

**Gottfried Wilhelm
Leibniz Universität Hannover
Fakultät für Elektrotechnik und Informatik
Institut für Praktische Informatik
Fachgebiet Software Engineering**

**Verbesserung der Werkzeugunterstützung für die
Verwaltung von Produktlinien-Informationen in
einem Automotive-Unternehmen**

Masterarbeit

im Studiengang Informatik

von

Lars Rumpf

Prüfer: Prof. Dr. Kurt Schneider

Zweitprüfer: Dr.-Ing. Daniel Lübke

Betreuer: M. Sc. Wasja Brunotte, M.Sc. Melanie Busch

Hannover, 30. Juli 2019

Erklärung der Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbständig und ohne fremde Hilfe verfasst und keine anderen als die in der Masterarbeit angegebenen Quellen und Hilfsmittel verwendet habe.

Die Masterarbeit hat in gleicher oder ähnlicher Form noch keinem anderen Prüfer und/oder Prüfungsamt vorgelegen.

Mit der Übermittlung meiner Arbeit an externe Dienste zur Plagiatsprüfung durch Plagiatssoftware erkläre ich mich einverstanden.

Hannover, den 30.07.2019

Vorname, Nachname

Zusammenfassung

Excel Tabellen werden in der Industrie häufig für das Speichern und Verwalten von Informationen genutzt. Dateizugriffe erfolgen oftmals zeitgleich und aufgrund komplexer Prozesszusammenhänge gibt es in vielen Fällen einen hohen Grad an Informationsredundanz zwischen Tabellen. Diese Arbeit zeigt anhand eines realen Beispiels aus der Industrie eine Möglichkeit auf, wie sich ein Prozess für die Verwaltung von Produktlinien-Informationen basierend auf einer Excel Tabelle durch eine geeignete Softwarelösung verbessern lässt. Dazu wurde zuerst der zugrundeliegende Prozess durch Interviews analysiert. Nachfolgend wurden Anforderungen definiert und sortiert. Basierend auf den Anforderungen wurde ein Anwendungskonzept erstellt und anschließend durch einen Web-basierten Softwareprototyp implementiert. Dieser wurde darauffolgend hinsichtlich der Nutzbarkeit und Akzeptanz überprüft. Die Ergebnisse zeigten, dass sich die Mitarbeiter eine unterstützende Anwendung wünschen und diese akzeptieren. Es konnte ebenfalls gezeigt werden, dass in den meisten Fällen die benötigte Arbeitszeit reduziert werden konnte und sich gleichzeitig die Benutzbarkeit verbesserte.

Abstract

Excel spreadsheets are widely used in the industry for storing and managing information. Files were accessed at the same time and due to complex process relationships. There is also a chance of information redundancy between different Spreadsheets. This paper shows, based on a real-world example from the industry, how to improve a process for managing product line information based on an Excel spreadsheet with an appropriate software solution. The first step was an analysis of the underlying process through interviews. Thereafter requirements were defined and sorted. An application concept was created based on the defined requirements. Subsequently a web-based software prototype was implemented. Afterwards the prototype was evaluated for usability and acceptance. The results showed that employees need a supportive application and an acceptance is available. The results also shown that in most cases the needed working hours could be reduced and at the same time the usability was improved.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation.....	1
1.2	Zielsetzung.....	2
1.3	Struktur der Arbeit.....	2
2	Grundlagen	3
2.1	BPMN.....	3
2.2	Usability	4
2.3	Relationale Datenbanken.....	5
2.4	NoSQL Datenbanken.....	5
2.4.1	Document Stores	6
2.4.2	Graphdatenbanken	7
3	Prozessanalyse.....	9
3.1	Allgemeine Prozessübersicht.....	10
3.2	Aufbau der Variantenmatrix	13
3.3	Allgemeine Herausforderungen	18
3.4	Verarbeitung von Kundenspezifikationen.....	20
3.4.1	Überblick und Analyse	20
3.4.2	Erkannte Probleme	23
3.5	Variantenverwaltung	25
3.5.1	Überblick und Analyse	25
3.5.2	Erkannte Probleme	29
3.6	Projektmanager.....	30
3.6.1	Überblick und Analyse	30
3.7	Zertifizierung	31
3.7.1	Überblick und Analyse	31
3.7.2	Erkannte Probleme	32
3.8	Konfigurationsmanagement	33
3.8.1	Überblick und Analyse	33
3.9	Zusammenfassung der Prozessanalyse	34
4	Anforderungsanalyse.....	37
4.1	Funktionale Produkthanforderungen.....	37
4.2	Nichtfunktionale Produkthanforderungen	46

4.3	Sicherheit.....	48
4.4	Einschränkung von Anforderungen.....	50
4.4.1	Relevante und sortierte Anforderungen an einen Prototyp.....	50
4.4.2	Exemplarische Use Cases	51
5	Konzeptentwicklung	55
5.1	Informationsflussmodell	55
5.2	Visuelles Konzept	57
5.3	Architektur.....	60
5.3.1	Web Plattform Architektur	60
5.3.2	Architektur der Webanwendung	61
5.3.3	Datenbanksysteme.....	62
5.4	Vergleich zu existierender Lösung.....	64
6	Implementation	65
6.1	Webanwendung.....	65
6.1.1	Redux.....	69
6.2	Wahl der Datenbank und heuristische Regeln.....	71
6.2.1	Heuristische Regeln	71
6.2.2	Wahl der Datenbank.....	72
6.2.3	Tools und Kommunikation	72
6.2.4	Sicherheit	73
6.2.5	Datentransformation.....	73
7	Evaluation.....	75
7.1	Studiendesign	75
7.1.1	Ziele der Studie	75
7.1.2	Evaluationsumgebung.....	75
7.1.3	Zu bearbeitende Aufgaben	76
7.1.4	Metriken	77
7.1.5	System Usability Scale Fragebogen.....	78
7.1.6	Hypothesen	78
7.1.7	Studiensubjekte	79
7.2	Auswertung/Ergebnisse der Evaluation	80
7.2.1	Beobachtungen und Äußerungen während der Studie.....	80
7.2.2	Auswertung der Bearbeitungszeiten.....	82
7.2.3	Auswertung des System Usability Scores	86

7.2.4	Validität	87
7.3	Zusätzliche Erweiterungen an die Anwendung	88
8	Verwandte Arbeiten	91
9	Fazit und Ausblick	93
9.1	Fazit	93
9.2	Ausblick	94
9.2.1	Variantenmatrix und Anwendung	94
9.2.2	Der Prozess	97
9.2.3	Studie und Tests	98
	Verzeichnisse	101
	Quellen- und Literaturverzeichnis	101
	Tabellen- und Abbildungsverzeichnis	104
	Anhang	107
	Einverständniserklärung und Fragebogen	107
	Ergebnisdaten	111
	Inhalt der beiliegenden CD	112
	Danksagungen	113

1 Einleitung

1.1 Motivation

Das weltweite Wachstum der Automobilindustrie scheint ununterbrochen zu sein¹ und Infotainment Systeme spielen eine immer zentralere Rolle als Schnittstelle zu den Passagieren. Sie müssen sich den gegebenen Märkten und Gesetzen anpassen sowie immer neuere Technologien integrieren. Neben einer Radio- und Navigationsfunktion bieten die Systeme oftmals Zugriff auf Informationen des Fahrzeugs und ermöglichen das Einstellen vieler Komfortfunktionen. Das Unternehmen Robert Bosch Car Multimedia GmbH in Hildesheim entwickelt und produziert Infotainment Systeme für mehrere Automobilhersteller. Dabei werden Produktlinien eingesetzt, wodurch auf Basis weniger Plattformen unterschiedliche Varianten kostengünstig realisiert werden können. Diese werden in komplexen, miteinander vernetzten Excel Tabellen verwaltet und organisiert. Ein Großteil der Informationen wird in einer sog. Variantenmatrix festgehalten. Durch die steigende Anzahl nötiger Features und Funktionen sowie neuer Produktlinien und Varianten stößt die Technik an ihre Grenzen. Außerdem können die Excel-basierten Tabellen nicht alle benötigten Abhängigkeiten und Bedingungen der Daten darstellen. Für eine Validität der in den Tabellen gespeicherten Informationen kann ebenfalls nicht ausreichend gesorgt werden, weshalb es bei Fehlern zu hohen Kosten kommen kann. Je größer die von den Mitarbeitern manuell verwalteten Tabellen werden, desto schwieriger wird es für sie, diese zu benutzen. Es muss also dafür gesorgt werden, dass die verteilten Informationen in eine Datenbank überführt, Konsistenzen überprüft und die Daten für die Mitarbeiter einfacher zugänglich gemacht werden.

¹ Quelle: <http://www.oica.net/category/production-statistics/> - Abgerufen am 05.07.2019 um 18:55 Uhr

1.2 Zielsetzung

Ziel der Arbeit ist es, anhand eines realen Beispiels aus der Industrie eine Möglichkeit aufzuzeigen, wie sich der Prozess der Verwaltung von Produktlinien-Informationen durch eine geeignete Softwarelösung verbessern lässt. Durch Befragungen und Interviews sollen der zugrundeliegende Prozess sowie die verwendeten Excel Tabellen dokumentiert und analysiert werden. Es sollen mögliche Heuristiken zur Überprüfung der Informationen erkannt und Daten der Excel Tabellen in eine Datenbank überführt werden. Diese sollen mithilfe einer Anwendung dem Nutzer präsentiert werden. Die Anwendung soll anschließend hinsichtlich der Nutzbarkeit und Akzeptanz überprüft werden.

1.3 Struktur der Arbeit

Das nachfolgende Kapitel 2 erläutert Grundlagen, welche für das Verständnis dieser Arbeit notwendig sind.

Kapitel 3 befasst sich mit der Prozessanalyse eines bestehenden Prozesses, bei dem unter anderem Produktlinien-Informationen mithilfe der Variantenmatrix verwaltet werden.

In Kapitel 4 werden nötige Anforderungen an die Anwendung erhoben.

Kapitel 5 befasst sich mit der Entwicklung eines Anwendungskonzeptes basierend auf den vorherigen Anforderungen. Ebenfalls wird dort ein bestehendes Konzept erläutert.

Kapitel 6 beschreibt die Implementation eines Prototyps. Dabei werden die vorher genannten Konzepte aufgegriffen und Anforderungen umgesetzt. Außerdem werden Entscheidungen hinsichtlich der gewählten Datenbank und heuristischer Regeln getroffen.

In Kapitel 7 wird die Evaluation des entwickelten Prototyps beschrieben. Dieser wurde an mehreren Probanden getestet, um eine Aussage über die Nützlichkeit sowie der Akzeptanz treffen zu können.

Kapitel 8 befasst sich mit ähnlichen Arbeiten.

In Kapitel 9 wird ein abschließendes Fazit gezogen, sowie ein weitreichender Ausblick auf weitere mögliche und nötige Arbeiten sowie Untersuchungen gegeben.

2 Grundlagen

2.1 BPMN

Business Process Model and Notation (BPMN) ist eine grafische Spezifikationssprache im Prozessmanagement. Sie definiert grafische Elemente, mit denen Arbeitsabläufe und Geschäftsprozesse visuell dargestellt und festgehalten werden können. Ziel der Notation ist es, die Prozesse oder deren Fragmente für unterschiedliche Personengruppen wie zum Beispiel den Business Analysten als auch den technischen Entwicklern verständlich darzustellen (Silver, 2011 S. Preface VI, 3).

Die nachfolgende Abbildung 2.1 gibt einen kurzen Überblick über die in dieser Arbeit verwendeten Elemente. Nachfolgend werden diese jeweils beschrieben.

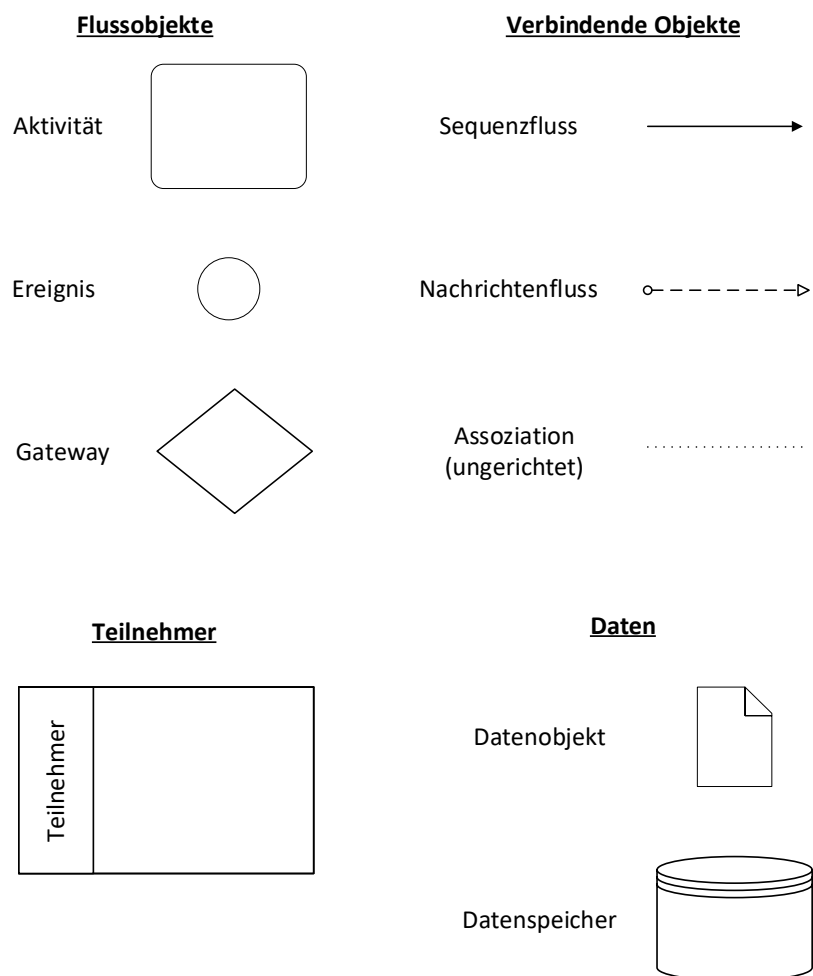


Abbildung 2.1 BPMN Basiselemente

Aktivitäten, Ereignisse und Gateways sind Flussobjekte. Eine Aktivität ist eine auszuführende Arbeit innerhalb eines Prozesses. Eine Aktivität kann entweder eine einzelne Aufgabe oder ein komplexerer Teilprozess sein. Komplexere Teilprozesse besitzen ein „Plus“-Element. Ein Ereignis ist etwas, was während des Prozesses stattfindet. Es beeinflusst den Fluss des Modells und hat einen Auslöser und eine Auswirkung. Es gibt Start-, Zwischen- und Endereignisse. In dieser Arbeit werden jedoch nur die Start- und Endereignisse verwendet. Gateways dienen zur Kontrolle der Divergenz und Konvergenz von Sequenzflüssen. In dieser Arbeit werden lediglich sog. „Exclusive Gateways“ genutzt: Nur einer der ausgehenden Pfade kann gewählt werden. Um Aktivitäten, Ereignisse und Gateways zu verbinden sowie deren Reihenfolge festzulegen werden Sequenzfluss-Objekte genutzt.

Ein Prozess kann aus mehreren *Teilnehmern* (auch Pools genannt) bestehen. Ein Teilnehmer wird ähnlich einer Schwimmbahn grafisch dargestellt. Er besitzt meist einen eigenen Aktivitätenfluss und kommuniziert mit anderen Teilnehmern. *Nachrichtenfluss*-Objekte dienen in diesem Fall der Verbindung mehrerer Teilnehmer.

Datenobjekte stellen Informationen zur Verfügung, die von Aktivitäten erzeugt werden können oder benötigt werden. *Datenspeicher* sind Ansammlungen vieler Daten. *Datenobjekte* werden über *Assoziationen* mit *Aktivitäten* verbunden. *Assoziationen* können ungerichtet oder gerichtet sein. (OMG (Object Management Group), 2011 S. 29ff.)

2.2 Usability

Der Begriff Usability bezeichnet laut ISO-Norm 9421-11 die Gebrauchstauglichkeit oder auch (Be-)Nutzerfreundlichkeit eines Systems: „*Usability ist das Ausmaß, in dem ein Produkt, System oder Dienst durch bestimmte Benutzer in einem bestimmten Anwendungskontext genutzt werden kann, um bestimmte Ziele effektiv, effizient und zufriedenstellend zu erreichen*“. Zusammengefasst ist die Usability ein Maß der Nutzbarkeit, gemessen daran, wie genau und schnell ein Nutzer sein Ziel durch die Nutzung des Produktes erreicht. Eine Anwendung besitzt ein hohes Maß an Usability, wenn die Benutzerschnittstellen selbstbeschreibend, steuerbar,

erwartungskonform, fehlertolerant, individualisierbar, lernförderlich als auch der Aufgabe angemessen sind. Usability betrachtet hier vor allem die (grafische) Oberfläche, das User Interface (UI), einer Anwendung. (Jacobsen, et al., 2017 S. 34f.)

2.3 Relationale Datenbanken

Relationale Datenbanken beruhen auf tabellenbasierten relationalen Datenbankmodellen. Eine Relation, die Grundlage des Datenbankkonzeptes, ist eine Beziehung zwischen Elementen einer Menge. Eine Tabelle mit Relationen besitzt folgende drei Eigenschaften (Codd, 1990 S. 1f.):

- Jede Zeile stellt ein Tupel von Relationen dar.
- Die Reihenfolge von Zeilen spielt keine Rolle.
- Alle Zeilen unterscheiden sich inhaltlich voneinander.

Die Spalten eines relationalen Modells besitzen einen Namen, die durch ein vorher definiertes Datenbankschema festgelegt werden. Das Datenbankschema legt ebenfalls fest, welche Daten in der Datenbank gespeichert werden dürfen und wie diese in Beziehung zu anderen Daten stehen. Ein Datenbankschema reduziert Redundanz, bietet Zugriff auf Informationen, um Datenbankabfragen zu vereinfachen und gewährleistet die Integrität und Richtigkeit der Daten.

