionalen und nichtfunktionalen Anforderungen für die Evaluierung umgesetzt werden konnten.

Funktionale Anforderungen

FA	U/NU/F	A	FA	U/NU/F	A
ALS-1	U		ALS-10	U	
ALS-2	U		ALS-11	U	
ALS-3	NU		ALS-12	U/NU	Alle Formulare sind auswählbar
ALS-4	U		ALS-13	U	
ALS-5	U		ALS-14	NU	
ALS-6	U	Images fehlen	ALS-15	U	
ALS-7	U		ALS-16	NU	
ALS-8	U		ALS-17	U	
ALS-9	U	Nicht getestet	ALS-18	F	

Tabelle 6.1.: Umsetzung der funktionalen Anforderungen. Legende: FA - Funktionale Anforderung, U - Umgesetzt, NU - Nicht umgesetzt, F - Fehlt (nicht Teil der Arbeit), A - Anmerkung

Nichtfunktionale Anforderung (Teil 1)

NFA	U/NU/F	A
NFSL-1	U	Die Bezeichnungen sind identisch mit denen von der Excel-Tabelle
NFSL-2	F	
NFSL-3	NU	Fehlerbehandlung noch nicht umgesetzt
NFSL-4	NU	Fehlerbehandlung noch nicht umgesetzt
NFSL-5	U	
NFSL-6	F	
NFSL-7	U	Eine Einführung zu Unterschied zwischen Lokal/FTP muss sein

Tabelle 6.2.: Umsetzung der nichtfunktionalen Anforderungen. Legende: NFA - Nichtfunktionale Anforderung, U - Umgesetzt, NU - Nicht umgesetzt, F - Fehlt (nicht Teil der Arbeit), A - Anmerkung

Nichtfunktionale Anforderung (Teil 2)

NFA	U/NU/F	A
NFSL-8	U	Mit Auswahl von Formularen, Zeit ca. 30 Sekunden
NFSL-9	NU	
NFSL-10	NU	
NFSL-11	-	
NFSL-12	U	
NFSL-13	U	
NFSL-14	U	
NFSL-15	U	

Tabelle 6.3.: Umsetzung der nichtfunktionalen Anforderungen. Legende: NFA - Nichtfunktionale Anforderung, U - Umgesetzt, NU - Nicht umgesetzt, F - Fehlt (nicht Teil der Arbeit), A - Anmerkung

6.3. Auswertung

Für die Auswertung wurden diese drei Fragen gestellt:

1. Was hat Ihnen an der Anwendung gefallen?
2. Was hat Sie an der Anwendung gestört bzw. soll geändert werden?
3. Was hat Ihnen bei der Anwendung gefehlt?

Die Ergebnisse der Auswertung wurden im Folgenden zusammengefasst:

Gefallen

- Die Möglichkeit, mehrere Formulare gleichzeitig zu öffnen und diese durch das Dock-Window verschieben/bewegen zu können.
- Aufteilung der Zustände, um besser erkennen zu können, in welchem Arbeitsschritt sich die einzelnen Produkte aktuell befinden.
- Dass der Koordinator bzw. die Regionalverantwortlichen die Produkt-Informationen von den Inspektoren nicht nochmal zusätzlich abtippen müssen.

- Die dynamische Anpassung der XAML und XML.

Gestört bzw. ändern

- Eingabefelder, die nicht bearbeitet werden können und die automatisch berechnet werden, sollen farblich hervorgehoben werden.
- Die Validierung der Eingabefelder (zum Beispiel, ob der Output sinkt) soll nicht mehr durchgeführt werden, weil keine Richtwerte für die Validierung bestimmt werden können. Es soll aber weiterhin auf zulässige Zeichen geprüft werden, zum Beispiel, ob bei Output nur Ziffern eingetippt worden sind.
- Die Benennung der Ribbon-Tabs soll geändert werden. Die Überlegung ist, ob *File* und *FTP* in einem Ribbon-Tab zusammengefasst werden können.

Gefehlt

- Die Formular-Fenster können verkleinert werden, aber es erscheint kein Scrollbalken. So sind die unteren Eingabefelder nicht erreichbar.
- Der letzte Stand (vor dem aktuellen) sollte sichtbar sein, zum Beispiel über einen Tooltip.
- Dass der Inspektor bzw. Regionalverantwortliche im XML noch setzen kann, dass das Produkt fertig ist.