Um Daten abzufragen oder sie zu manipulieren, wird vorrangig SQL (Structured Query Language) genutzt, deren Grundlage das relationale Modell ist (Codd, 1990 S. 12).

2.4 NoSQL Datenbanken

NoSQL ist ein etwas unkonventioneller und provokanter Begriff, der darauf aufmerksam machen möchte, dass es sich um ein nicht-relationales Datenmodell handelt. Es wird von der Community auch als „not only SQL“ bezeichnet, um die Abgrenzung zu SQL aufzuweichen, denn es gibt bei relationalen Datenbanksystemen ebenfalls weitere Abfragesprachen, wie es auch bei NoSQL Datenbanksystemen SQL (ähnliche) Abfragesprachen gibt. Nicht-relationale Systeme existieren schon seit vielen Jahrzehnten, erlangten ihren Durchbruch aber erst durch die Verbreitung des Internets. Ziel

von nicht-relationalen Datenbanksystemen ist ein einfacheres Design sowie eine bessere Skalierbarkeit über verteilte Systeme und deren Konsistenzverhalten gegenüber relationalen Datenbanksystemen. Ein NoSQL Datenbanksystem ist oft schemafrei oder hat nur schwächere Schemarestriktionen.

NoSQL Systeme können in mehrere Kategorien eingeordnet werden:

- NoSQL Kernsysteme:
 - Wide Column Stores / Column Families
 - Document Stores
 - Key/Value/Tuple Stores
 - Graphdatenbanken
- Nachgelagerte NoSQL Systeme:
 - Objektdatenbanken
 - XML-Datenbanken
 - Grid-Datenbanken
 - Weitere nicht-relationale Systeme

Ebenfalls gilt zu erwähnen, dass es keine klare Grenze zwischen relationalen Datenbanksystemen und NoSQL Systemen gibt (Edlich, et al., 2010 S. 2ff.).

In den nachfolgenden Abschnitten werden die im Laufe dieser Arbeit erwähnten Document Stores und Graphdatenbanken näher definiert.

2.4.1 Document Stores

Document Stores speichern mit einem eindeutigen Schlüssel zugeordnete semistrukturierte Datensammlungen wie zum Beispiel JSON², BSON³, YAML⁴ oder RDF⁵ Dokumente. Die Datenbank legt oft nur fest, auf welches Datenformat ein Schlüssel zeigt. Somit folgen die Daten selten einem festen Schema (Edlich, et al., 2010 S. 8). Es gibt beim Einfügen oder Editieren keine Normalisierung der Daten, daher können redundante Daten

² JavaScript Object Notation – Datenformat

³ Binary JSON – Datenformat

⁴ Auszeichnungssprache ähnlich wie XML

⁵ Datenformat, das Metadaten enthält

vorkommen. Es können Abfragen anhand bestimmter Eigenschaften der Dokumente erfolgen. So kann zum Beispiel in einem Store „Telefonbuch“ die in Abbildung 2.2 genannten JSON Objekte gespeichert werden.

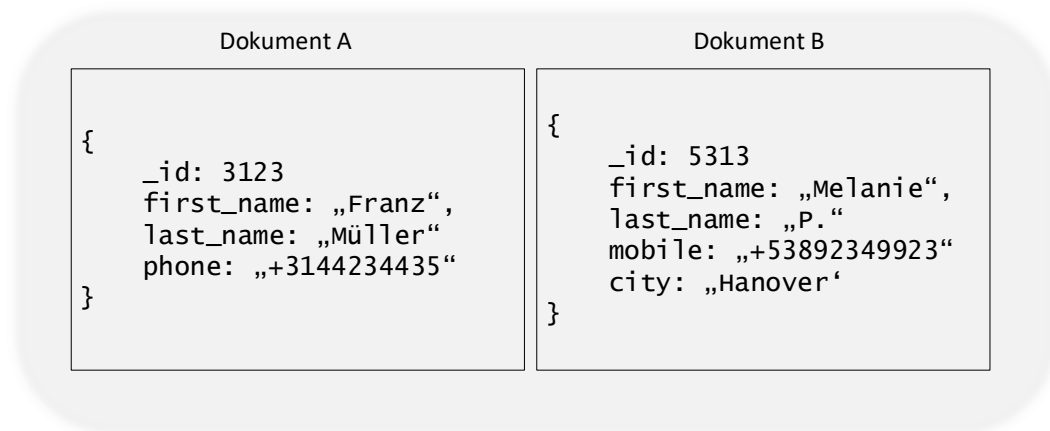


Abbildung 2.2 Document Store Beispiel

Dokument B in Abbildung 2.2 besitzt teilweise andere Attribute gegenüber Dokument A und trotzdem können diese in einem Document Store nebeneinander liegen. Da es in Document Stores kein festes Schema geben muss, kann es zu Redundanzen kommen. Ebenfalls können Integrität sowie Richtigkeit der Daten nicht gewährleistet werden. So kann es vorkommen, dass in Dokument X ein Attribut „color“ genannt wird, in Dokument Y „Color“ und Dokument Z nennt es vielleicht „colour“. Um die Eigenschaften eines Dokuments wiederzufinden müssen die Daten gesondert indexiert werden. Ein Index kann auch nachträglich erstellt werden. Document Stores bieten, wie die meisten No-SQL Datenbanken keine SQL Abfragesprache. CRUD-Operationen (Create, Read, Update, Delete) werden meist durch objektspezifische Methoden der jeweiligen Programmiersprache realisiert.

2.4.2 Graphdatenbanken

Graphdatenbanken nutzen Graphen um Informationen zu speichern und diese darzustellen. Ein Graph besitzt Knoten und Kanten. Gerichtete Kanten verbinden Knoten und können mit Eigenschaften versehen werden. Knoten besitzen Attribute. So können Informationen wie „Alice kennt Bob“ sehr einfach dargestellt und gespeichert werden. In diesem Beispiel ist eine Person ein Knoten und kann beliebige Attribute besitzen (Alter, Geschlecht, etc.). Die Kante ist meist durch ein Verb definiert und setzt die Knoten in

Beziehung zueinander. Graphdatenbanken lassen sich im Gegensatz zu relationalen Datenbanken sehr einfach traversieren (Edlich, et al., 2010 S. 9). Es gibt eine Vielzahl unterschiedlicher Graphdatenbanken. Die höchste Bedeutung haben derzeit native Graphdatenbanken, die sog. Property-Graphen modellieren. Die nachfolgende Abbildung 2.3 zeigt eine mögliche graphische Darstellung einer Property-Graph Datenbank.

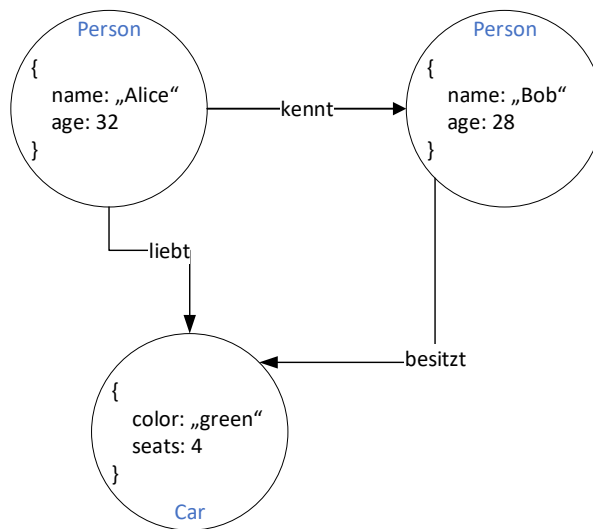


Abbildung 2.3 Graphdatenbank Beispiel

Graphdatenbanken können ein festes Schema besitzen, müssen dies aber nicht zwingend. Daher kann es ebenfalls zu Redundanzen kommen. Es kann ebenfalls keine Integrität sowie Richtigkeit der Daten gewährleistet werden (Edlich, et al., 2010). Es gibt keine standardisierte Abfragesprache, aber unter anderem eine SQL-ähnliche (GraphQL).

3 Prozessanalyse

In diesem Kapitel wird der Prozess zur Variantenerstellung und Variantenverwaltung bei Bosch erläutert und analysiert. Ziel ist es, die Einsatzdomäne einer neuen Software detaillierter zu beleuchten um somit genauere Anforderungen an diese zu erlangen.

Es wird im ersten Abschnitt dieses Kapitels ein kurzer Überblick über den aktuellen Prozess der Variantenerstellung bei Bosch gegeben. Der Prozess wird genauer erläutert und ein Überblick über die einzelnen Prozessabschnitte aufgezeigt. Danach werden einige allgemeine Probleme genannt und der aktuelle Aufbau der Variantenmatrix erklärt.

Es folgen weitere Abschnitte, in denen genauer auf Teilbereiche innerhalb des Prozesses eingegangen wird. Erforderliche Arbeitsabläufe der Mitarbeiter werden beschrieben sowie weitere rollenspezifische Probleme und mögliche Fehlerquellen oder Problemstellen innerhalb des Prozesses genannt. Dazu wurden mehrere verantwortliche Mitarbeiter interviewt. Die Interviews wurden vorwiegend in Besprechungsräumen durchgeführt und immer schriftlich protokolliert. Ein Interview dauerte mindestens 25 Minuten, aber nicht länger als zwei Stunden. Interviews wurden in Deutsch oder Englisch durchgeführt. Einige Personen wurden mehrfach interviewt, da der vorgesehene Zeitrahmen nicht ausreichte. Die Mitarbeiter sollten während des Interviews die Excel-basierte Variantenmatrix benutzen und ihre häufigsten Anwendungsfälle aufzeigen. Ziel der Interviews war es, den gesamten Prozess genauer zu verstehen sowie Abhängigkeiten der Variantenmatrix zu finden.

Bei der Analyse des nachfolgenden Prozesses kommt es an einigen Stellen zu komplexeren Abhängigkeiten mit anderen Prozessen. Dies wird an den jeweiligen Stellen in den Erklärungen verdeutlicht.

In diesem und den darauffolgenden Kapiteln sind Kunde und Automobilhersteller gleichbedeutend.

In diesem Kapitel werden Prozesse mit Hilfe von BPMN visualisiert. Diese grafische Spezifikationssprache stellt einfache Symbole zur Darstellung komplexer Geschäfts- und Prozessabläufe zur Verfügung (Silver, 2011).

3.1 Allgemeine Prozessübersicht

Im nachfolgenden Prozess geht es um die Verwaltung von Infotainment-Systemen und deren Produktlinien-Informationen, ausgehend von der Bestellung durch den Automobilhersteller, bis hin zum Start der Produktion. Die nachfolgende Abbildung 3.1 zeigt den Prozess auf sehr stark abstrahierter Ebene und ähnelt einem Prozessfluss mit wenigen internen sowie externen Abhängigkeiten.

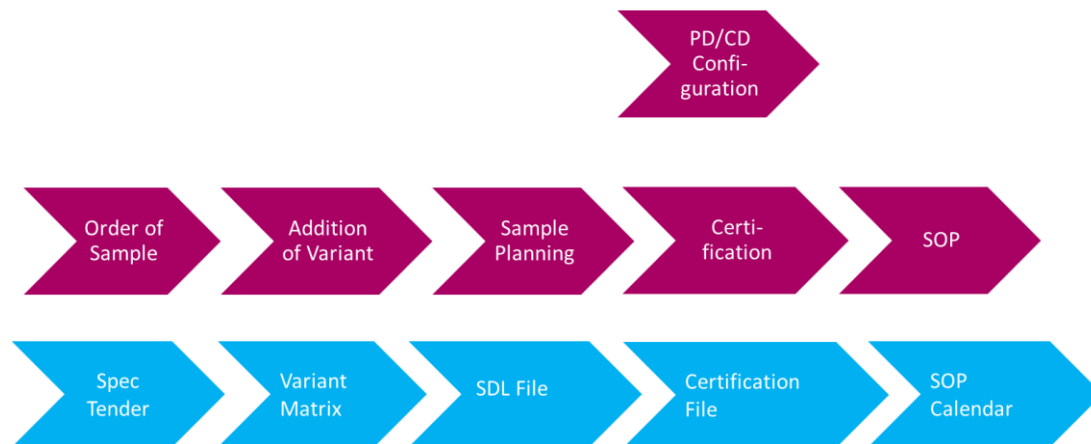


Abbildung 3.1 Prozessübersicht - Grafik leicht überarbeitet. Quelle: Robert Bosch Car Multimedia GmbH – Hildesheim (2019)

Der Prozess ist unterteilt in Aufgaben (violett) und die dazu benötigten Dateien (blau). Er beginnt mit einer sogenannten *SpecTender* Datei. Dies ist eine gewünschte Spezifikation des Automobilherstellers. Die Datei beinhaltet wichtige Angaben zu den neuen gewünschten Varianten: Hardwareinformationen, Anschlussmöglichkeiten, Teilenummer des Herstellers, Bestimmungsland und viele weitere Informationen. Sie wird bei der Erstellung eines neuen Auftrags (*Order of Sample*) zwingend mitgegeben.

Die Spezifikation wird bearbeitet und im nachfolgenden Schritt das zu produzierende Modell in eine *Variantenmatrix* geschrieben (*Addition of Variant*). Dort wird ein Großteil der Hardwareinformationen aufgeschlüsselt. Eine genauere Beschreibung der Variantenmatrix findet in Abschnitt 3.5 statt.

Nachfolgend werden Testprodukte geplant und gebaut (*Sample Planning*). Zum einen muss die Hardware mit Hilfe der Testsamples getestet werden, zum anderen werden die Produktion und deren Abläufe während des Prozessabschnittes geprüft. Die aktuellen Produktlinien und benötigten

Samples in dieser Phase werden in einer *Sample Distribution List (SDL)* Datei festgehalten.

Während und nach der Testproduktion werden die Infotainment-Systeme für die jeweiligen Länder zertifiziert. Ohne diese Zertifizierung darf ein Gerät nicht auf den Markt eingeführt werden. Der Status der Zertifizierungen wird in einer anderen speziellen *Certification* Datei gespeichert.

Die (Software-)Konfiguration (*Production Diagnostic - PD / Customer Diagnostic - CD*) findet während des kompletten Prozesses statt, weshalb die *PD/CD Configuration* außerhalb platziert ist. Während dieses Prozesses werden Dateien erstellt, welche wichtige Software-Eigenschaften des Gerätes bestimmen. Einige davon sind fest im Gerät hinterlegt (Production Diagnostics - PD) und können nachträglich nicht geändert werden, andere hingegen (Customer Diagnostics - CD) können vom Kunden über ein OBD⁶ Gerät verändert werden.

Am Ende des gesamten Prozesses steht der Start der Produktion (*Start of Production – SOP*). Es gibt ebenfalls für diesen Abschnitt eine eigene Datei, einen *SOP Calendar*, welche wichtige Termine beinhaltet.

Der gesamte Prozess findet in einem globalen Kontext statt. Dabei werden Teile des Prozesses von Personen aus anderen Kontinenten ausgeführt und gesteuert.

Wie schon zuvor erwähnt, gibt diese Abbildung nur einen sehr groben Überblick. In der nachfolgenden Abbildung 3.2 wird der Prozess genauer dargestellt. Es ist erkennbar, dass die Variantenmatrix, die im Rahmen dieser Arbeit hauptsächlich betrachtet wird, eine zentrale Rolle innerhalb des gesamten Prozesses spielt. Sie wird von mehreren Teilprozessen und damit auch vielen verschiedenen Mitarbeitern genutzt.

Ebenfalls ist zu erkennen, dass Beziehungen zwischen den Dateien und Prozessabschnitten sehr viel komplexer sind. Hinzugekommen sind einige wichtige verwendete oder generierte Daten und Objekte.

⁶ On-Board-Diagnose: Fahrzeugdiagnosesystem

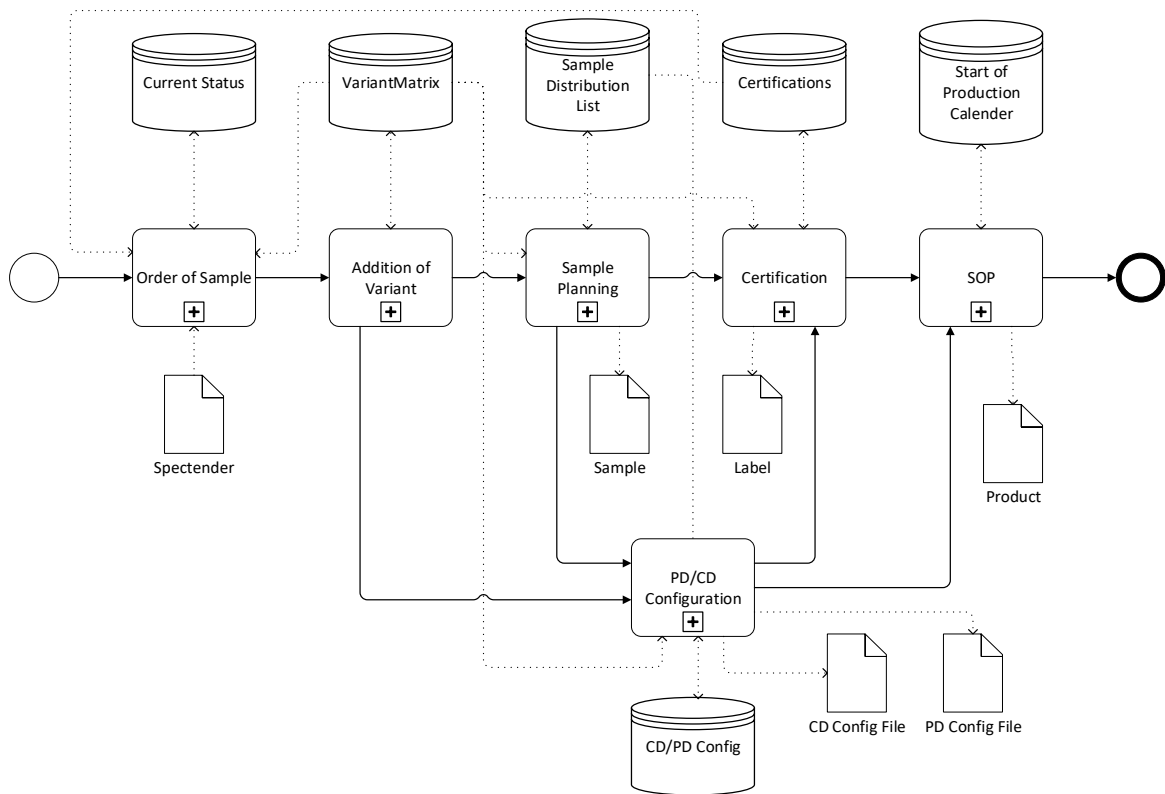


Abbildung 3.2 Erweiterter Prozessablauf

Die Kundenspezifikation (*Spectender*) ist ein Dokument, das vom Fahrzeughersteller kommt. Diese wird in dem ersten Schritt (*Order of Sample*) von einem Mitarbeiter geprüft. Hierbei werden Daten aus der *Variantenmatrix* sowie aus der Liste der Zertifizierungen (*Certifications*) hinzugezogen. Die Daten der Spezifikation werden bei Annahme dieser in eine weitere Datei über den aktuellen Status (*Current Status*) gespeichert. Hier finden sich ebenfalls Daten zum Stand der Zertifizierungen sowie ein Index aus der Variantenmatrix wieder. Die Verarbeitung der Kundenspezifikation wird in Abschnitt 3.4 näher betrachtet.

Im nächsten Schritt (*Addition of Variant*) wird die neue oder geänderte Variante in die *Variantenmatrix* übertragen sowie die erforderlichen Hardwareinformationen festgehalten. Danach wird eine Hardware ID berechnet. Dies wird von Hardware-Experten durchgeführt. Weitere Informationen zu der Variantenverwaltung durch die Hardware-Experten sind in Abschnitt 3.5 zu finden.

Für die Musterplanung (*Sample Planning*), als auch für die Zertifizierung (*Certification*), wird ebenfalls auf die Variante aus der Variantenmatrix zugegriffen. Am Ende entstehen ein oder mehrere Testmuster (*Sample*) sowie

ein *Label*. Die Muster und der Bedarf werden in der *Sample Distribution List (SDL)* durch die Projektmanager verwaltet. Nähere Informationen hierzu sind in Abschnitt 3.6 zu finden. Das *Label* ist ein produktbezogener Aufkleber, auf dem die Zertifizierungslogos und andere wichtige Informationen zu Homologationen sichtbar sind. Zuständig hierfür sind die Zertifizierungsstelle und deren Mitarbeiter. Eine genauere Übersicht zeigt Abschnitt 3.7. Analog zur Abbildung 3.1 auf Seite 10 ist die (Software-)Konfiguration (*PD/CD Configuration*) außerhalb des Hauptstranges abgebildet. Erstellt und verwaltet werden die Konfigurationen vom Konfigurationsmanager, siehe Abschnitt 3.8. Es fließen Informationen aus der Variantenmatrix und der Musterplanung ein. Am Ende entstehen unter anderem zwei Dateien (*CD / PD Config File*) und Informationen, welche an die Zertifizierung und an den SOP weitergeleitet werden.

Der letzte Abschnitt im Prozess (*SOP*) stellt eher einen Termin dar. Ab diesem ist das Produkt fertigungsreif. Diese Termine werden in einem *Start of Production Calendar* gespeichert und verwaltet. Zuständig sind hier vor allem die Projektmanager. Sie haben ebenfalls die Kontrolle, welche Spezifikationen am Anfang vom Kunden angenommen werden und stehen dem Kunden sehr nahe. Das Ergebnis ist ein fertiges Produkt, wie es an den Fahrzeughersteller ausgeliefert und in Serie produziert werden kann.

3.2 Aufbau der Variantenmatrix

Dieser Abschnitt beschreibt in kurzer Form den Aufbau der Variantenmatrix. Diese ist in ihrer aktuellen Form (Stand Februar 2019) über 2100 Zeilen und über 230 Spalten groß.

Die nachfolgenden Abbildungen zeigen einige Spalten der Variantenmatrix inklusiv nachgebildeter Einträge, um einen kurzen Überblick zu geben. Aufgrund der Größe der Variantenmatrix wird in dieser Arbeit nicht genauer auf jede einzelne Spalte eingegangen. Außerdem werden nicht alle Spalten in den Abbildungen gezeigt.

Genauere Informationen über den Aufbau der Variantenmatrix gibt die Bachelorarbeit von Paul Glewe (2019 S. 8ff.).

Index	Bosch PN	Customer PN	Customer ID	Scope	Region
CONFIG ENTRY					
323.-.A	0	Config Entry	Config Entry	SC 2.0	USA
456.432.Z	0	Config Entry	Config Entry	SC 2.1	GBR/EUR
323.104.J	0	Config Entry	Config Entry	SC 2.1	JPN
Carline 1					
456.432.Z	7231259834	72945 123FG	J7	SC 2.1	GBR/EUR
Carline 2					
323.-.A	2839234230-12	5642A 7832J	-	SC 2.0	USA

Abbildung 3.3 Variantenmatrix: Index, Teilenummer, Scope, Region

Die ersten ca. 380 Zeilen der Variantenmatrix sind mit sogenannten Konfigurationseinträgen (*Config Entry*) gefüllt. Dies sind alle möglichen Hardwarekonfigurationen, die aktuell vom Automobilhersteller gewählt werden können. Sie entsprechen somit einem „Bestellkatalog“. Die darauffolgenden Zeilen stellen konfigurierte und / oder produktionsreife Varianten dar. Diese sollten immer eine Teilenummer (*Bosch PN*) besitzen. Die konfigurierten Varianten sind wiederum unterschiedlichen Automobil-Produktlinien (*Carline*) zugeordnet, getrennt durch eine blaue Linie, sichtbar in Abbildung 3.3. Jeder Konfigurationseintrag besitzt einen *Index*. Dieser *Index* wird ebenfalls bei den implementierten Varianten angegeben, so ist bei der Nutzung des Filters nach dem *Index* sichtbar, ob diese schon einmal gebaut wurde.

Jede Bosch-Teilenummer (*Bosch PN*) ist einzigartig. Zu jeder Variante gibt es eine zusätzliche Teilenummer des Automobilherstellers (*Customer PN*). Diese kann ebenfalls aufgrund von Änderungen seitens des Automobilherstellers unterschiedlich für denselben Index sein. Es kann aufgrund des Baues von Testgeräten zu einem *Index* mehrere Teilenummern und evtl. auch mehrere Automobilherstellerteilenummern geben. Zu jeder Automobilherstellerteilenummer gibt es mindestens eine Teilenummer. Ein Index kann mehrere Teilenummern besitzen oder gar keine (wenn die Variante noch nicht gebaut wurde). Daneben kann eine Variante zusätzlich eine Kunden ID (*Customer ID*) haben.

Jede Variante besitzt einen *Scope*. Dieser sagt aus, welche Features die Variante besitzen kann. Neben dem *Scope* steht die *Region*, welche die Zielregion der Variante definiert. Hiervon sind unter anderem die verwendete Sprache als auch die verwendeten Karten für das Feature Navigation abhängig. Ebenfalls kann es länderspezifische Hardwarekonfigurationen aufgrund unterschiedlicher Gesetze und Regelungen geben.

Die nachfolgende Abbildung 3.4 gibt den rechts angehängten fortgeführten Verlauf der Variantenmatrix aus vorheriger Abbildung 3.3 wieder.

Status	Linked to SOP Calendar	Customer	System Type	DA or Navi	Unit Type	External Navigation
valid	NO	Customer A	Entry A	DA	W2DIN	DA
valid	YES	Customer A	Mid A	NAVI	W2DIN	BOSCH
CRQ	YES	Customer A	Mid A	NAVI	W2DIN	EXTNAV JPN
valid	YES	Customer A	Mid A	NAVI	W2DIN	BOSCH
valid	YES	Customer A	Entry A	DA	W2DIN	DA

Abbildung 3.4 Variantenmatrix: Status, SOP, Kunde, Typ

Zu jeder Variante gibt es einen *Status*. Dieser kann gültig, ungültig oder CRQ⁷ sein. Bei einem CRQ ist das Feld rot markiert, um den *Status* visuell hervorzuheben. Eine Variante ist in dem CRQ Status, wenn der Kunde eine Änderung an den Konfigurationseinträgen wünscht. Änderungen in den Konfigurationseinträgen wirken sich nicht auf produzierte Varianten aus, diese müssen nach einem erfolgreichen CRQ durch eine neue Variante ersetzt werden. Neben dem Status gibt es eine Information, ob die Variante in dem SOP Kalender integriert ist (*Linked to SOP Calendar*). Daneben wird der Kunde (*Customer*) genannt. Der Kunde definiert, für welche Klasse der Automobilreihe das Infotainment System bestimmt ist (*System Type*). Dies ist in diesem Fall die Einstiegsklasse (*Entry*) und Mittelklasse (*Mid*). Die

⁷ Change request

Spalte *DA or Navi* gibt an, ob es sich um ein Gerät ohne oder mit Navigationsfunktion handelt. Der *Unit Type* kann *W2DIN* für ein nach ISO 7736 genormte Höhe von 100,3 mm, oder *L-shape* sein. In dieser Tabelle sind alle Varianten *W2DIN*. Die rechte Spalte (*External Navigation*) in Abbildung 3.4 gibt an, ob es ein externes Navigationssystem gibt, sofern es sich um ein Infotainment System mit Navigationsfunktion handelt. Falls nicht, wird dort ebenfalls DA (Display Audio) eingetragen.

Die nachfolgende Abbildung 3.5 stellt analog zur vorherigen eine Weiterführung der Variantenmatrix dar. Die in der zweiten Zeile mit ‚sub cols‘ markierten Spalten weisen darauf hin, dass es sehr viele Unterspalten zu den jeweiligen Oberkategorien geben kann. Diese werden hier für eine bessere Übersicht nicht genauer erklärt.

Fascia	Display	Appearance in Printing Data File	Sample Phase	Plant	PCB	Tuner
sub cols	sub cols				sub cols	sub cols
			-	Plant W		
			-	Plant Y		
			-	Plant X		
			PP7.Y	Plant Y		
			PP10.W	Plant W		

Abbildung 3.5 Variantenmatrix: Front, Sample Phase, PCB, Tuner

Die Spalte *Fascia* besteht in der realen Variantenmatrix aus mehreren Spalten, die durch ein Bild die verwendete Front des Infotainment Systems definiert. Hier gibt es eine enge Verbindung mit den Automobilreihen (Blaue Linie), denn nicht jede Front passt in jede Automobilreihe. Das *Display* wird in der richtigen Matrix ebenfalls in mehreren Spalten durch Displaytyp, Hersteller, Auflösung etc. definiert.

Ob es ein Label von der Zertifizierungsabteilung für die Variante gibt, definiert die Spalte *Appearance in Printing Data File*, sobald diese grün markiert ist. Die *Sample Phase* gibt an, in welcher Phase sich das Produkt befindet. Die hier dargestellten Produkte befinden sich in der *PreProduction (PP)* Phase. Die Ziffer innerhalb der Phase legt ein Zeitfenster fest. Die nachfolgende Abkürzung (hier „Y“ oder „W“) ist eine Zuordnung der Produktionsstätte, welche ebenfalls in der Spalte *Plant* angegeben wird.

Die Spalten *PCB* und *Tuner* sind ebenfalls aus mehreren Spalten zusammengesetzt. Sie geben die Board-Konfiguration, also Board-Teilenummer, Board-Typ, etc. sowie die Tuner-Konfiguration, also FM/AM⁸, DAB⁹, etc., an.

Memory and Processor	External Interfaces	Destination	Software	External Connectors	Carline 1	Carline 2	Carline 3
sub cols	sub cols		sub cols	sub cols			
		USA				x	
		GBR			x		
		JPN					x
		GBR			x		
		USA				x	

Abbildung 3.6 Variantenmatrix: Speicher, Externe Anschlüsse, Software

Die Abbildung 3.6 ist ebenfalls ein stark zusammengefasster Überblick einiger Spalten der Variantenmatrix.

Die Spalte *Memory and Processor* gibt an, welcher Typ von Prozessor und wie viel Speicher (ROM, RAM) verbaut sind sowie den Hersteller des Speichers.

Die *Externen Interfaces* definieren unter anderem die CAN¹⁰ Bus Art sowie zum Beispiel USB OTG¹¹. Nochmals wird hier die Zielregion (*Destination*)

⁸ FM – Frequenzmodulation / AM – Amplitudenmodulation

⁹ Digital Audio Broadcasting – Digitaler Radio Übertragungsstandard

¹⁰ Controller Area Network, serielles Bussystem

¹¹ USB On-the-go, ermöglicht Verbindung zwischen zwei USB Geräten.

angegeben. Diese kann, sollte aber nicht unterschiedlich zu der in Abbildung 3.3 sein. Darauf aufbauend gibt es mehrere Spalten für die *Software*. Jeder Produktlinie werden speziell generierte Software-Images aufgespielt, welche hier abhängig von der Hardwarekonfiguration festgelegt werden. Ebenfalls werden hier IDs mit einfachen WENN, DANN-Formeln berechnet. Die *Externen Anschlussmöglichkeiten* in der nächsten Spalte sind durch Bilder visualisiert. Es handelt sich hier um verschiedene Stecker für Antennen, Strom und Datenbusse. Daneben gibt es eine feste Verknüpfung einer Variante mit den Automobilreihen (*Carline*).

Zusätzlich gibt es noch viele weitere Spalten, auch zwischen den oben genannten, die zum Beispiel die ID der zugehörigen Spezifikation festlegen, Softwarefeatures wie TTS¹² definieren, oder andere Hardwarefeatures wie ein CD-Laufwerk bestimmen. Eine detailliertere Beschreibung der über 230 Spalten ist im Rahmen dieser Arbeit nicht notwendig.

3.3 Allgemeine Herausforderungen

Es wurden während der Analyse des Prozesses eine Vielzahl von möglichen Problem- und Fehlerquellen entdeckt. Genauere teilprozessspezifische Problem- und Fehlerquellen werden in den jeweiligen nachfolgenden Abschnitten erläutert.

Eine der größtmöglichen Gefahrenquellen ist auch der Grund dieser Arbeit: Der nahezu komplette Prozess wird durch Excel-Tabellen verwaltet und organisiert. Diese Tabellen werden sowohl für die Kommunikation als auch für den Daten- und Informationsaustausch verwendet. Ein Grund ist laut Mitarbeitern die einfache Benutzbarkeit von Excel, aber auch die historische Verwendung innerhalb der Abteilungen. Mitarbeiter kennen sich seit langem mit Excel aus. Sie arbeiten täglich mit verschiedensten Tabellen und Formeln. Die Verwendung von Excel als allgemeines Hilfsmittel ist zunächst einmal nicht falsch. Sie wird jedoch in einigen Fällen problematisch, wenn mehrere Personen zur gleichen Zeit auf ein und die dieselbe Datei zugreifen und diese bearbeiten wollen. Die konkrete Änderung ist nicht in der Historie

¹² Text to Speech

sichtbar. Um das Problem des Änderungsgrundes zu verringern, werden von dem Unternehmen die Dateien in einem Xerox Docushare Dokumenten-Management System verwaltet. Dort sind die Mitarbeiter bei Änderungen dazu angehalten, den Grund der Änderung anzugeben. Ebenfalls müssen Änderungen innerhalb der Variantenmatrix in einer dafür vorgesehenen Tabelle eintragen.

Dateizugriffe als auch Änderungsgründe werden so dokumentiert. Dies erfordert einen disziplinierten Mitarbeiter, der mit dem Prozessablauf vertraut ist. Eine schnelle Änderung kann für den Mitarbeiter zu aufwändig für die Dokumentation sein und verliert somit die erforderliche Nachverfolgbarkeit. Ebenfalls problematisch wird es, wenn Änderungen unabsichtlich geschehen. Unbemerkt hat sich ein kleiner Fehler eingeschlichen. Dies kann bei Tabellen mit mehreren tausend Zeilen und hunderten Spalten durchaus vorkommen. Aufgrund der fehlenden Möglichkeit von Reviews, als auch eines Diff-Tools¹³ zwischen den Versionen, bleiben solche Fehler lange unbemerkt und könnten im schlimmsten Fall erst in der Produktion auffallen, verbunden mit sehr hohen Kosten.

Fehler können sich ebenfalls durch unbedachtes Kopieren und Einfügen von Daten einschleichen. Mangelnde Konzentration durch wiederholte Prozessabläufe oder Stress erhöhen das Risiko, beim Kopieren von größeren Datenmengen Fehler einzubauen oder gar fortzuführen (Chevez Bernaldo de Quiros, et al., 2008).

Die Größe der Tabelle sorgt ebenfalls für Probleme bei der Benutzung. Um Zeileninhalte lesen zu können, muss horizontal gescrollt werden. Dabei verliert der Nutzer sehr schnell den Fokus. Ebenfalls sind ausgewählte Filter nicht direkt ersichtlich.

Weitere Herausforderungen sind sich ändernde Anforderungen innerhalb des Unternehmens als auch vom Kunden. Dies erfordert eine laufende Anpassungsmöglichkeit, die aber aufgrund der Größe der Excel-Tabellen die Benutzbarkeit einschränkt.

Zusätzlich gibt es leere oder fehlerhafte Kommentare in der Tabelle, defekte Formeln und Einträge, die unkommentiert durchgestrichen sind. Es gibt

¹³ Dateivergleichsprogramm

viele zusammenklappbare Elemente und ebenfalls viele unterschiedliche und kräftige Farben.

3.4 Verarbeitung von Kundenspezifikationen

Nach Erläuterung des grundlegenden Aufbaus der Variantenmatrix und der Probleme, wird nachfolgend der gesamte Prozess und dessen Teilabschnitte näher beleuchtet, in dem die Variantenmatrix verwendet wird.