6.4. Fazit

Bei dem Interview hat sich herausgestellt, dass die Validierung der Eingabefelder so nicht möglich ist. Es kann passieren, dass der Output gegenüber dem letzten Stand sinkt, zum Beispiel, wenn durch Qualitätskontrollen nachträglich fehlerhafte Produkteinheiten aussortiert wurden. Ebenso kann nicht vorherbestimmt werden, wie viel in einem Monat produziert werden kann. Die Menge ergibt sich aus der Produktionsleistung vor Ort. Diese Validierungen sollen in Zukunft nicht mehr durchgeführt werden, weil keine Richtwerte für die Validierung bestimmt werden können.

Die Möglichkeit, mehrere Formulare auswählen und deren Fenster nach Belieben verschieben zu können, wurde sehr gut angenommen.

7. Schluss

Im letzten Kapitel dieser Arbeit werden die Ergebnisse und das Vorgehen zusammengefasst. Anschließend erfolgt ein Fazit. Abschließend werden im Abschnitt **Ausblick** Weiterentwicklungsmöglichkeiten beschrieben.

7.1. Zusammenfassung

Inhalt dieser Arbeit war es, am Beispiel einer Prozesskontrolle den Prozess *Produktionsüberwachung und -erfassung* des Unternehmens Schmidt Group GmbH zu bewerten und zu verbessern. Ziel war es, den Prozess in Bezug auf Transparenz und Prozessgeschwindigkeit zu optimieren.

Der aktuelle Prozess wurde erfasst, dokumentiert und analysiert, um die relevanten Schwachstellen und damit Optimierungsmöglichkeiten zu finden. Anschließend wurde ein Konzept für den neuen SOLL-Zustand besprochen und entwickelt. Aus den möglichen Vorschlägen wurde, mit den Ressourcen, die zum Zeitpunkt dieser Arbeit zur Verfügung standen, eine grundlegende Architektur-Lösung und ein Ablauf des SOLL-Zustandes entwickelt. Es sollte ein System entwickelt werden, das Daten vom ERP-System erhalten und an das ERP-System senden kann, weil keine direkte Verbindung von außerhalb des Schmidt Group-Netzes auf das ERP-System möglich ist. Deswegen wurde der FTP-Server in der DMZ von der Schmidt Group genutzt, um Daten zwischen den Systemen zu transferieren. Die Mitarbeiter in den Schwellenländern sollten eine Anwendung erhalten, damit ihre Arbeit erleichtert und die Produktionsinspektion transparenter wird. Außerdem sollte es auch möglich sein, die Anwendung offline zu verwenden. Anschließend wurde für das zu entwickelnde System (SL-System) ein Anforderungsdokument erstellt und daraus die Spezifikation erzeugt. Die Anwendung wurde in C# geschrieben und die grafische Benutzeroberfläche wurde mit der WPF realisiert.

Aus den Erkenntnissen der Analyse und der Spezifikation wurde ein Konzept für die Umsetzung der Anwendung entwickelt. Damit der Client offline funktionieren kann, werden die Daten lokal persistiert. Diese Daten werden anschließend mit dem FTP-Protokoll an den FTP-Server gesendet. Die daraus folgende Inkonsistenz wird mit dem optimistischen Synchronisationsverfahren gelöst. Für die Unterscheidung, welche der Daten bearbeitet bzw. neuer sind, wird das Erstell- und Änderungsdatum geprüft. Um unterscheiden zu können, welche Produkte sich in welchem Zustand befinden, wurde eine Ordnerstruktur ausgewählt, die die Zustände der Produkte widerspiegelt. Für den Datentransfer wurde XML als Format gewählt. Dieses dient der Anwendung als „Dateninsel“. Durch XPath kann im XAML-Code ein XML-Element oder ein -Attribut ausgewählt und angezeigt werden. Wenn ein Element zur XML-Datei hinzugefügt wird, dann müssen anstatt der Software-Logik lediglich die zur Laufzeit gelesene Dialoge angepasst werden. Die XAML- und XML-Dateien befinden sich auf dem FTP-Server, deswegen ist das Thema Sicherheit von Bedeutung. Neben dem unsicheren FTP-Protokoll ist es riskant, dass XAML- und XML-Dateien frei auf dem FTP-Server liegen. Diese könnten manipuliert werden und somit Client und Server gefährden. Deswegen wurde entschieden, dass der FTP-Server auf das SFTP-Protokoll umgestellt und die Dateien verschlüsselt werden müssen. Anschließend wurden für das ERP-System und SL-System Designentscheidungen beschrieben, unter anderem, welches Framework und welche Libraries für das SL-System genutzt werden sollen.

In der Evaluierung wurde ein Prototyp vorgestellt. Zuerst wurde besprochen, welche Funktionalitäten implementiert sind und welche noch fehlen. Anschließend wurde der Prototyp vorgestellt. Hierbei wurde geprüft, wie die IT-Leitung und der Regionalverantwortlichen-Koordinator mit der Anwendung zurecht kommen. Die Bewertung der einzelnen Bereiche des Prototypen erfolgte anhand von drei Fragestellungen: „Was hat gefallen? Was hat gefehlt? Was hat gestört?“. Die jeweiligen Antworten wurden protokolliert, um eine anschließende Bewertung und Verbesserung zu ermöglichen.

7.2. Fazit

Ziel dieser Arbeit war es, einen neuen SOLL-Zustand für den Prozess *Produktionsüberwachung und -erfassung* sowie ein entsprechendes System zu entwickeln. Der neue SOLL-Zustand führt dazu, dass der Regionalverantwortlichen-Koordinator die inspizierten Daten nicht zusätzlich in das ERP-System eintragen muss und somit entlastet wird. Dadurch, dass die Inspektoren die Daten nun einheitlich erfassen, müssen diese von den Regionalverantwortlichen nicht mehr zusammengefasst werden. Die Regionalverantwortlichen müssen lediglich die Daten prüfen und freigeben. Aufgrund dessen, dass nun die Produkte einzeln erfasst und gesendet werden, kann der Stand einzelner Prozessschritte deutlich besser erkannt werden. Theoretisch ist der Prozess insgesamt schneller, weil die einzelnen, unnötigen Aufgaben wegfallen. Ob die gewünschten Optimierungen erzielt und keine neue Schwachstellen entstanden sind, muss in der nächsten Prozessanalyse geprüft und bewertet werden.

Bis zur Nutzung müssen noch die restlichen Anforderungen und die Anforderungen, die sich aus der Evaluierung ergeben haben, für das SL-System umgesetzt werden. Technisch ist es möglich mit der Anwendung zu arbeiten, weil die Grundfunktionalitäten des SL-Systems implementiert sind. Trotzdem müssen die Anforderungen an das ERP-System noch umgesetzt werden, damit den Inspektoren und Regionalverantwortlichen Daten zu Verfügung stehen, um Produktinformationen erfassen zu können. Neben den Anforderungen an das ERP-System muss ebenso die Sicherheit des Datentransfers geregelt werden. Zurzeit sind die Dateien und der Datentransfer unverschlüsselt.

Ob die Anwendung wirklich genutzt wird, hängt maßgeblich vom Regionalverantwortlichen-Koordinator ab. Er muss die Anwendung in den Schwellenländern einführen und auch dafür sorgen, dass sie genutzt wird.

Die Architektur wurde auf einem FTP-Server aufgebaut, was zu verstärkten Problemen mit der Nebenläufigkeit führte. Durch die verschiedenen Ordner musste geprüft werden, ob die Produkte möglicherweise mehrmals vorhanden waren oder geändert wurden. Zusätzlich persistiert jeder Client seine Daten lokal und das ERP-System in der Datenbank. Diese Datenhaltung erfordert eine jeweils verstärkte Konsistenzüberwachung zwischen ERP-System bzw. Client und FTP-Server.

7.3. Ausblick

Abgesehen von den nicht umgesetzten Anforderungen, die noch erledigt werden müssen, ist es wichtig, dass vor dem Betrieb der FTP-Server vom FTP-Protokoll auf das SFTP-Protokoll umgestellt wird. Dies müsste dann im neu entwickelten System angepasst werden.

Für den Fall, dass später ein Webserver zu Verfügung steht, wäre es auch möglich diesen (zusätzlich) zu nutzen. Die Daten könnten direkt an das ERP-System weitergeleitet werden, anstatt auf dem FTP-Server zwischengelagert zu werden. Diese Möglichkeit würde wahrscheinlich die Probleme mit der Nebenläufigkeit besser umgehen. Es müsste nur noch zwischen dem ERP-System und SL-System auf Konsistenz geprüft werden.

Eines der Nebenläufigkeits-Probleme wurde so gelöst, dass der Anwender entscheiden kann, ob er die Änderungen überschreiben will oder nicht. Mit dem Microsoft Sync Framework kann man die Daten zusammenführen. Dies könnte, je nachdem, wie die Anwendung genutzt wird, ebenfalls integriert werden.