In diesem Abschnitt geht es um die Verarbeitung von Kundenspezifikationen. Die Spezifikationen werden von drei Mitarbeitern angenommen und analysiert, beantwortet und verwaltet. Die Mitarbeiter arbeiten während des Prozesses eng mit der Verkaufsabteilung und dem Projektmanager zusammen. Sobald es zu Fragen seitens der Spezifikation kommt, halten sie ebenfalls Rücksprache mit dem Kunden.

3.4.1 Überblick und Analyse

Die nachfolgende Abbildung 3.7 gibt einen Überblick über die Kommunikation mit dem Automobilhersteller.

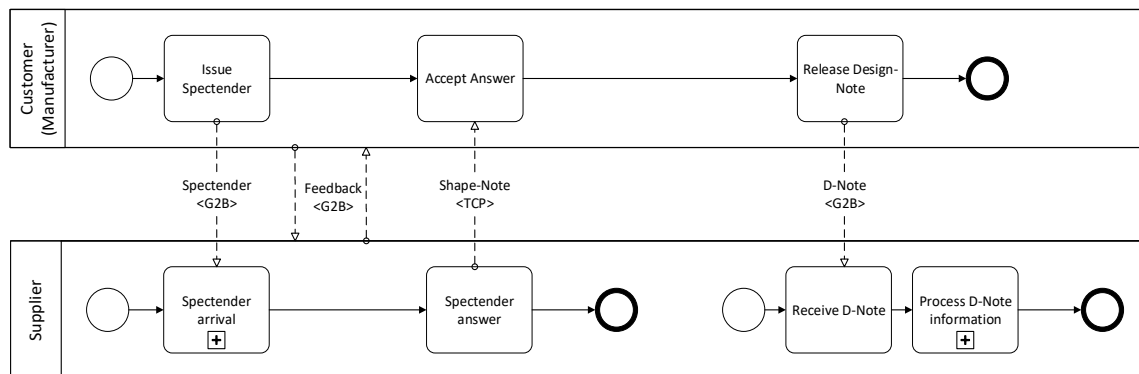


Abbildung 3.7 Spezifikation Prozess mit Kunden

Spezifikationen (*Spectender*) werden von den Mitarbeitern (hier: *Supplier*) bearbeitet, sobald diese über ein nicht näher definiertes *G2B*¹⁴ System vom Automobilhersteller (*Customer*) überreicht wurden. Die Spezifikationen werden dann von den Mitarbeitern geöffnet und verarbeitet (*Spectender arrival*). Während des Prozesses kann es *Feedback* an den Automobilhersteller geben, woraufhin die Spezifikation überarbeitet oder abgelehnt wird, da

¹⁴ Government to Business

sie aus verschiedenen Gründen nicht umsetzbar ist. Andernfalls wird die Spezifikation akzeptiert (*Spectender answer*) und es wird daraufhin eine offizielle *Shape-Note* an den Automobilhersteller übertragen. Dieser akzeptiert die Antwort (*Accept Answer*) und überträgt eine *Design-Note*. Diese Design-Note wird verarbeitet, denn sie ist im weiteren Prozess sehr wichtig (*Process D-Note information*). Sie ist am Ende auf dem Label des Gerätes sichtbar, damit die Werke des Kunden wissen, um welches Gerät es sich handelt. Ebenfalls ist die Übertragung der Design-Note eine offizielle Bestätigung, dass dieses Gerät in der vorher festgelegten Spezifikation bestellt wird.

In Abbildung 3.7 ist die Bearbeitung der Spezifikation nach deren Ankunft (*Spectender arrival*) ein wichtiger Abschnitt für diese Arbeit, denn hier kommt es zu mehreren Interaktionen mit der Variantenmatrix durch die Mitarbeiter. Die nachfolgende Abbildung 3.8 verdeutlicht dies.

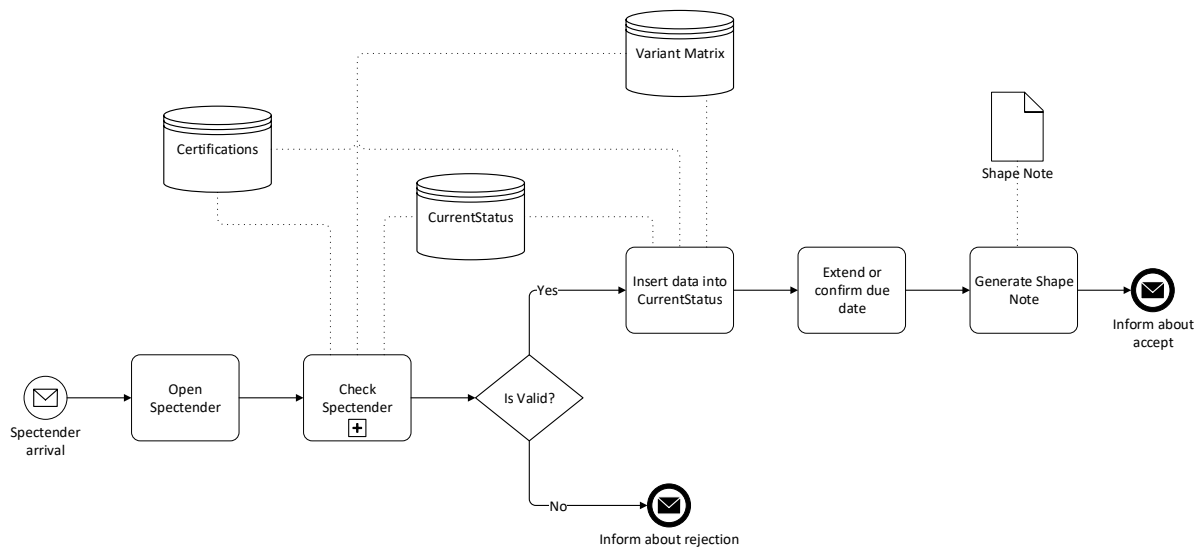


Abbildung 3.8 Ankunft der Spezifikation

Wie zuvor beschrieben wird die Spezifikation im ersten Schritt von den Mitarbeitern geöffnet (*Open Spectender*). Die Spezifikation ist eine Excel-Datei mit mehreren Sheets. Die erste Seite des Dokuments listet wichtige Informationen über die gewünschten Varianten auf. Ebenfalls wird festgelegt, für welche Automobilreihe die nachfolgenden Varianten bestimmt sind. Die Auflistungen enthalten die Kundenteilenummer, Informationen über die Variante, Zielregion, fahrzeugspezifische Informationen, beispielsweise ob es sich um Rechts- oder Linkslenker handelt sowie den Variantenindex aus

der Variantenmatrix. Spezifikationen können nach der Überprüfung (*Check Spectender*) akzeptiert oder abgelehnt werden. Bei einer Ablehnung wird der Kunde informiert. Bevor die Spezifikation akzeptiert wird, werden Daten aus dieser in die Excel-Tabelle *Current Status* gespeichert. Diese enthält den aktuellen Bearbeitungsstatus aller Spezifikationen sowie einen Link zum Index der Variantenmatrix. Zusätzlich gibt es in der Excel-Tabelle *Current Status* einer Kopie aller Zertifizierungsstatus.

Doch auch in der Abbildung 3.8 ist nicht ersichtlich, in welchem Umfang die Variantenmatrix während des Prozesses verwendet wird. Der Index der Variantenmatrix wird zwar in der Excel-Tabelle *Current Status* eingepflegt, doch auf weitere Assoziation mit der Variantenmatrix während des Überprüfens geht erst die nachfolgende Abbildung 3.9 ein. Sie erläutert den komplexeren Teilprozess *Check Spectender* aus Abbildung 3.8.

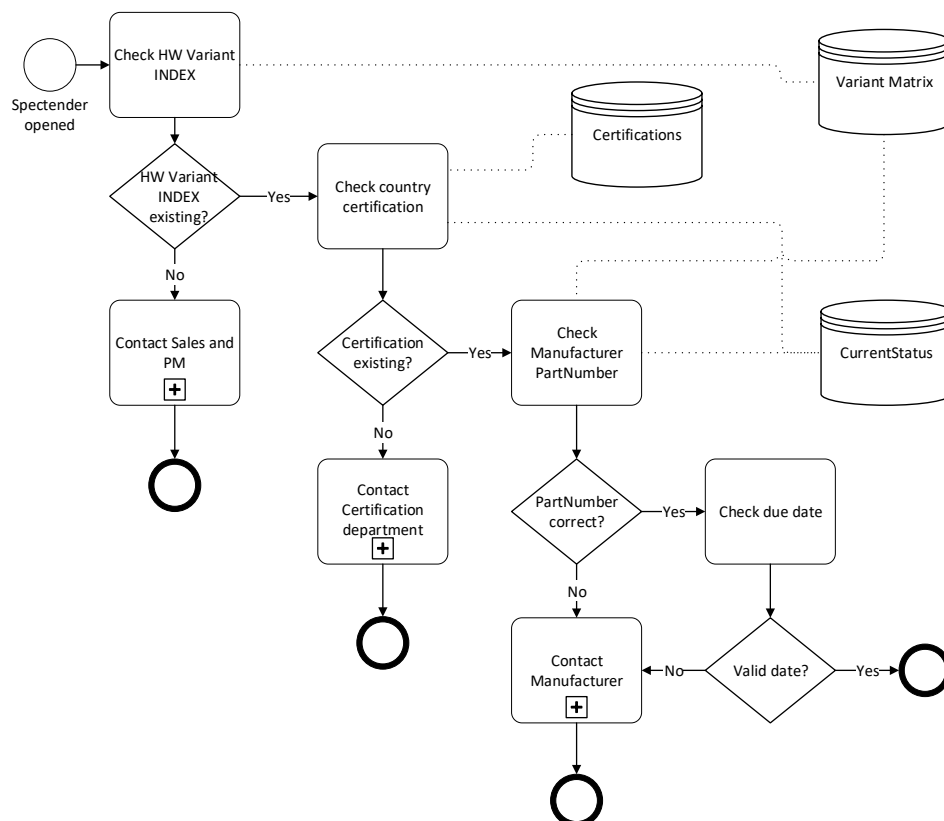


Abbildung 3.9 Spezifikation überprüfen

Nachdem eine Spezifikation geöffnet wurde, wird diese geprüft (*Check HW Variant INDEX*). Zuerst wird der Variantenmatrix-Index der Spezifikation mit dem der Variantenmatrix verglichen. Hierfür wird nach einem Index gesucht. Sollte es keinen oder einen ungültigen Index in der Spezifikation geben,

werden der Projektmanager und/oder die Verkaufsabteilung kontaktiert (*Contact Sales and PM*). Dieser komplexere Teilprozess wird bewusst nicht weiter erläutert, denn er ist situationsspezifisch und visuell nicht festhaltbar. Im nächsten Schritt (*Check country certification*) werden die Zertifizierungen und Homologationen der gewünschten Zielländer überprüft. Ohne gültige Zertifizierung darf ein Gerät nicht auf den dortigen Markt gebracht werden. Der Prozess der Zertifizierung ist sehr aufwendig und kostspielig. Er wird in Abschnitt 3.7 genauer erläutert. Sollte es keine gültigen Zertifizierungen für ein Land geben, wird zuerst die Zertifizierungsabteilung kontaktiert. Zusätzlich kann es je nach Situation auch Rücksprache mit dem Kunden geben. Dieser Vorgang wird ebenfalls nicht weiter erläutert, da er ebenfalls sehr komplex und nicht relevant für diese Arbeit ist.

Eine dritte Prüfung gilt der Kundenteilenummer (*Check Manufacturer Part-Number*). Es wird geprüft, ob diese in der *Current Status* Tabelle schon vorhanden ist, oder ob es sich um ein Variantenupdate handelt. Es wird überprüft, ob die fahrzeugspezifischen Informationen aus der Spezifikation mit denen der Variante aus der Variantenmatrix vereinbar sind. Die Herstellerteilennummer muss auch die geforderten Varianteneigenschaften wieder spiegeln. So besitzt beispielsweise ein Gerät mit Navigationseigenschaft die Ziffer 5 an der fünften Stelle der Herstellerteilennummer.

Zuletzt wird geprüft, bis wann die Variante produktionsreif sein soll (*Check due date*). Gibt es hier oder bei vorheriger Prüfung ein Problem, wird Rücksprache mit dem Kunden gehalten (*Contact Manufacturer*).

Der Gesamtprozess wird mehr als zehn Mal in der Woche von den Mitarbeitern durchgeführt.

3.4.2 Erkannte Probleme

Im Folgenden werden Probleme des Teilprozesses genannt. Einige dieser sind generelle Probleme des Prozesses, andere zeigen Grenzen einer Automatisierung auf.

Problem 1: Ein automatisierter Abgleich der Daten der Spezifikationen ist aufgrund von Differenzen zwischen diesen nicht möglich. Es gibt teilweise größere inhaltliche Unterschiede zwischen Spezifikationen unterschiedlicher Automobil-Produktlinien. Diese sind vor allem durch unterschiedliche

Bearbeiter seitens des Kunden begründet. Hier sollte der G2B Prozess enger definiert werden und Formate vorgegeben werden.

Problem 2: Die Projektkomplexität führt zu Verschiebungen von Terminvorgaben. Die Testproduktion, Qualitätssicherung, Erstellung maßgeschneiderter Software, Lizenzierung und Vorbereitungen der Werke erfordern mehrere Monate Vorlaufzeit. Manchmal kommen Spezifikationen zu spät oder mit neuen Anforderungen, welche sich nicht in den gewünschten zeitlichen Rahmen umsetzen lassen.

Problem 3: Anforderungen innerhalb einer Spezifikation vom Kunden können inkonsistent sein. Hier ist es erforderlich, den Kunden darauf hinzuweisen.

Problem 4: Änderungen an den Konfigurationseinträgen dürfen nur über ein CRQ erfolgen, welche ebenfalls einige Vorlaufzeit benötigen. Der Kunde verlangt dennoch eine schnelle Änderung und versucht daher Änderungen außerhalb des CRQ-Prozesses vornehmen zu lassen. Dies kann durch nicht vorhersehbare Nebeneffekte Auswirkungen auf die Qualität des Produktes haben. Beispiel: Hardwarefeature für Markt A soll für Markt B ausgerollt werden, ohne dass es eine Softwareanbindung für das Feature auf dem Markt B gibt. Daraus resultieren unter anderem folgende Fragen: Wie verhält sich dies bei zukünftigen Softwareupdates? Soll das Hardwarefeature in die Software eingebaut werden? Ist dies für den Markt B überhaupt gesetzlich möglich? Es muss konsequent dafür gesorgt werden, dass es keine Änderungen an den Varianten außerhalb von CRQs gibt.

Problem 5: Es werden Informationen aus anderen Excel-Dokumenten verschiedener Abteilungen in die eigenen Dokumente kopiert. Dadurch enthält der Prozess sehr viele redundante Informationen. Dies erfordert ständiges manuelles Pflegen der eigenen Dokumente über neue und geänderte Informationen anderer Abteilungen. Es wird vor allem manuell kopiert und verglichen. Dies ist sehr zeitintensiv und erfordert hohe Aufmerksamkeit. Fehler bei größeren Datensätzen können aufgrund einer schlechte Übersicht schnell übersehen werden.

3.5 Variantenverwaltung

In diesem Abschnitt geht um die Verwaltung von Varianten durch Hardwarespezialisten. Hierfür werden vor allem die Variantenmatrix, aber auch viele andere Informationsquellen und -kanäle genutzt.

Die Mitarbeiter der Variantenverwaltung fügen den Konfigurationseinträgen hauptsächlich neue Varianten hinzu oder ändern bestehende. Ebenfalls werden zu implementierenden Varianten den jeweiligen Automobilreihen hinzugefügt oder verändert. Ziel ist es, mit Hilfe der Informationen der Variantenmatrix die Bestückung der Leiterplatten festzulegen.

3.5.1 Überblick und Analyse

Die Hardwarespezialisten haben den Überblick über alle aktuell verwendeten Leiterplatten. Es gibt sechs verschiedene Grundlayouts, die in viele Versionen mit unterschiedlichen Features aufgeteilt sind. Die Anzahl der verschiedenen Versionen liegt hier im hohen zweistelligen Bereich. Davon werden aktuell nur wenige für die Massenproduktion verwendet. Eine Leiterplatte besitzt ungefähr 1800 bis 2000 Bauteile. Für die einzelnen Bauteile der Leiterplatte gibt es ebenfalls oft mehrere Hersteller, Güteklassen und sonstige technische Unterschiede. Es gibt innerhalb der Bauteile-Konfiguration zusätzlich Abhängigkeiten. Ein Großteil der Informationen wird in unterschiedlichen weiteren Excel-Tabellen verwaltet.

Die nachfolgende Abbildung 3.10 gibt einen groben Überblick über die Tätigkeiten der Variantenverwalter in Verbindung mit der Variantenmatrix.

Nachdem die Hardwarespezialisten von der für Kundenspezifikationen zuständigen Abteilung eine Benachrichtigung über eine neue Variante erhalten haben (*New Variant request*), wird im ersten Schritt die Anfrage überprüft (*Check request*). Es gibt mehrere Typen von Anfragen, wie zum Beispiel eine QLU¹⁵ oder ein CRQ¹⁶. Jede Anfrage ist unterschiedlich. So gibt es Anfragen, ganz neue Varianten einzuführen, meist jedoch gibt es nur Änderungen an der Softwareversion oder kleinerer Hardwarefeatures.