Die Struktur der Ordner wurde so gewählt, dass neben der Produkterfassung zusätzliche Informationen gesendet werden können. Eine Möglichkeit ist zum Beispiel das Hinzufügen der Option, dass Bilder mitgesendet werden.

Das SL-System kann ebenso um weitere Prozesse erweitert werden. Eine Nutzung auch außerhalb der Schwellenländer ist möglich. Das System könnten auch Personen nutzen, die in Deutschland arbeiten, aber keinen Zugriff auf das ERP-System haben. So können sie Daten erhalten, ausfüllen und zurücksenden.

Abkürzungsverzeichnis

AERP Anforderung ERP

ASL Anforderung Schwellenland (SL)

BPM Business Process Model

BPMN Business Process Model and Notation

DMZ Demilitarisierte Zone

ERP Enterprise-Resource-Planning

FTP File Transfer Protocol

GUI Graphical-User-Interface

JSON JavaScript Object Notation

MVC Model-View-Controller

MVVM Model-View-ViewModel

NFERP Nichtfunktionale Anforderung ERP

NFSL Nichtfunktionale Anforderung Schwellenland (SL)

SCP Secure Copy Protocol

SFTP Secure(SSH) File Transfer Protocol

SL-System Schwellenland-System

VPN Virtual Private Network

VPN-Client Virtual Private Network-Client

WebDAV Web-based Distributed Authoring and Versioning

WPF Windows Presentation Foundation

XAML Extensible Application Markup Language

XML Extensible Markup Language

Glossar

Demilitarisierten Zone Die DMZ ist eine Firewall-Technik und schützt das interne Netz vor unberechtigten Zugriffen von außen⁴¹.

DIN EN ISO 9000 Das ISO-9000 ist ein Zertifikat für die Qualitätsmanagementnorm⁴².

Enterprise-Resource-Planning Ein ERP System ist eine Anwendungssoftware, die alle Ressourcen (Auslieferung, Kapital, Personal, Vertrieb, usw.) eines Unternehmens verwaltet und steuert⁴³.

Extensible Application Markup Language XAML ist eine von Microsoft entwickelte XML-basierte Markupsprache und Teil des WPF. Sie ist, wie HTML, für die visuelle Darstellungen zuständig⁴⁴.

Extensible Markup Language XML ist ein textbasiertes Datenaustauschformat, das der Darstellung hierarchisch strukturierter Daten dient.

JavaScript Object Notation JSON ist ein lesbares, textbasiertes Datenaustauschformat und wird in JavaScript Syntax dargestellt.

Mockups Ein Mockup (deutsch: Attrape) ist eine Art Skizze für Darstellungen, deren Aufbau und Struktur.

⁴¹<http://www.itwissen.info/definition/lexikon/demilitarized-zone-DMZ-Demilitarisierte-Zone.html>. Abgerufen am 30.09.14

⁴²<http://www.anleiten.de/qmsysteme/iso9000.html>. Abgerufen am 30.09.14

⁴³<http://wirtschaftslexikon.gabler.de/Definition/enterprise-resource-planning-system.html?referenceKeywordName=ERP-System>. Abgerufen am 30.09.14

⁴⁴<http://msdn.microsoft.com/de-de/library/cc295302.aspx>. Abgerufen am 22.10.14

Secure Copy Protocol SCP ist ein Protokoll zur verschlüsselten Übertragung von Daten⁴⁵.

UserControl UserControl bietet die Möglichkeit, ein Steuerelement zu erstellen. Steuerelemente unterstützen umfangreiche Inhalte, Formate und Auslöser. Es ist möglich, Steuerelemente mehrfach zu nutzen, ohne diese für jede Darstellung neu zu erstellen⁴⁶.

Virtual Private Network Durch VPN ist der Rechner mit dem entsprechenden Netz verbunden und kann so die von außen gesperrten Serverdienste erreichen⁴⁷.

Virtual Private Network-Client Ist ein Softwarepaket, um sich mit einem VPN verbinden zu können.

Web-based Distributed Authoring and Versioning WebDAV ist ein offener Standard zur Bereitstellung von Daten im Internet und kann ganze Verzeichnisse übertragen. Dazu ist es eine Erweiterung des Protokolls HTTP/1.1, die bestimmte Einschränkungen von HTTP aufhebt⁴⁸.

Window Window bietet die Möglichkeit, die Lebensdauer von Fenstern und Dialogfeldern zu erstellen, zu konfigurieren, anzuzeigen und zu verwalten⁴⁹.