¹⁵ Quality-Level-Up

¹⁶ Change request

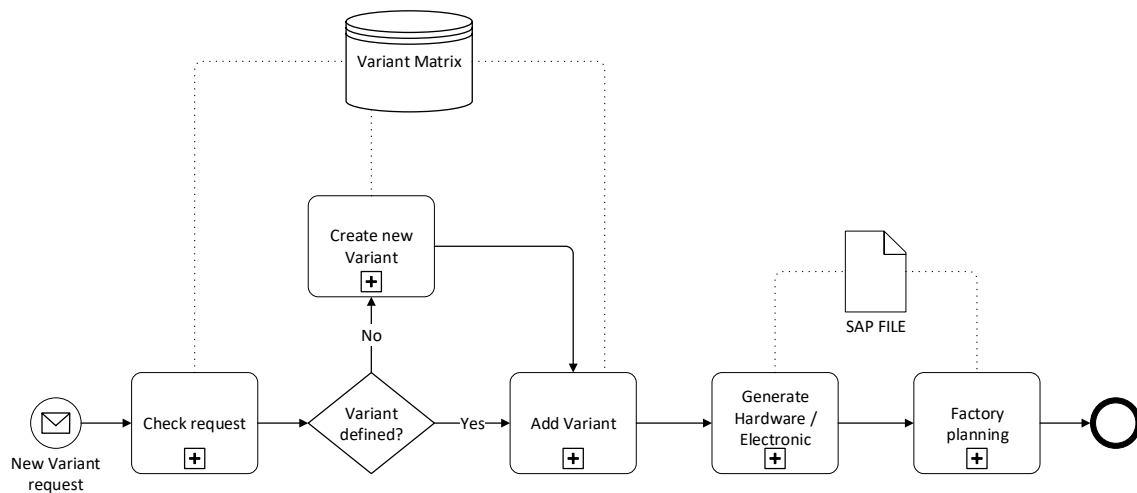


Abbildung 3.10 Variantenverwaltung: Überblick

Für die Überprüfung der Anfrage werden Variantenmatrix und Such- und Filterfunktion von Excel genutzt, um Varianten und deren Informationen zu finden. Nachfolgende Abbildung 3.11 schlüsselt den Teilprozess „*Create new Variant*“ aus Abbildung 3.10 weiter auf.

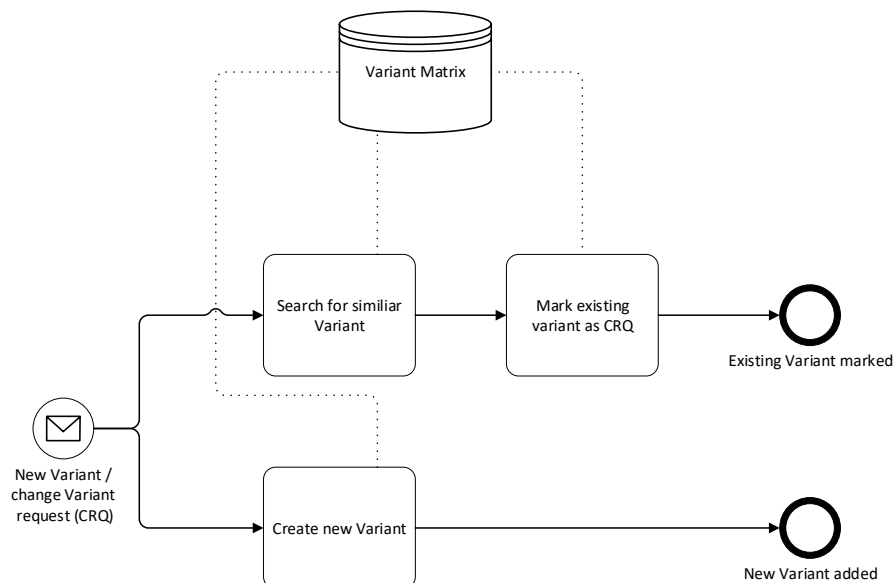


Abbildung 3.11 Variantenverwaltung: Erstellung / Änderung einer Variante

Sollte die Variante nicht vorhanden, oder ein CRQ nötig sein (Abbildung 3.10 – *variant defined* → *no*), wird dies von einem weiteren Mitarbeiter bearbeitet (Abbildung 3.11). Der Mitarbeiter fügt entweder die neue Variante zu den Konfigurationseinträgen hinzu (*Create new Variant*), oder markiert bestehende als CRQ (*Mark existing variant as CRQ*). Sollen Änderungen an einer bereits bestehenden Variante erfolgen, so werden der Konfigurationseintrag sowie die zugehörigen implementierten Varianten gesucht

(*Search for similar Variant*). Diese werden dann als CRQ markiert und im Hintergrund ein nicht näher definierter Prozess gestartet, der die geforderten Änderungen für die Varianten anstößt. Erklärungen der Konfigurationseinträgen und weiterer Felder der Variantenmatrix sind in Abschnitt 3.2 zu finden. Sobald der Mitarbeiter nach bestehenden Varianten sucht, benutzt er ebenfalls die Such und Filterfunktionen in Excel.

Die neue Variante in den Konfigurationseinträgen wird in einem größeren Prozess manuell der Variantenmatrix hinzugefügt (Abbildung 3.10 – *Add Variant*). In diesem Schritt wird ebenfalls die zu implementierende Variante aus der Anfrage durch ein spezielles VBA¹⁷-Script in die Variantenmatrix geschrieben. Dafür ist wiederum ein weiterer Mitarbeiter zuständig. Das Script ist in fünf Abschnitte unterteilt und benötigt neben der Variantenmatrix noch weitere komplexe Excel-Tabellen mit relevanten Informationen aus anderen Abteilungen.

Vor Ausführung des Skripts, werden die benötigten Dateien in einen neuen Ordner im Dateisystem kopiert. Dies ist eine Sicherheitsmaßnahme des Mitarbeiters, um mögliche fehlerhafte Änderungen an anderen Dateien zu vermeiden. Anschließend wird das Excel Dokument mit dem Skript geöffnet. Dieses muss zuerst konfiguriert werden. Dazu wird der zu implementierende Index aus der Variantenmatrix eingegeben. Danach muss manuell die Automobilreihe angegeben werden, für welche die Variante implementiert werden soll. Nachfolgend wird eine geplante Sample Phase und das Werk, in dem die Variante produziert werden soll, manuell festgelegt. Erst dann kann das Script gestartet werden.

Im ersten Schritt wird lediglich die Variantenmatrix geöffnet, es wird die Zeile in den Konfigurationseinträgen markiert, von welcher die neue Variante abgeleitet werden soll. Dies wird vom Mitarbeiter überprüft, bevor er den nächsten Schritt des Scripts ausführt. Der zweite Schritt fügt grobe Informationen aus dem Konfigurationseintrag in eine neue Zeile unterhalb der zugehörigen Automobilreihe ein. Im dritten Schritt des Scripts werden weitere feiner aufgelöste Informationen zu der Variante in die neue Zeile eingefügt. Dabei kommen die mit in dem Ordner liegenden weiteren Excel

¹⁷ Visual Basic for Applications - Skriptsprache

Dateien zum Einsatz. Das Script öffnet diese und fragt den Benutzer an einigen Stellen, welche Einträge aus den Tabellen genutzt werden sollen. Der Benutzer wird über eine erfolgreiche Durchführung des dritten Schrittes informiert. Im vierten Schritt werden weitere PCB¹⁸ Informationen in die Zeile der Variantenmatrix eingefügt. Dazu wird analog zum vorherigen Schritt der Benutzer aufgefordert, die richtigen Informationen aus anderen Tabellen anzuklicken oder anzugeben. Der letzte Schritt ist mittlerweile obsolet und diente der Replikation des neuen Eintrags in der Variantenmatrix zur Reservierung von Teilenummern. Aufgrund der Unübersichtlichkeit wird dies nicht mehr durchgeführt.

Das Script beinhaltet einfache Gültigkeitsüberprüfungen. Es vereinfacht das Kopieren von Informationen erheblich und sorgt für eine deutlich schnellere Erstellung von Varianten. Die Durchführungsdauer des Scripts beträgt ungefähr acht Minuten, manuell würde der Prozess nach Angaben des Mitarbeiters ungefähr fünfmal so lange dauern und erfordere eine viel größere Aufmerksamkeit.

Im nachfolgenden Schritt (Abbildung 3.10 – *Generate Hardware / Electronic*) wird eine *SAP-Datei* erstellt, welche die Hardwareinformationen aus der Variantenmatrix beinhaltet. Diese Datei als auch der zugehörige Prozess wurden aus Zeitgründen nicht näher untersucht. Die Fabrikplanung (*Factory planning*) nutzt die Daten aus dem nicht näher spezifizierten SAP System. Der Projektmanager wird ebenfalls informiert, damit er die Informationen der neuen Variante in die Sample Distribution List eintragen kann.

Eine weitere Aufgabe der Mitarbeiter in der Variantenverwaltung ist die Überprüfung von Differenzen zwischen den Varianten. Hierfür wird ebenfalls die Filterfunktion von Excel genutzt, um Varianten zu finden. Um Unterschiede zwischen zwei und mehreren Varianten zu erkennen werden die Einträge teilweise manuell in eine neue Tabelle kopiert und analysiert. Dieser manuelle Prozess nimmt laut dem befragten Mitarbeiter viel Zeit in Anspruch.

¹⁸ printed circuit board - Leiterplatte

Gesucht und gefiltert wird vor allem über folgende Spalten: Index, Teilenummer, Leiterplatte (PCB), Sample Phase, Display und Scope. Es kann auch Fälle geben, in denen Filter kombiniert oder zusätzliche Informationen anderer Spalten der Variantenmatrix hinzugezogen werden.

Ebenfalls werden Änderungswünsche der Matrix angenommen und umgesetzt. So kommt es aufgrund externer (~70% Anteil) als auch interner Prozessanpassungen (~30% Anteil) zu Änderungen der Spalten der Variantenmatrix. Spalten können umbenannt werden, neue Spalten hinzugefügt oder existierende Spalten entfernt werden. Änderungen sind alle ein bis zwei Monate nötig. Externe Anpassungen sind begründet durch Kundenwünsche sowie Gesetzesänderungen und somit nicht kontrollierbar.

3.5.2 Erkannte Probleme

Auch bei der Variantenverwaltung gibt es mögliche Problemstellen.

Problem 1: Ein Großteil von prozessrelevanten, wichtigen Informationen, wird manuell in Excel-Tabellen verwaltet. Viele dieser Tabellen werden von wenigen Personen gepflegt.

Problem 2: Das manuelle Hinzufügen neuer Varianten ist sehr zeitaufwendig. Zur Verringerung des Arbeitsaufwandes haben die Mitarbeiter komplexe VBA-Skripte gebaut, die jedoch fehleranfällig sein können.

Problem 3: Anforderungen an die Variantenmatrix ändern sich im laufenden Prozess. Ungefähr 70% der nötigen Änderungen entstehen durch externe Prozesse, 30% durch interne Änderungen angrenzender Prozesse. Zu den externen Prozessen gehören unter anderem sich ändernde Kundenanforderungen oder Gesetzesänderungen. Hier entsteht zum einen die Gefahr, dass für den Gesamtprozess nicht relevante Informationen in der Variantenmatrix abgebildet werden, um z.B. Kundenanforderungen umzusetzen. Der Kunde sollte über eine eigene Ansicht verfügen, die eventuell unabhängig von der Variantenmatrix existiert.

Zum anderen sind Änderungen von Anforderungen und somit auch Spalten der Variantenmatrix für eine mögliche Datenbanklösung ebenfalls problematisch. Dieser Punkt wird in Abschnitt 6.2 näher erläutert.

3.6 Projektmanager

In diesem Abschnitt wird die Rolle des Projektmanagers innerhalb des Prozesses näher untersucht. Der Projektmanager ist hauptverantwortlich für die SDL-Dateien und steht ebenfalls im engen Kontakt mit den Kunden. Zusätzlich plant und überwacht er Teile des Gesamtprozesses, um die vom Kunden vorgegebenen Meilensteine zu erreichen.

Aufgrund mangelnder Zeit der Mitarbeiter für die Interviews, erscheint der Überblick der Rolle kürzer. Die kurzen Interviews genügten dennoch, um festzustellen, dass es Ähnlichkeiten zwischen den Mitarbeitern der in der Verarbeitung der Kundenspezifikation (siehe Abschnitt 3.4) und den Projektmanagern bei der Benutzung der Variantenmatrix gibt.

Aufgrund des Zeitmangels wurden keine weiteren Probleme ersichtlich. Die allgemeinen Probleme, genannt in Abschnitt 3.3, existieren auch bei den Projektmanagern.

3.6.1 Überblick und Analyse

Der Projektmanager ist hauptverantwortlich für das Infotainment-System einer oder mehrerer Automobilreihen. Er ist der erste Ansprechpartner des Kunden und verantwortlich für die erfolgreiche Entwicklung eines Infotainment-Systems. Ziel ist es den Start der Produktion einzuhalten und den Kunden mit den gewünschten Geräten zu beliefern.

Aufgrund seines engen Kontaktes zu dem Kunden hat der Projektleiter oft Einblick in den dortigen Entwicklungsprozess und kann dadurch neue mögliche Varianten erkennen, und zwar oft, bevor die erste Spezifikation bei der Rolle der Spezifikationsverarbeitung (siehe Abschnitt 3.4) erscheint.

Nachdem eine neue Variante in die Variantenmatrix eingetragen wurde (Abschnitt 3.5.1 und Abbildung 3.10), wird der Projektmanager informiert. Dieser sucht die neue Variante und erweitert die *Sample Distribution List* (SDL)-Datei (Abbildung 3.2) mit den Informationen aus der Variantenmatrix. Hierfür wird die Such- und Filterfunktion von Excel genutzt, gefundene Daten werden dann manuell aus der Variantenmatrix in die SDL-Datei kopiert. Ebenfalls fließen weitere Informationen, wie zum Beispiel die Bestellmenge vom Kunden, Informationen aus der Spezifikation und weitere nicht näher

untersuchte Schnittstellen in die SDL-Datei. Ziel ist es, den Werken einen Plan über Test-Produktionen und deren Mengen zu verschaffen. Die SDL-Datei besitzt ebenfalls Informationen, welches Testsample aktuell von welcher Abteilung oder welchem Mitarbeiter benutzt wird.

Samples in der SDL-Datei werden bestimmten Phasen, Zeitrahmen und Werken zugeordnet. Diese Informationen fließen zurück in die Variantenmatrix (Abbildung 3.5). Aktuell werden die Informationsflüsse manuell getätigt. Hier kann es schnell zu Abweichungen kommen.

Die Projektmanager wünschen sich einen Überblick über alle aktuell genutzten Geräte sowie über die aktuelle Sample Phase und deren Geräte.

3.7 Zertifizierung

Dieser Abschnitt gibt einen kurzen Einblick in die Rolle der Zertifizierung. Jede Hardwarekombination benötigt für die jeweiligen Zielmärkte gültige Zertifizierungen der einzelnen Länder. Der Zertifizierungsprozess ist oft sehr langwierig und bedarf deshalb einer gewissen Vorausplanung. Größere technische Änderungen sind nachträglich nicht mehr möglich. Ohne gültige Zertifizierung des Infotainment-Systems darf das Kraftfahrzeug in den Zielmärkten nicht in Betrieb genommen werden. Der Prozess der Zertifizierung ist mit hohen Kosten verbunden. Es entsteht am Ende ein Label, das sogenannte Homologation-Sheet, mit Zeichen und Nummern, welches auf jedem Gerät sichtbar angebracht werden muss. Ebenfalls gibt es Hinweise in der Bedienungsanleitung zu den Homologationen.

3.7.1 Überblick und Analyse

Eine Homologation ist fahrzeugspezifisch. Sobald sich bestimmte Hardwarebauteile des Infotainment-Systems ändern, ist eine erneute Homologation erforderlich, nicht aber bei Softwareänderungen. Das System benötigt zwingend eine gültige Zertifizierung für die Zielmärkte, denn ansonsten ist, wie eingangs erwähnt, das Fahrzeug nicht zulassungsfähig. Für die Verwaltung der Homologationen verwendet die Rolle hauptsächlich zwei unterschiedliche Excel-Tabellen.

In der einen werden die Infotainment-Systeme der Automobilreihen auf der Horizontalen, den Ländern und Regionen auf der Vertikalen, zugeordnet. Sobald es eine gültige Homologation für ein Land oder eine Region gibt, wird das Feld grün markiert und eine Ziffer eingetragen. Die Bedeutung der Ziffer ist äquivalent zu der grünen Markierung. Fehlende oder laufende Homologationen werden gelb markiert und besitzen eine jeweils andere Ziffer. Geräte, die noch nicht vollständig homologiert oder nicht für diese Länder oder Regionen bestimmt sind, werden so erkannt.

In einer anderen Excel-Tabelle werden Daten für ein sogenanntes *Printing Data File* zusammengefasst. Dort befindet sich der Name der Automobilreihe, eine Teilenummer, die D-Note (siehe Abschnitt 3.4) und alle weiteren nötigen Zertifikate. Viele der Informationen stammen aus der Variantenmatrix. Die Mitarbeiter benutzen die Suche und den Filter aus Excel, um relevante Informationen zu finden. Anschließend werden die Informationen kopiert und in die Zielfeile eingefügt. Im Folgenden wird daraus ein Label generiert. Eines für die PreProduction Phase, und im weiteren Verlauf ein weiteres für die Produktion. Es ist äußerst wichtig, dass hier die richtigen Daten auf dem Label stehen, denn ansonsten wird der Kunde die Systeme nicht in die Fahrzeuge einbauen.

Die Variantenmatrix besitzt eine Spalte „*Unit appearance in Printing Data File*“, in der Felder grün markiert sind, sobald ein Gerät ein Label besitzt. Auch hier muss sichergestellt werden, dass es zu keinen Abweichungen kommt. Aktuell werden die Daten manuell übertragen.

3.7.2 Erkannte Probleme

Während des Interviews wurde festgestellt, dass sich die Mitarbeiter in der Nutzung der Variantenmatrix nicht sonderlich unterscheiden. Es werden, ähnlich wie bei dem Projektmanager (Abschnitt 3.6), Informationen über die Suche und den Filter gesucht und diese in eigene Dateien und Tabellen kopiert. Ein Status wird anschließend wieder in die Variantenmatrix zurückgeführt. Der Prozess ist sehr fehleranfällig, da er ebenfalls manuell erfolgt. Der Arbeitsaufwand, die Daten aktuell zu halten, ist ebenfalls nicht zu unterschätzen.

Ein weiteres Problem sind die in den verschiedenen Tabellen benutzten Länderkürzel und -namen. Die Variantenmatrix selbst besitzt drei unterschiedliche Angaben zu den Regionen. Die Mitarbeiter der Zertifizierung verfügen über eine weitere Darstellungsvariante der Zielregionen. Zusätzlich nennt der Automobilhersteller gleiche Regionen unterschiedlich. Mitarbeiter verlieren hier schnell den Überblick über die Ländernamen und deren Kürzel.

3.8 Konfigurationsmanagement

Der Konfigurationsmanager ist, wie in Abbildung 3.2 auf Seite 12 sichtbar, während des gesamten Prozesses involviert. Er ist zuständig, zwei verschiedene Dateien zu generieren, die in den Speicher des Geräts geflasht¹⁹ werden. Die eine Datei nennt sich *PD-Configuration* (Production Diagnosis), die andere *CD-Configuration* (Customer Diagnosis). Ebenfalls ist die Rolle für das Erstellen und Überprüfen von Konfigurationsregeln zuständig.

Da bei der Analyse der Rolle keine prozessspezifischen Probleme erkannt wurden, entfällt der Abschnitt.

3.8.1 Überblick und Analyse

Die Rolle des Konfigurationsmanagers ist, wie eingangs genannt, verantwortlich für die PD/CD-Konfiguration. Der Konfigurationsmanager erstellt auf Anfrage die nötigen Konfigurationen. Anfragen werden über Jira²⁰ erstellt und verwaltet. Ebenfalls erstellt er neue Konfigurationsregeln oder prüft und ändert bestehende.

Sobald er eine neue Anfrage zu einer Konfiguration über Jira erhält, sucht sich der Mitarbeiter die benötigten Informationen aus der Variantenmatrix sowie vielen weiteren Excel-Tabellen. Dabei kopiert er benötigte Daten in eigene Excel-Tabellen. Im Nachfolgenden werden fahrzeugspezifische Regeln konfiguriert. Anschließend gibt es ein Script, welches die PD- und CD-Konfiguration erstellt. Das Script durchsucht dabei die eigenen Tabellen nach relevanten Informationen und verarbeitet diese. Die generierten

¹⁹ Flashen: ROM-Speicher beschreiben

²⁰ Software zur Vorgangs- und Projektverfolgung

PD/CD-Konfigurationen besitzen einen Hash-ID-Wert, welcher nach erfolgreichem Durchlauf des Scripts in eine Excel-Tabelle geschrieben wird. Diese ID ist einzigartig. Die anfallenden Dateien werden auf einem Server abgelegt und müssen dort aufgrund rechtlicher Regelungen mindestens 10 Jahre verfügbar bleiben. Eine einzige PD und CD Configuration kann dabei mehrere hundert Megabyte groß sein. Das Erstellen einer Konfiguration dauert je nach Automobilreihe mehrere Stunden.

Zusätzlich ist die Rolle zuständig für Änderungen von Konfigurationsregeln. Diese werden ebenfalls über Jira kommuniziert. Diese Regeln legen mögliche Konfigurationskombinationen fest und sind ebenfalls in Excel-Tabellen gespeichert. Aufgrund der Unabhängigkeit zur Variantenmatrix, werden die Regeln nicht weiter analysiert.

Der Konfigurationsmanager benötigt eine Vielzahl an Informationen aus der Variantenmatrix, die je nach Arbeitsschritt und Aufgabe unterschiedlich sind.

3.9 Zusammenfassung der Prozessanalyse

Die in dieser Arbeit zu verbessernde Variantenmatrix ist ein wichtiger Bestandteil des Prozesses. Es fließen Informationen von sehr vielen Quellen in die Tabelle, es werden auch genauso viele Informationen aus der Variantenmatrix entnommen und in weitere Excel-Tabellen eingefügt. Ein Großteil dieser Arbeiten geschieht manuell. Es gibt aber auch Mitarbeiter, die selbst entwickelte Skripte für sich wiederholende Aufgaben entwickelt haben.

Wie in nachfolgender Tabelle 1 sichtbar wird, nutzen die Mitarbeiter die Variantenmatrix hauptsächlich als Informationsquelle. Sie stellt die Analyse der Zugriffe auf die Excel basierte Variantenmatrix in einem Zeitraum von 2 Wochen dar. Ein größerer Zeitraum ist aufgrund von Systemeinschränkungen nicht wählbar.

Zugriffsart	Zugriffe gesamt	Accounts
lesen	504	>110
schreiben	26	6

Tabelle 1 Zugriffsstatistik Excel Variantenmatrix

Ungefähr 95% aller Zugriffe erfolgen ausschließlich lesend. Dies deckt sich mit den Erfahrungen, die bei kurzen Vorstellungen, Gesprächen und Interviews mit Mitarbeitern gesammelt wurden. Die Zugriffsstatistik enthält ebenfalls Nutzer, die nicht Teil des in diesem Kapitel vorgestellten Prozesses sind. Die Variantenmatrix wird ebenfalls von vielen anderen Mitarbeitern unterschiedlicher Positionen als Informationsquelle genutzt. Ein Großteil der Mitarbeiter außerhalb des hier vorgestellten Prozesses nutzt die Variantenmatrix nur gelegentlich.

Insgesamt lassen sich folgende drei Gruppen von Nutzern bilden:

Gruppe 1 sind Nutzer, die täglich mit der Variantenmatrix arbeiten und sehr viele Informationen bearbeiten dürfen. Hierzu gehören die Mitarbeiter der Variantenverwaltung (siehe Abschnitt 3.5).

Gruppe 2 sind Nutzer, die häufig bis täglich mit der Variantenmatrix arbeiten, aber nur wenige bis gar keine Informationen bearbeiten dürfen. Hierzu gehören die Mitarbeiter der Verarbeitung von Kundenspezifikation (siehe Abschnitt 3.4), Projektmanager (siehe Abschnitt 3.6), Mitarbeiter der Zertifizierung (siehe Abschnitt 3.7) sowie Konfigurationsmanager (siehe Abschnitt 3.8).

Gruppe 3 sind Nutzer, welche seltener Informationen aus der Variantenmatrix suchen und keine Informationen bearbeiten dürfen. Hierzu gehören alle anderen Mitarbeiter.

In Verbindung mit Tabelle 1 lässt sich folgendes aussagen:

- Die 95% der lesenden Zugriffe erfolgten von Gruppe 1, Gruppe 2 und Gruppe 3.
- Schreibende Zugriffe erfolgten ausschließlich von Mitarbeitern der Gruppe 1 und / oder einigen der Gruppe 2.

4 Anforderungsanalyse

Nachdem im vorherigen Kapitel der Prozess genauer betrachtet wurde, werden nun Anforderungen an eine mögliche Software genannt, welche die Mitarbeiter bei der Nutzung der Variantenmatrix unterstützen soll. In den ersten drei Abschnitten des Kapitels werden alle Anforderungen beschrieben und in einem jeweils darauffolgenden Absatz erklärt. Sämtliche Anforderungen besitzen eine Nummer sowie eine Abkürzungsform. Die Kurzform gibt an, um welche Anforderung es sich handelt. Funktionale Produktanforderungen beginnen mit „R“ und werden im ersten Abschnitt 4.1 genannt. Nichtfunktionale Produktanforderungen beginnen mit der Abkürzung „NR“ und werden in Abschnitt 4.2 genannt. Sicherheitsrelevante Anforderungen beginnen mit der Abkürzung „SR“. Diese werden im Abschnitt 4.3 dieses Kapitels genannt. Die Kombinationen aus Nummer und Abkürzung sind eindeutig.

Das Ziel der Anforderungsanalyse ist es, die für die Mitarbeiter relevanten Anforderungen festzuhalten, damit das richtige Produkt entwickelt wird.

Alle Anforderungen werden nach Schablonen (Ebert, 2008 S. 148, ff.) sowie (Rupp, 2014) formuliert, die dabei helfen, sie klar auszudrücken und zu strukturieren.

Aufgrund des zeitlichen Rahmens als auch der Komplexität der Variantenmatrix innerhalb des Prozesses können nicht alle Anforderungen mit einem Prototyp umgesetzt werden. Es folgt daher ein weiterer Abschnitt 4.4, der die vorherigen Anforderungen einschränkt und festlegt, welche für einen Prototyp relevant sind.

4.1 Funktionale Produktanforderungen

Es folgen nun durch Interviews und Gespräche erkannte Anforderungen. Diese werden, wie zuvor beschrieben, nummeriert und besitzen eine Abkürzung, um sie den Anforderungstypen zuordnen zu können. Jede Anforderung wird in einem kurzen nachfolgenden Text beschrieben.

Es werden alle erkannten Anforderungen genannt, dabei kann es noch zu Unstimmigkeiten zwischen diesen kommen. Es ist daher wichtig, die

Anforderungen anschließend zu sortieren sowie einzuschränken. Es werden nicht alle Anforderungen bis in das kleinste Detail genannt, sie sollen aber ausreichend großen Überblick über wichtige geforderte Funktionen geben.

[R01]: *Die Anwendung muss dem Benutzer die Möglichkeit bieten, das Benutzerinterface in den Sprachen Deutsch und Englisch anzuzeigen.*

Da Bosch ein international tätiges Unternehmen ist und Mitarbeiter aus unterschiedlichen Ländern und Regionen zusammenarbeiten, muss die Anwendung mindestens mit den Sprachen Deutsch und Englisch nutzbar sein.

[R02]: *Die Anwendung muss dem Benutzer die Möglichkeit bieten, alle Inhalte der zugrundeliegenden Excel-Variantenmatrix anzuzeigen.*

Es sollen alle Inhalte der Excel-Variantenmatrix in der Anwendung abgerufen werden können. Dabei darf es nicht zu Inkonsistenzen kommen. Somit dürfen keine Daten verloren gehen oder falsche Daten angezeigt werden.

[R03]: *Die Anwendung muss dem Benutzer die Möglichkeit bieten, Inhalte der Variantenmatrix zu durchsuchen.*

Wie in Abschnitt 3.9 genannt sowie in den Interviews erkannt, wird in der Excel-Tabelle der Variantenmatrix die Suchfunktion sehr häufig genutzt. Daher ist es äußerst wichtig, eine Möglichkeit zur Durchsuchung der vorhandenen Daten anzubieten.

[R04]: *Falls eine Sucheingabe zu keinem Ergebnis führt, muss die Anwendung dem Benutzer einen Hinweis anzeigen.*

Das System sollte den Nutzer immer über den momentanen Status informieren (Nielsen, et al., 1994).

[R05]: *Die Anwendung muss dem Benutzer die Möglichkeit bieten, Suchanforderungen bestehend aus mehreren Begriffen zu verarbeiten.*

Bei Beobachtungen während der Interviews wurde festgestellt, dass die Benutzer komplexere Suchen durchführten. Damit die Anwendung die

Mitarbeiter in ihrer Nutzung nicht einschränkt, muss sie die Suche über mehrere Begriffe ermöglichen.

[R06]: *Die Anwendung muss dem Benutzer die Möglichkeit bieten, die Sucheingaben zu löschen.*

Falls der Anwender eine Suche rückgängig machen möchte, muss dieses vom System ermöglicht werden.

[R07]: *Die Anwendung muss dem Benutzer die Möglichkeit bieten, die für ihn sichtbaren Inhalte der Variantenmatrix zu filtern.*

Wie in Abschnitt 3.9 sowie in allen Interviews und Mitarbeitergesprächen genannt, wird der in Excel verfügbare Filter genutzt, um die anzuzeigenden Geräte in ihrer Auswahl einzuschränken. Er wird ebenfalls genutzt, um Geräte mit bestimmten Eigenschaften zu finden, sowie für exploratives Suchen nach anderen Geräten.

[R08]: *Die Anwendung muss dem Benutzer die Möglichkeit bieten, einen Filtervorgang mit mehreren aktivierten Filtern durchzuführen.*

Diese Anforderung ist eine Erweiterung von [R07] und legt fest, dass auch mehrere Filter zu gleicher Zeit aktiviert sein können. Mitarbeiter können dadurch die Auswahl weiter einschränken.

[R09]: *Falls zu einem ausgewählten Filter oder einer Filterkombination keine Ergebnisse vorhanden sind, muss die Anwendung dem Benutzer einen Hinweis anzeigen.*

Identisch zu Anforderung [R04] muss die Anwendung immer ihren aktuellen Zustand anzeigen und den Nutzer bei einer erfolglosen Suche / Filterwahl informieren.

[R10]: *Die Anwendung muss dem Benutzer die Möglichkeit bieten, die für ihn sichtbaren Inhalte kopieren zu können.*

Für die Kommunikation zwischen den Mitarbeitern sowie der Pflege anderer Excel-Tabellen, müssen die Inhalte der Variantenmatrix kopiert werden können.

[R11]: Die Anwendung soll dem Benutzer „Administrator“ die Möglichkeit bieten, in einer Benutzerverwaltung die Lese und Schreibzugriffe anderer Benutzer(gruppen) einstellen zu können.

Mitarbeiter der Variantenverwaltung sollen zum Beispiel Schreibzugriff auf alle Felder erhalten (siehe Abschnitt 3.5.1). Projektmanager (Abschnitt 3.6) oder Mitarbeiter des Konfigurationsmanagements (Abschnitt 3.8) haben fast ausschließlich lesend Zugriff (siehe nachfolgende Anforderungen). Leitende Mitarbeiter haben den Wunsch nach einer Benutzerverwaltung geäußert, in dem berechnigte Mitarbeiter die Zugriffe anderer einstellen und Verwalten können.

[R12]: Die Anwendung muss sicherstellen, dass nur authentifizierte Nutzer Zugriff auf die Daten haben.

Jeder Benutzer muss sich bei Nutzung der Anwendung authentifizieren, um so deren Zugriffsart festzulegen.

[R13]: Die Anwendung sollte der Benutzergruppe „Kunde/Automobilhersteller“ ausschließlich lesenden Zugriff gewähren.

Der Kunde hat ebenfalls Zugriff auf die Variantenmatrix, darf aber die Inhalte nicht editieren können.

[R14]: Die Anwendung sollte der Benutzergruppe „Kundenspezifikationsverarbeitung“ ausschließlich lesenden Zugriff gewähren.

Mitarbeiter der Verarbeitung von Kundenspezifikationen (Abschnitt 3.4) nutzen die Excel-basierte Variantenmatrix ausschließlich zum Suchen und Vergleichen von Informationen. Es dürfen keine Änderungen von den Mitarbeitern durchgeführt werden.

[R15]: Die Anwendung sollte der Benutzergruppe „Variantenverwaltung“ lesenden sowie schreibenden Zugriff gewähren.

Wie in Anforderungserklärung [R11] sowie in Abschnitt 3.5 genannt, sind die Mitarbeiter der Variantenverwaltung für das Ändern und Hinzufügen von Varianten verantwortlich. Sie müssen daher lesenden und schreibenden Zugriff besitzen.

[R16]: Die Anwendung sollte der Benutzergruppe „Projektmanager“ für das Attribut „Sample Phase“ schreibenden Zugriff gewähren und auf alle anderen Attribute lesenden Zugriff.

Projektmanager (Abschnitt 3.6) dürfen nur in das Feld „Sample Phase“ eines Eintrags in der Variantenmatrix schreiben. Hier sollen sie angeben können, in welcher Sample Phase (siehe Abschnitt 3.2) sich das Gerät befindet. Der Projektmanager benötigt nicht jedes Attribut der Variantenmatrix. Die genauen Bezeichnungen der benötigten Attribute konnten aufgrund unterschiedlicher Anforderungen einzelner Projektleiter nicht ermittelt werden. Daher besitzt er in aktueller Anforderung lesenden Zugriff auf alle Attribute. Für die Festlegung der relevanten Attribute müssen weitere Projektleiter interviewt werden oder die Anwendung muss eine Schnittstelle bieten, in der ein Benutzer seine für ihn relevanten Informationen wählen kann.

[R17]: Die Anwendung sollte der Benutzergruppe „Zertifizierung“ für das Attribut „Unit Appearance in Printing Data File“ schreibenden Zugriff gewähren und auf alle anderen Attribute lesenden Zugriff.

Mitarbeiter der Zertifizierung (Abschnitt 3.7) dürfen nur in das Feld „Unit Appearance in Printing Data File“ schreiben, denn dort wird angegeben, ob eine Variante ein Label besitzt. Die Mitarbeiter der Zertifizierung benötigen, wie der Projektleiter, ebenfalls nicht alle Informationen einer Variante. Es müssen für eine Festlegung der sichtbaren Felder weitere Interviews geführt werden, oder die Anwendung muss eine weitere Schnittstelle bieten (siehe Beschreibung der Anforderung [R16]).

[R18]: Die Anwendung sollte der Benutzergruppe „Konfigurationsmanagement“ ausschließlich lesenden Zugriff gewähren.

Mitarbeiter des Konfigurationsmanagements (Abschnitt 3.8) greifen auf die aktuelle Excel-basierte Variantenmatrix lesend zu. Dies sollte ebenfalls in der neuen Anwendung festgelegt sein.

[R19]: Die Anwendung muss allen authentifizierten Benutzern lesenden Zugriff gewähren.

Diese Anforderung gilt für alle bisher nicht erfassten Benutzer(gruppen) und legt fest, dass diese in der Anwendung ausschließlich lesenden Zugriff haben. Es sind wie in [R16] und [R11] denkbar mögliche Schnittstellen denkbar. Dadurch können weitere Benutzer(gruppen) angelegt werden und die Nutzer können sich die nur für sie relevanten Informationen anzeigen lassen.

[R20]: Falls der Benutzer Änderungen an einer Variante vornimmt, oder eine neue Variante hinzufügt, sollte das System Zeit und Autor der Änderung speichern.

Mitarbeiter fordern eine, gegenüber der Excel-Tabelle, verbesserte Nachverfolgbarkeit von Änderungen. Daher ist es sinnvoll, dass die Anwendung sowohl den Autor / die Autorin als auch einen Zeitstempel bei Änderung oder Erstellung speichert.

[R21]: Die Anwendung muss dem Benutzer die Möglichkeit bieten, durch den Benutzer ausgewählte Varianten miteinander zu vergleichen.