XPath Ruft eine XPath-Abfrage ab, die den Wert der zu verwendenden XML-Bindungsquelle zurückgibt, oder legt diese fest.⁵⁰

⁴⁵<http://www.itwissen.info/definition/lexikon/SCP-secure-copy-protocol-SCP-Protokoll.html>. Abgerufen am 30.09.14

⁴⁶<http://msdn.microsoft.com/de-de/library/system.windows.controls.usercontrol.aspx>. Abgerufen am 24.10.14

⁴⁷<http://www.haw-hamburg.de/vpn.html>. Abgerufen am 30.09.14

⁴⁸<http://de.wikipedia.org/wiki/WebDAV>. Abgerufen am 30.09.14

⁴⁹<http://msdn.microsoft.com/de-de/library/system.windows.window.aspx>. Abgerufen am 24.10.14

⁵⁰<http://msdn.microsoft.com/de-de/library/system.windows.data.binding.xpath.aspx>. Abgerufen am 24.10.14

Abbildungsverzeichnis

2.1.	Funktionsbaum Prozesskontrolle	5
2.2.	Der camunda BPM-Kreislauf (Freund und Rücker, 2012, S.4)	7
2.3.	camunda BPMN-Framework (Freund und Rücker, 2012, S.15)	8
2.4.	Die Basiselemente der BPMN (Freund und Rücker, 2012, S.21)	10
2.5.	Model-View-Controller-Entwurfsmuster	11
2.6.	Model-View-ViewModel-Entwurfsmuster	12
3.1.	Im IST-Zustand verwendete (gekürzte) Excel-Tabelle zur Erfassung der Produktdaten. Jede Zeile entspricht einem Produkt. Die Zahlenwerte sind fiktiv.	16
3.2.	IST-Prozessmodell (BPMN-Modell)	17
3.3.	Kausalkette zur Darstellung der Prozessschwächen und ihrer Ursachen (nach Freund und Rücker (2012, S.142))	19
3.4.	Kontextsicht des Gesamtsystems	24
3.5.	SOLL-Prozessmodell (BPMN-Modell)	26
4.1.	Fachliches Datenmodell für das SL-System	36
4.2.	Produkt-Zustände im SL-System	37
4.3.	Produkt-Zustände im ERP-System	38
4.4.	Use Cases des SL-Systems	39
4.5.	Use Cases des ERP-Systems	47
4.6.	GUI Mockup File	51
4.7.	GUI Mockup FTP	52
4.8.	GUI Mockup Options	52
4.9.	GUI Mockup Produkt Auswahl	53
4.10.	GUI Mockup Formular Auswahl	53
5.1.	Verteilungsdiagramm	55
5.2.	Client-Server Architektur (Tanenbaum und van Steen, 2007)	56
5.3.	Drei Phasen der optimistische Synchronisationsverfahren	58

5.4.	Ordnerstruktur für FTP-Server Variante 1	60
5.5.	Ordnerstruktur für FTP-Server Variante 2	61
5.6.	Bausteinsicht des ERP-System	64
5.7.	Laufzeitsicht des ERP-System	65
5.8.	Bausteinsicht des SL-System	73
5.9.	Innensicht für das SL ProductMonitoring	74
5.10.	Laufzeitsicht: Produkt auf dem FTP-Server auswählen.	75
5.11.	Laufzeitsicht: Produkt vom FTP-Server herunterladen.	77
5.12.	Laufzeitsicht: Produkt an FTP-Server senden.	79
5.13.	GUI-Design mit verschiedene geöffnete Formulare	80
A.1.	Einfache IST-Prozessmodell (BPMN-Modell)	104

Tabellenverzeichnis

4.1. Anwendungsfall „Formulare aktualisieren“	40
4.2. Anwendungsfall „Formulare Auswahl“	41
4.3. Anwendungsfall „Produkt Auswahl“	42
4.4. Anwendungsfall „Produkt erfassen/prüfen“	43
4.5. Anwendungsfall „Produkt übermittelt“	44
4.6. Anwendungsfall „Produkt freigeben“	45
4.7. Anwendungsfall „Produkt freigabe entfernen“	46
4.8. Anwendungsfall „Produkt übermitteln an FTP“	48
4.9. Anwendungsfall „Produkt übermitteln an ERP“	49
4.10. Anwendungsfall „Produkt erhalten“	50
6.1. Umsetzung der funktionalen Anforderungen. Legende: FA - Funktionale Anforderung, U - Umgesetzt, NU - Nicht umgesetzt, F - Fehlt (nicht Teil der Arbeit), A - Anmerkung	86
6.2. Umsetzung der nichtfunktionalen Anforderungen. Legende: NFA - Nichtfunktionale Anforderung, U - Umgesetzt, NU - Nicht umgesetzt, F - Fehlt (nicht Teil der Arbeit), A - Anmerkung	86
6.3. Umsetzung der nichtfunktionalen Anforderungen. Legende: NFA - Nichtfunktionale Anforderung, U - Umgesetzt, NU - Nicht umgesetzt, F - Fehlt (nicht Teil der Arbeit), A - Anmerkung	87