Mitarbeiter der Variantenverwaltung haben häufig die Aufgabe, Unterschiede zwischen Varianten festzustellen (siehe Abschnitt 3.5.1). Um deren Arbeitsprozess einfacher zu gestalten, sollte die Anwendung die Möglichkeit besitzen, mehrere Varianten miteinander zu vergleichen. Dabei ist es egal, ob zwei oder mehrere Varianten miteinander verglichen werden.

[R22]: *Sobald der Benutzer ausgewählte Varianten miteinander vergleicht, sollte das System die Unterschiede visuell aufbereiten.*

Um Unterschiede zwischen Varianten schneller zu erkennen, sollten diese durch eine geeignete Farbe oder durch ein anderes visuelles Element markiert werden.

[R23]: *Die Anwendung muss sicherstellen, dass Änderungen nur bei CRQ-markierten Varianten möglich sind.*

Wie in Abschnitt 3.3, in den allgemeinen Problemen genannt, dürfen Varianten nur verändert werden, wenn ein CRQ vorliegt. Dies muss die Anwendung sicherstellen.

[R24]: *Die Anwendung muss es dem Benutzer ermöglichen, vorhandene Varianten als CRQ zu markieren.*

Mitarbeiter der Variantenverwaltung (Abschnitt 3.5) müssen Varianten ändern können. Änderungen an bestehenden Varianten dürfen nur durch CRQs erfolgen. Daher müssen die zu ändernden Varianten als diese markiert werden können.

[R25]: *Sobald eine Variante mit CRQ markiert ist, sollte das System dem Benutzer ein Eingabefeld der CRQ-Nummer anzeigen.*

Nachdem ein Eintrag als CRQ markiert wurde, muss dem Benutzer ein Eingabefeld angezeigt werden, in der er eine Nummer einträgt. Diese wird bei erfolgreicher Eingabe gespeichert und im System hinterlegt.

[R26]: *Sobald eine Variante mit CRQ markiert ist und eine CRQ-Nummer besitzt, muss das System dem Benutzer die Möglichkeit bieten, diese Variante zu ändern.*

Nachdem ein Eintrag als CRQ markiert wurde und eine CRQ-Nummer gespeichert wurde, darf ein Eintrag durch den Benutzer geändert werden.

[R27]: *Die Anwendung sollte dem Benutzer die Möglichkeit bieten, zu den als CRQ markierten Varianten einen CRQ-Status hinzuzufügen.*

Sobald eine Variante als CRQ markiert wurde, kann der Benutzer einen Status hinzufügen. Mit Hilfe des Status können andere Mitarbeiter den aktuellen Stand sehen. Dadurch verringert sich die notwendige Kommunikation, erfordert aber zugleich eine Wartung des Status.

[R28]: *Die Anwendung muss dem Benutzer die Möglichkeit bieten, neue Spalten/Attribute zu einer Variante hinzuzufügen.*

Mitarbeiter der Variantenverwaltung (Abschnitt 3.5) dürfen sowohl zu bestehenden als auch neuen Varianten neue Attribute hinzufügen. Dabei werden andere Varianten nicht verändert. Dies ist eine wichtige Anforderung, denn Änderungen an den Attributen finden laut den Mitarbeitern häufiger statt (siehe Abschnitt 3.5.1 und 3.5.2).

[R29]: *Die Anwendung muss dem Benutzer die Möglichkeit bieten, Spalten/Attribute einer bereits existierenden Variante verändern zu können.*

Die Mitarbeiter der Variantenverwaltung (Abschnitt 3.5) dürfen neben dem Hinzufügen von Spalten/Attributen, diese auch umbenennen oder verändern. Weitere Information, siehe [R28].

[R30]: *Die Anwendung muss dem Benutzer die Möglichkeit bieten, Spalten/Attribute einer bereits existierenden Variante zu löschen.*

Ebenfalls müssen die Mitarbeiter der Variantenverwaltung Attribute und Felder einer bestehenden Variante entfernen dürfen. Weitere Information, siehe [R28].

[R31]: *Die Anwendung sollte dem Benutzer die Möglichkeit bieten, ein Attribut einer Variante als „möglicherweise fehlerhaft“ markieren zu können.*

Um eine komplexere Kommunikation zu vereinfachen, wünschen sich einige Mitarbeiter, dass man möglicherweise fehlerhafte Informationen markieren kann. Das markierte Feld oder der markierte Eintrag wird dann von einem anderen Mitarbeiter geprüft [R33].

[R32]: Sobald ein Eintrag als „möglicherweise fehlerhaft“ markiert wurde, muss die Anwendung dem Benutzer ein Eingabefeld für einen benötigten Kommentar bieten.

Zusätzlich zu einer Markierung, muss der Benutzer einen Kommentar angeben. Dort könnte eine Begründung enthalten sein. Das Kommentarfeld wird in der Nähe des möglicherweise fehlerhaften Feldes angezeigt.

[R33]: Sobald ein Eintrag als möglicherweise fehlerhaft markiert wurde, muss die Anwendung einer korrigierender Benutzergruppe dies anzeigen.

Es gibt Benutzergruppen, welche möglicherweise fehlerhafte Einträge kontrollieren können. Diese erhalten eine Information, dass es einen zu prüfenden Eintrag gibt.

[R34]: Die Anwendung muss dem Benutzer die Möglichkeit bieten, einen möglichen fehlerhaften Eintrag zu korrigieren.

Sobald ein Eintrag fehlerhaft ist, darf er korrigiert werden. Hier gilt es zu beachten, dass es nicht zu Änderungen neben einem CRQ kommt. Daher sollte über eine solche Anforderung nochmals diskutiert werden.

[R35]: Die Anwendung muss dem Benutzer die Möglichkeit bieten, vorhandene Varianten in ein von Excel lesbares Dateiformat zu exportieren.

Mehrere Mitarbeiter nutzen VBA-Skripte, basierend auf der Excel-Variantenmatrix. Damit diese arbeitserleichternden Skripte weiterhin funktionieren, muss die Anwendung die Möglichkeit bieten, Inhalte in ein Excel-lesbares Format zu exportieren. Dabei ist es wichtig, dass die Spalten an identischer Position bleiben und die gleichen Namen besitzen.

[R36]: Die Anwendung sollte dem Benutzer die Möglichkeit bieten, Suchvorschläge bei der Eingabe einer Suche anzuzeigen.

Nach einer ersten sehr frühen Prototypvorstellung wurde von den Mitarbeitern der Wunsch geäußert, Suchvorschläge anzuzeigen. Oft wüssten die Mitarbeiter nicht genau, wonach sie suchen. Suchvorschläge würden ihnen dabei helfen.

[R37]: Die Anwendung muss fähig sein, in einem Browser benutzt zu werden.

Die Anwendung muss in einem Browser benutzt werden können, es gibt keine Einschränkungen seitens Bosch bei den Versionen und Browsertypen. Ebenfalls gibt es keine Einschränkungen der minimalen und maximalen Fenstergröße. Daher sollte die Anwendung responsive sein, sich also dem Fenster anpassen können.

[R38]: Die Anwendung muss dem Benutzer die Möglichkeit bieten, Filter und deren Sortierung zu personalisieren.

Jeder der interviewten Mitarbeiter benötigen andere Filter oder nutzen Filter unterschiedlich häufig. Daher ist es wichtig, dass die Mitarbeiter die Sortierung der Filter selbstständig vornehmen können, damit diese den Bedürfnissen und Arbeitsabläufen gerecht werden.

4.2 Nichtfunktionale Produktanforderungen

Es folgen weitere nichtfunktionale Anforderungen und deren Erklärungen.

[NR01]: Mitarbeiter sollten neue Einträge in weniger als acht Minuten hinzufügen können.

Aktuell benötigt der Mitarbeiter der Variantenverwaltung (Abschnitt 3.5) maximal acht Minuten zum Hinzufügen einer neuen Variante. Daher ist es wichtig, dass der Vorgang bei Nutzung der Anwendung nicht mehr Zeit in Anspruch nimmt.

[NR02]: Nach Eingabe eines Suchbegriffs oder der Änderung des Filters sollte die Anwendung innerhalb einer Sekunde die Ergebnisse anzeigen.

Für einige Mitarbeiter ist es wichtig, dass die Anwendung schnell reagiert. Es gab in der Vergangenheit eine Anwendung, welche sehr träge war, woraufhin die Benutzbarkeit eingeschränkt war. Es gilt hier aber auch zu beachten, dass die Reaktionsgeschwindigkeit ebenfalls von der Infrastruktur

(Hardware, Netzwerkauslastung, etc.) abhängig ist und nicht immer direkt beeinflussbar.

[NR03]: *Mitarbeiter müssen Änderungen an den Varianten und deren Attributen vornehmen können, ohne dass die Anwendung selbst verändert werden muss.*

Diese Anforderung spielt eine Schlüsselrolle bei der Auswahl einer geeigneten Datenbank sowie der Architektur. Sie wird von den Mitarbeitern als sehr wichtig angesehen, denn Änderungen im Allgemeinen sind schon aufwendig. Wenn zusätzlich die Anwendung oder die Datenbank manuell geändert werden muss, steigt der Aufwand enorm. Die Anwendung muss flexibel genug sein, Änderungen während des Betriebs zu integrieren.

Änderungen werden von den Mitarbeitern der Variantenverwaltung durchgeführt (Abschnitt 3.5). Änderungen beinhalten das Verändern, Hinzufügen und Löschen von Attributen einer oder mehrerer Varianten.

[NR04]: *Bei der Benutzung der Anwendung sollte der Benutzer nicht horizontal scrollen müssen, um Inhalte vollständig lesen zu können.*

Ein Problem, das in Abschnitt 3.3 genannt wird, ist der Verlust der Orientierung beim horizontalen Scrollen. Dieses soll durch diese Anforderung gelöst werden.

[NR05]: *Mitarbeiter müssen die Anwendung weltweit benutzen können, ohne dass es zu Einschränkungen der Geschwindigkeit oder Benutzbarkeit kommt.*

Da die Mitarbeiter bei Bosch über mehrere Kontinente verteilt arbeiten, muss sichergestellt werden, dass die Anwendung auch in anderen Ländern und Zeitzonen funktioniert. Ebenfalls muss sichergestellt werden, dass sehr viele Mitarbeiter zur gleichen Zeit ohne Einschränkung auf die Anwendung zugreifen können.

4.3 Sicherheit

In diesem Abschnitt werden nötige Sicherheitsmechanismen erläutert und weitere Anforderungen genannt.

Benutzer müssen sich für die Nutzung der Variantenmatrix authentifizieren. Dies kann über einen einfachen Login erfolgen, bei dem ein eindeutiger Benutzername und ein Passwort genutzt wird. Da mit dem Login eventuell personenbezogene Daten verarbeitet und übertragen werden, muss sichergestellt werden, dass diese Daten verschlüsselt werden. Da es ein betriebsinternes Tool ist, wird davon ausgegangen, dass eine einfache Verschlüsselung ausreichend ist.

Anschließend werden weitere Informationen zu einer möglichen Registrierung und Benutzerverwaltung gegeben.

[SR01]: *Die Anwendung muss das Passwort ausreichend sicher verschlüsselt speichern.*

Es darf kein Passwort im Klartext gespeichert werden. Ein ausreichend sicherer Verschlüsselungsalgorithmus muss gewählt werden. So sollen die Daten für Angreifer nutzlos sein.

[SR02]: *Die Anwendung muss sicherstellen, dass alle Daten verschlüsselt übertragen werden.*

Ebenfalls müssen alle Daten verschlüsselt übertragen werden. Es soll dadurch verhindert werden, dass Informationen während der Übertragung von Dritten gelesen oder gar verändert werden können. Es empfiehlt sich daher eine HTTPS/TLS²¹ Verschlüsselung für die Übertragung der Daten.

[SR03]: *Die Anwendung muss dem Benutzer die Authentifizierung entziehen, sobald der Benutzer länger als eine Stunde inaktiv ist.*

Dadurch wird verhindert, dass unbefugte Personen Zugriff auf Daten bekommen, falls der Benutzer vergessen hat, sich abzumelden.

²¹ Transport Layer Security – Verschlüsselungsprotokoll bei Datenübertragung

[SR04]: *Die Anwendung muss dem Benutzer die Möglichkeit bieten, sich über einen Benutzernamen und Passwort anmelden zu können.*

Wie am Anfang dieses Abschnitts beschrieben, sollen sich die Mitarbeiter durch ein Passwort und Nutzernamen authentifizieren.

[SR05]: *Falls der Benutzer authentifiziert ist, muss die Anwendung dem Nutzer die Möglichkeit bieten, sich von der Anwendung abmelden zu können.*

Es sollte somit ebenfalls eine Möglichkeit geben, mit der sich der Benutzer von der Anwendung abmelden kann. Für eine erneute Nutzung muss sich der Benutzer wieder mit seinem Benutzernamen und Passwort authentifizieren.

Für folgende Punkte werden im Rahmen dieser Arbeit keine weiteren Anforderungen festgelegt, sollten aber bei der Implementierung der Anwendung beachtet werden:

Damit sich Benutzer authentifizieren können, müssen diese zuerst angelegt werden. Dies kann manuell erfolgen, sollte aber durch eine Registrierung ermöglicht werden. Dabei werden Nutzernamen und Passwort festgelegt. Ein Nutzer gilt dann als erfolgreich registriert, wenn er von einem Administrator einer Benutzergruppe zugewiesen und freigeschaltet wurde. Die Benutzergruppenzuweisung ist für die Verwaltung der Rechte relevant.

Benutzer müssen ihre Passwörter zurücksetzen können. Der Mechanismus zum Zurücksetzen des Passworts muss sowohl manuell als auch automatisiert erfolgen können. Ein automatisches Zurücksetzen des Passworts muss bei einer möglichen Kompromittierung der Datenbank erfolgen. Manuell kann der Benutzer sein Passwort zurücksetzen, sobald er es vergessen hat.

Für einen solchen Mechanismus muss wahrscheinlich eine E-Mail des Benutzers gespeichert werden.

Ebenfalls muss es Mechanismen zum Löschen von Benutzern geben. Dabei müssen alle personenbezogenen Daten entfernt werden.

4.4 Einschränkung von Anforderungen

Ein Ziel dieser Arbeit ist es, einen Prototyp zu entwickeln und diesen anschließend zu evaluieren. Der Prototyp ist ein Versuchsmuster, der in diesem Fall die Möglichkeit einer Verbesserung der Arbeitswerkzeuge prüfen soll, aber nicht alle Anforderungen umsetzt.

Wie in Abschnitt 3.9 gezeigt, nutzen die Mitarbeiter die Variantenmatrix hauptsächlich als Informationsquelle. In den Interviews und Gesprächen mit den Mitarbeitern wurde festgestellt, dass sie vorwiegend die Such- und Filterfunktion von Excel nutzen, um Varianten zu finden. Der Prototyp wird daher sein Hauptaugenmerk auf diese Funktionen sowie eine intuitive Vergleichsmöglichkeit legen. Er soll außerdem wichtige grundlegende Funktionen wie Autorisierung, Datenexport sowie Datenanzeige bieten. Daher werden in diesem Abschnitt die bisher genannten Anforderungen aus den Abschnitten 4.1, 4.2 und 4.3 auf für den Prototyp relevante Anforderungen reduziert und sortiert. Die Sortierung wurde zusammen mit Mitarbeitern von Bosch vorgenommen.

Der erste Abschnitt 4.4.1 dieses Unterkapitels listet die relevanten Anforderungen, sortiert nach ihrer Bedeutsamkeit. Der zweite Abschnitt 4.4.2 zeigt zwei Anwendungsfälle (engl.: Use Cases), welche einige der genannten Anforderungen umsetzen.

4.4.1 Relevante und sortierte Anforderungen an einen Prototyp

Nachfolgend werden die ausschließlich für einen Prototyp relevanten Anforderungen sortiert genannt. Wie zuvor erwähnt, werden nicht alle bisher beschriebenen Anforderungen vom Prototyp umgesetzt. Wichtige Anforderungen werden zuerst genannt, unwichtige zuletzt. Die eindeutige Reihenfolge der Anforderungen hilft bei der Entwicklung durch eine Gewichtung der Aufgaben. Ebenfalls erhalten Stakeholder schon nach kurzer Entwicklungszeit einen Prototyp, der wichtige Funktionen besitzt. Die Reihenfolge der Anforderungen wurde zusammen mit Mitarbeitern von Bosch festgelegt.

[R37]	Sehr wichtige Anforderungen
[R01]	
[R02]	
[R03]	
[R07]	
[NR03]	
[R05]	
[R06]	
[R36]	
[R04]	
[R08]	
[R09]	
[NR02]	Wichtige Anforderungen
[SR04]	
[SR05]	
[R12]	
[R19]	
[R10]	
[R36]	
[R21]	
[R22]	
[NR04]	Nützliche Anforderungen
[NR05]	
[SR01]	
[SR02]	
[SR03]	

Tabelle 2 Anforderung an den Prototyp, sortiert

4.4.2 Exemplarische Use Cases

In diesem Abschnitt werden zwei exemplarische Anwendungsfälle (engl.: Use Cases) der Anwendung in Form einer Tabelle aufgezeigt. Use Cases sind eine Beschreibung einer Interaktion zwischen Akteuren und dem System oder der Anwendung, durch das der Hauptakteur ein bestimmtes Ziel

erreicht. Use Cases beschreiben das extern wahrnehmbare Systemverhalten. Sie repräsentieren funktionale Anforderungen. Aus ihnen können Testfälle abgeleitet werden (Schneider, 2018).

Der Aufbau der Tabellen ist in Anlehnung an Cockburn, A.: Writing Effective Use Cases (2001 S. 65ff.). Der Use Case 1 beschreibt das Suchen und Filtern von Daten in der Anwendung. Der Use Case 2 beschreibt das Vergleichen von mehreren Varianten. Die Anwendungsfälle setzen nicht alle genannten Anforderungen aus Abschnitt 4.4.1 um.