Auflistungsverzeichnis

5.1. Grundlegende XML-Struktur eines Produktes.	66
5.2. Zwei Baumstrukturen eines Produktes, um den letzten und den aktuellen Informationsstand zu erfassen.	68
5.3. Der letzte Informationsstand für ein XML-Element als Attribute.	68
5.4. Summenformel in einem XML-Element.	69
5.5. Beispielhafte Formular View, um XML Inhalte darzustellen.	70

Literaturverzeichnis

- [Alexander (DevExpress Support) 2013] SUPPORT), Alexander (: *DevExpress MVVM Framework. Introduction to Services, DXMessageBoxService and DialogService*. 2013. – URL <https://community.devexpress.com/blogs/wpf/archive/2013/09/30/devexpress-mvvm-framework-introduction-to-services-dxmessageboxservice-and-dialogservice.aspx>. – Zugriffsdatum: 31.08.2014
- [Balzert 2011] BALZERT, Helmut: *Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb, 3. Aufl. [Online-Ausg.]*. Heidelberg : Spektrum Akademischer Verlag, 2011. – ISBN 978-3-8274-1706-0
- [Bauernhansl 2014] BAUERNHANSL, Thomas: *Industrie 4.0 in Produktion, Automatisierung und Logistik : Anwendung · Technologien · Migration*. Wiesbaden : Springer Fachmedien Wiesbaden, 2014. – ISBN 978-3-658-04682-8
- [Bernardi und iGreen Partnern 2014] BERNARDI, Ansgar; iGreen P.: *Intelligente Wissenstechnologien für das öffentlichprivate Wissensmanagement im Agrarbereich*. 2014. – URL http://www.igreen-projekt.de/iGreen/fileadmin/Download/iGreen_Schlussbericht_Verbund_final.pdf. – Zugriffsdatum: 31.08.2014
- [Coulouris, Dollimore und Kindberg 2002] COULOURIS, Jean ; KINDBERG T.: *Verteilte Systeme, Konzepte und Design*. Pearson Education Deutschland, 2002. – ISBN 978-3-827-37186-7
- [Feldman und Daymon 2008] FELDMAN, Maxx: *WPF in Action with Visual Studio 2008*. 2008. – ISBN 1-933988-22-3
- [Ferstl, Bodendorf, Bartmann und Sinz 2011] FERSTL, Freimut; BARTMANN Dieter; SINZ Elmar J.: *Dienstorientierte IT-Systeme für hochflexible Geschäftsprozesse*. University of Bamberg Press 2011, 2011. – ISBN 978-3-86309-009-8

- [Freund und Rücker 2012] FREUND, Bernd: *Praxishandbuch BPMN 2.0. 2., aktualisierte Aufl.* München [u.a.] : Hanser, 2012. – ISBN 978-3-446-42986-4
- [Hackl 2004] HACKL, Peter: *Prozesskontrolle.* 2004
- [Hausladen 2014] HAUSLADEN, Iris: *IT-gestützte Logistik : Systeme - Prozesse - Anwendungen.* Wiesbaden : Gabler Verlag/ Springer Fachmedien Wiesbaden GmbH, Wiesbaden, 2014. – ISBN 978-3-8349-4665-2
- [Hueber 2012] HUEBER, Markus: *Prozess- und Projektmanagement für ITIL : Nutzen Sie ITIL optimal 1. Aufl., 2011. [Online-Ausg.].* Wiesbaden : Vieweg+Teubner Verlag / Springer Fachmedien Wiesbaden GmbH, Wiesbaden, 2012. – ISBN 978-3-8348-1482-1
- [Jobst 2010] JOBST, Daniel: *Service- und Ereignisorientierung im Contact-Center : Entwicklung eines Referenzmodells zur Prozessautomatisierung[Online-Ausg.].* Wiesbaden : Gabler Verlag / Springer Fachmedien Wiesbaden GmbH, 2010. – ISBN 978-3-8349-8989-5
- [Kersken 2011] KERSKEN, Sascha: *IT-Handbuch für Fachinformatiker. Der Ausbildungsbegleiter. 5. Aufl.* Galileo Press, 2011. – ISBN 978-3-8362-1744-6
- [Kühnel 2012] KÜHNEL, Andreas: *Galileo Computing Visual C# 2012.* 2012. – URL http://openbook.galileocomputing.de/visual_csharp_2012/. – Zugriffsdatum: 31.08.2014
- [Kriha und Schmitz] KRIHA, Roland: *Sichere Systeme: Konzepte, Architekturen und Frameworks*
- [Microsoft-MVC 2003] MICROSOFT: *Model-View-Controller.* URL <http://msdn.microsoft.com/en-us/library/ff649643.aspx>. – Zugriffsdatum: 1.10.2014, 2003
- [Microsoft-MVVM 2014] MICROSOFT: *Implementing the MVVM Pattern.* URL <http://msdn.microsoft.com/en-us/library/gg405484.aspx>. – Zugriffsdatum: 1.10.2014, 2014
- [OMG 2011] OMG: *Business Process Model and Notation.* 2011. – URL <http://www.omg.org/spec/BPMN/2.0/PDF>. – Zugriffsdatum: 31.08.2014
- [RIGGERT] RIGGERT, Wolfgang: *EDI Electronic Data Interchange.* – URL <http://www.wi.fh-flensburg.de/fileadmin/dozenten/Riggert/bildmaterial/Dokumentenmanagement/2-Capture-EDI.pdf>. – Zugriffsdatum: 31.08.2014

- [Schmidt 2009] SCHMIDT, Götz: *EUROPEAN ASSOCIATION OF BPM: Common Body of Knowledge for BPM*. Wettenberg : Schmidt, 2009. – ISBN 978-3-921-31380-0
- [Schmitz 2014] SCHMITZ, Matthias: *Strategie für die Portierung von Desktop-Business-Anwendungen auf iOS-gestützte Endgeräte*. Wiesbaden : Gabler Verlag / Springer Fachmedien Wiesbaden GmbH, 2014. – ISBN 978-3-658-04769-6
- [Schneider und Romano 2012] SCHNEIDER, Roger: *ICTGeschäftsprozessunterstützung und Akzeptanzförderung : Grundlagen zur Prozessoptimierung und Veränderungsbegleitung mit Beispielen, Fragen und Antworten. 1.* 2012. – URL <http://books.google.de/books?id=eOwXg3BASHYC>. – Zugriffsdatum: 31.08.2014
- [Starke 2014] STARKE, Gernot: *Effektive Softwarearchitekturen : Ein praktischer Leitfaden. 6., überarbeitete Auflage*. München : Hanser, 2014. – ISBN 978-3-446-43614-5
- [Starke-Beispiel 2014] STARKE, Gernot: *Beispiele für Qualitätsanforderungen an Software*. 2014. – URL <https://bitbucket.org/arc42/quality-requirements>. – Zugriffsdatum: 1.10.2014
- [Tanenbaum und van Steen 2007] TANENBAUM, Maarten: *Verteilte Systeme Prinzipien und Paradigmen. 2. Auflage*. Pearson, 2007. – ISBN 978-3-8273-7293-2
- [Technologies 2013] TECHNOLOGIES, South R.: *What's the Difference? FTP, SFTP, and FTP/S*. 2013. – URL <http://www.webdrive.com/whats-difference-ftp-sftp-ftps>. – Zugriffsdatum: 31.08.2014
- [Wegener 2013] WEGENER, Jörg: *WPF 4.5 und XAML : grafische Benutzeroberflächen für Windows inkl. Entwicklung von Windows Store Apps*. München : Hanser, 2013. – ISBN 978-3-446-43467-7
- [Weitzel, Harder und Buxmann 2001] WEITZEL, Thomas; BUXMANN P.: *Electronic Business und EDI mit XML*. Heidelberg, 2001. – ISBN 978-3-932588-98-3
- [WISE-FTP 2014] WISE-FTP: *Erläuterung des FTP- und SFTP-Protokolls*. 2014. – URL http://www.wise-ftp.de/know-how/ftp_und_sftp_protokoll.htm. – Zugriffsdatum: 31.08.2014

A. Anhang

A.1. Einfache IST-Prozessmodell

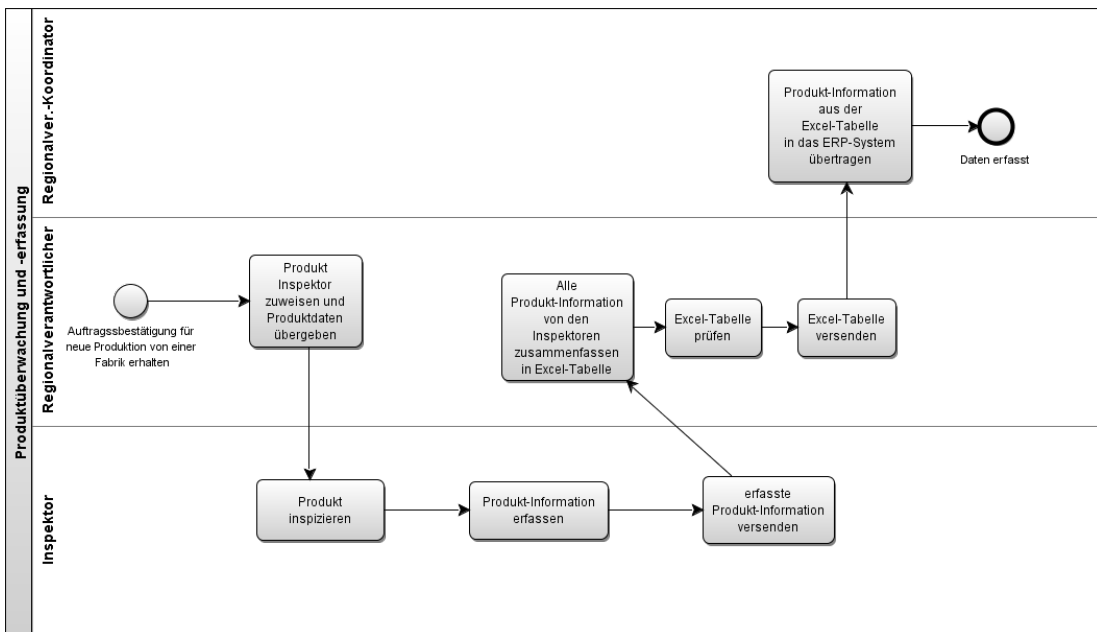


Abbildung A.1.: Einfache IST-Prozessmodell (BPMN-Modell)

A.2. Mögliche fehlerhaften Eingaben

- Output - Der aktuelle Stand der Output ist weniger als der letzte Stand
- Output - Die Eingabe ist mehr, als Produziert werden kann.

A.3. Formulare

A.3.1. General Information

Beschreibt die Stammdaten und wird vom ERP-System vorgegeben. Enthält Informationen (in Englisch):

- Sytle
- Knit/Woven
- BC-Merch
- Agent Merch
- Factory
- Total Lines of the Factory
- BC QC
- Order Placement
- Order Qty.
- Packs
- Order Qty. in pcs

A.3.2. Production and FRI status

Beschreibt die Produktioninspektion und muss vom Inspektor erfasst werden. Enthält Informationen (in Englisch):

- Production Report Date
- Cut

- Output
- Output Balance With 5% Extra - Berechnet durch $Order\ Qty.\ in\ pcs * 1.05 - Output$
- Delivery Date:Ex Ctg Date
- DDP
- FRI Done Qty.
- Balance Qty. - Berechnet durch $Order\ Qty. - FRI\ Done\ Qty.$

A.3.3. Forecast and Remarks

Beschreibt wie lange noch Zeit ist bis das Produkt verschifft wird. Muss vom Inspektor erfasst werden. Enthält Informationen (in Englisch):

- Days Left to ship - Berechnet durch $Delivery\ Date:Ex\ Ctg\ Date - Production\ Report\ Date$
- Running Lines
- Productivity/Line/Day
- Days required to Stitch Balance Qnty - Berechnet durch $Output\ Balance\ With\ 5\% \ Extra / (Running\ Lines * Productivity/Line/Day)$
- Remarks

A.4. Deutsch - Englisch

bearbeitet - edited

Formulare - Reports

freigegeben - approved

neu - new

Zustand - Status

Fabrik - Factory

Produkt - Product

Produktstand - Productstatus

Formulare aktualisieren - Refresh Reports

Formular Auswahl - Select Report

Produkt Auswahl - Select Product

Produkt übermitteln - Send Product

Produkt freigeben - Set Product approved

Produkt Freigabe entfernen - Remove Product approved

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 03.11.2014

Aria Rafi Nazari