In dieser Arbeit kann nicht auf alle Use Cases innerhalb des Prozesses eingegangen werden, daher werden hier nur die zwei häufigsten Anwendungsfälle (siehe Abschnitt 3.9) aufgezeigt.

UC1	Suchen / Filtern
Umfeld	An einem Arbeitsplatz (Computer) beim Kunden
Systemgrenze	Alles außerhalb der SW
Ebene	Hauptebene
Hauptakteure	Mitarbeiter
Stakeholder u. Interessen	Mitarbeiter möchte schnell und übersichtlich Ergebnisse zu seiner Suchanfrage angezeigt bekommen
Voraussetzungen	Es gibt einen Suchbegriff, nach dem gesucht werden soll Für jeden Suchbegriff/Filter gibt es eine Antwort von der Datenbank Das System ist voll funktionsfähig und mit einer Datenbank verbunden
Garantie	Es wird ein Suchergebnis angezeigt.
Erfolgsfall	Es wird mindestens ein Eintrag aus der Datenbank gefunden
Auslöser	Mitarbeiter öffnet die SW und möchte nach einer Variante suchen.
Beschreibung	1. Mitarbeiter öffnet den Bereich 'suchen' 2. System zeigt Maske für die Suche und Filter an 3. Mitarbeiter gibt seinen Suchbegriff ein 4. Mitarbeiter startet die Suche 5. System zeigt Ergebnisse an. 6. Mitarbeiter kann Suche/Filter ggf. wieder zurücksetzen
Erweiterung	3a. WENN der Mitarbeiter einen Filter wählt, DANN (5) 5a. WENN System keine Ergebnisse findet, DANN 5a1. zeige an 'Keine Ergebnisse gefunden'
Technologie	Keine weiteren Technologievarianten

Use Case 1: Suchen und Filtern

Der Use Case 1 setzt die Anforderungen R03, R04, R06, R07 und R09 um. Das Suchen nach Varianten wird von den Mitarbeitern aller Gruppen genutzt (siehe Abschnitt 3.9).

Der nachfolgende Use Case zeigt den Anwendungsfall des Vergleichens von mindestens zwei Varianten.

UC2	Vergleichen
Umfeld	An einem Arbeitsplatz (Computer) beim Kunden
Systemgrenze	Alles außerhalb der SW
Ebene	Hauptebene
Hauptakteure	Mitarbeiter
Stakeholder u. Interessen	Mitarbeiter möchte übersichtlich verschiedene (2) Varianten miteinander vergleichen
Voraussetzungen	Es gibt mindestens zwei Varianten, die miteinander verglichen werden können
Garantie	Das System ist voll funktionsfähig
Erfolgsfall	Die Varianten werden visuell gegenübergestellt.
Auslöser	Die Varianten werden visuell gegenübergestellt und Unterschiede hervorgehoben.
Beschreibung	Mitarbeiter öffnet die SW und will gelistete Varianten vergleichen.
	1. Mitarbeiter öffnet Liste von Varianten
	2. System zeigt Liste von Varianten an
	3. Mitarbeiter wählt mindestens zwei Varianten aus
	4. Mitarbeiter öffnet Vergleichsübersicht
	5. System stellt Varianten gegenüber und hebt Unterschiede visuell hervor
	6. Mitarbeiter kann ggf. Vergleich wieder beenden/schließen
Erweiterung	Keine weiteren Erweiterungen
Technologie	Keine weiteren Technologievarianten

Use Case 2: Vergleichen

Der Use Case 2 setzt die Anforderungen R21 und R22 um. Das Vergleichen von Varianten wird sehr häufig von den Mitarbeitern der Variantenverwaltung durchgeführt (siehe Abschnitt 3.5), ebenfalls auch von Mitarbeitern der anderen Gruppen (siehe Abschnitt 3.9). Ziel ist es, dass die Anwendung eine einfache Möglichkeit bietet, Unterschiede zwischen Varianten feststellen zu können, ohne dass besondere Excel-Kenntnisse nötig sind.

5 Konzeptentwicklung

In diesem Kapitel geht es um die Entwicklung eines Anwendungskonzeptes und den Vergleich zu einem bereits existierenden Prototyp. Die Anwendung soll die in den vorherigen Kapiteln festgelegten Anforderungen (siehe Abschnitt 4.4.1) umsetzen.

Zuerst wird in Kapitelabschnitt 5.1 ein Informationsflussmodell einige Daten der Variantenmatrix und deren Zusammenhänge innerhalb des in Kapitel 3 vorgestellten Prozess aufzeigen. Es folgt in Abschnitt 5.2 ein visuelles Konzept, das den Mitarbeitern präsentiert wurde. Abschnitt 5.3 befasst sich mit einer möglichen Architektur sowie unterschiedlichen Datenbanken. Es folgt ein weiterer Abschnitt 5.4, der das mögliche Konzept mit einem bereits existierenden Prototyp vergleicht und Unterschiede zwischen beiden feststellt.

5.1 Informationsflussmodell

Im nachfolgender Abbildung 5.1 wird ein Unified Modeling Language (UML) Informationsfluss dargestellt. Das Informationsflussmodell modelliert Kanäle zwischen Entitäten und Informationen, die darüber ausgetauscht werden. Informationsflusskanäle werden durch gestrichelte Linien, die mit `<<flow>>` markiert sind, angegeben. Sie sind gerichtet. Die „fließenden“ Informationen stehen ebenfalls an den Linien. Informationen können durch andere Informationen oder Attribute repräsentiert werden. Sie stehen unter der Informationsflusssdarstellung.

Die Entitäten sind die in den Kapiteln 3 erwähnten Excel-Tabellen der unterschiedlichen Abteilungen. Informationen werden aufgrund einer besseren Übersicht abgekürzt, Attribute werden ausgeschrieben. So ist zum Beispiel die Information „L“ eine *Extended Information* mit dem Attribut *Plant*, und der Information *Core Information (C)*. Die Information „C“, also *Core Information* besteht wiederum aus den Attributen *Carline* und *Region*.

Es können nicht alle Informationen übersichtlich in einem Diagramm dargestellt werden. So werden die jeweils wichtigsten Informationen der Variantenmatrix und anderer Excel-Tabellen dargestellt.

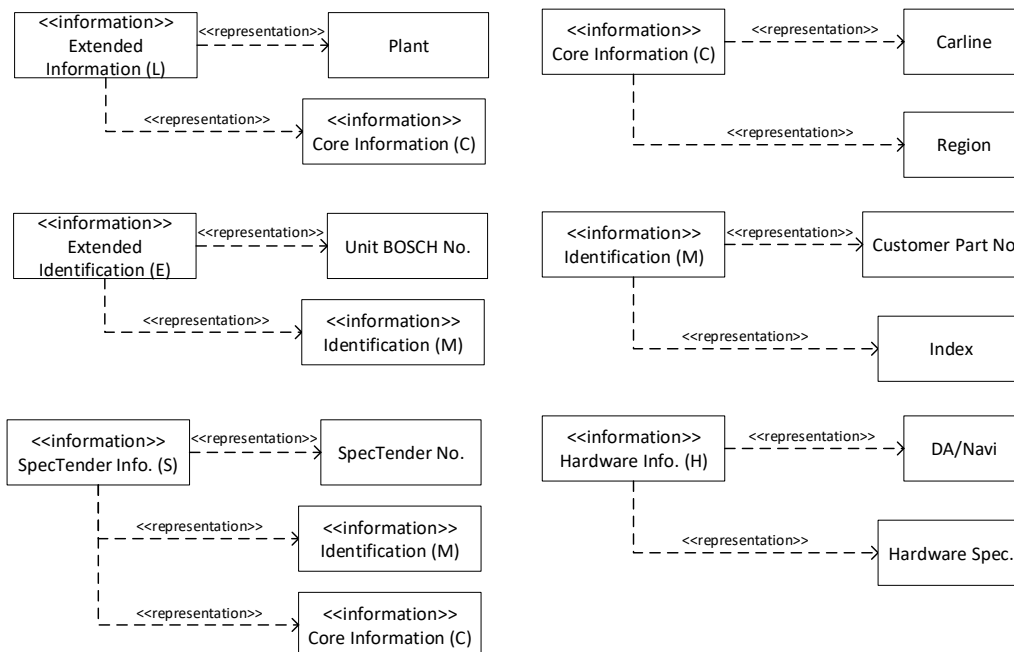
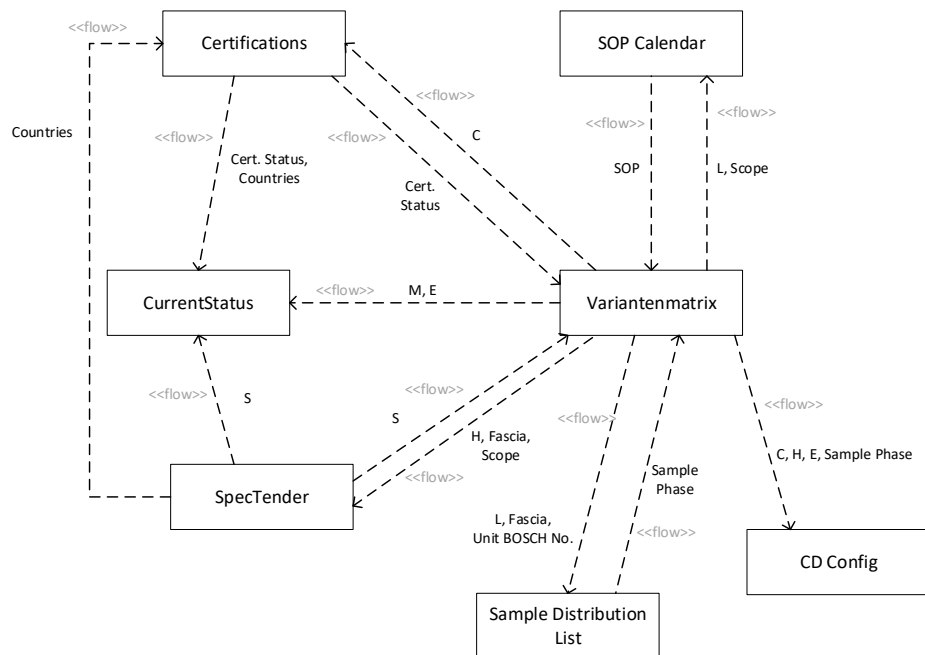


Abbildung 5.1 UML Informationsfluss zwischen Dateien der Abteilungen

5.2 Visuelles Konzept

Ziel der Anwendung ist es, dem Benutzer eine einfach zu bedienende Oberfläche zu bieten, die dennoch alle Möglichkeiten des Suchens und Filterns einer Excel-Tabelle bietet und die geforderten Anforderungen aus Abschnitt 4.4.1 umsetzt. Die Such- und Filterfunktion sollte auf das nötigste reduziert sein, um den Benutzer beim Start der Anwendung nicht zu überfordern. Die Suche und der Filter sollen erweitert werden können, um versierten Benutzern weitere Funktionen und Auswahlmöglichkeiten bieten zu können. Ein zusätzliches Vergleichsfeature ermöglicht das einfache Vergleichen von Varianten der Variantenmatrix. Ebenfalls sollte die Anwendung dem Benutzer ein gutes visuelles Feedback geben.

Das Suchen nach Einträgen und Informationen ist eine der häufigsten genutzten Funktionen im Internet²². Wie in Anforderung R36 gefordert, bieten Suchmaschinen wie Google oder Bing eine intelligente Suchvervollständigung (Abbildung 5.2).

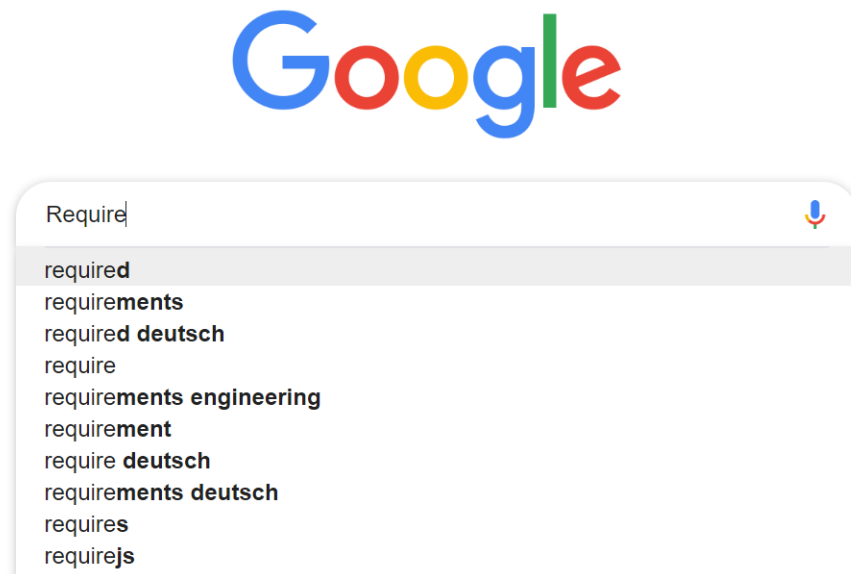


Abbildung 5.2 Suchvervollständigung Google - Quelle: www.google.de – Abgerufen am 01.06.2019 um 15:11 Uhr

Eine ähnliche Suchvervollständigung wurde mehrfach von den Mitarbeitern gewünscht, da sie oft nicht wissen, wonach sie genau suchen.

²² Quelle: SearchEngineLand: <https://searchengineland.com/google-now-handles-2-999-trillion-searches-per-year-250247> - Abgerufen am 01.06.2019 um 15:05 Uhr

Filter sollen die Suche weiter einschränken können (R07 und R08). Ein Filter soll ein bekanntes Attribut aller Objekte und deren Inhalt als Filterelement zur Verfügung stellen. Ähnlich geht hier zum Beispiel der Preisvergleich Geizhals im Internet vor (Abbildung 5.3).

☐	Hersteller:	AGPTEK Apple Arena Astell&Kern Auvisio Cowon Denver Difrnce Energy Sistem FiiO Grundig iBasso Intenso Lenco Maxell mpman Phillips Pioneer SanDisk Sony takeMS Technaxx Transcend TrekStor Alle
☐	Interner Speicher:	ab 4GB (34) ab 8GB (18) ab 16GB (14) ab 64GB (3) ab 128GB (2) ab 256GB (1)
☐	Audio:	MP3 (34) WMA (34) AAC (26) APE (9) FLAC (14) ALAC (6) WAV (18) AIFF (6) DSD (18)
☐	Schnittstellen:	USB (33) USB 2.0 (29) Bluetooth (14) NFC (13)
☐	Besonderheiten:	UKW-Tuner (22) Aufnahmefunktion (8) wassergeschützt (6) Hi-Res Audio (6)
▼		7 weitere Filter anzeigen Filter zurücksetzen Immer alle Filter anzeigen

Abbildung 5.3 Filterfunktion Preisvergleich Geizhals - Quelle: www.geizhals.de – Kategorie MP3-/MP4-Player (Ausschnitt) – Abgerufen am 01.06.2019 um 14:45 Uhr

Der Filter in oberer Abbildung hat mehrere Filterkategorien, die ein mögliche Attribute eines Objektes sind. Jede Kategorie besitzt Filterelemente, die Inhalte der Objektattribute sind. Es können pro Kategorie ein oder mehrere Elemente aktiviert werden. Durch ein „X“ kann ein Element aus dem aktiven Filter entfernt werden. In runden Klammern sind Mengenangaben vorhanden, wodurch auf die Anzahl der Suchergebnisse rückzuschließen ist. Es lassen sich ebenfalls weitere Filter anzeigen. Der Filter kann mit einem Klick zurückgesetzt werden. Sobald der Nutzer ein Filterelement aus der in Abbildung 5.3 versteckten Filter aktiviert und die Filter wieder minimiert, bleibt immer der aktivierte Filter sichtbar. Filterkategorien können zusätzlich über eine sog. Checkbox aktiviert werden. Es ändert sich dadurch die Ansicht der Suchergebnisse.

Das Vergleichen von verschiedenen Elementen (R21 und R22) ist oft ein Feature von Webshops. Dort werden unterschiedliche Produkte derselben Kategorie nebeneinander dargestellt. Unterschiede werden meistens farblich hervorgehoben, wodurch der Nutzer diese schnell wahrnimmt. Die nachfolgende Abbildung 5.4 zeigt einen solchen Vergleich anhand des Webshops von *Notebooksbilliger*.

	Viewsonic PX747 4K Beamer - UHD, HDR, 3.500 ANSI Lumen,...		Optoma UHD40 4K Beamer - UHD, 2.400 ANSI Lumen,...
	Details Aus Vergleich entfernen In den Warenkorb		Details Aus Vergleich entfernen In den Warenkorb
Preis	931,14 EURO	1.156,55 EURO	
Artikelnummer	A 724899	A 720346	
Kundenbewertung	★★★★★	★★★★★	
Projektionsdaten			
Typ	DLP-Beamer	DLP-Beamer	
Technologie	DLP	DLP	
Helligkeit	3500 ANSI-Lumen	2400 ANSI-Lumen	
Kontrast	12000 : 1	500000 : 1	
Auflösung	3.840 x 2.160	3.840 x 2.160	
Auflösungsform	4K-UHD	4K-UHD	
Bildformat	16:9 (5 wählbar)	16:9	

Abbildung 5.4 Vergleichsfunktion Webshop Notebooksbilliger - Quelle: www.notebooksbilliger.de – Kategorie Beamer – Vergleich von zwei Projektoren (Ausschnitt) – Abgerufen am 01.06.2019 um 17:03 Uhr.

Im Vergleich der Abbildung 5.4 werden Unterschiede zwischen den Produkten durch ein kräftiges Orange dargestellt. Sobald es Attribute gibt, die das andere Produkt nicht besitzt, wird dies ebenfalls durch ein Orange gekennzeichnet. Nach Befragung und während der Interviews einiger Mitarbeiter bei Bosch stellte sich heraus, dass sie sich eine ähnliche Funktion wünschen.

Es wurden nun drei unterschiedliche Webseiten und deren Funktionen vorgestellt, die benötigte Features des Prototyps sind. Nachfolgendes Vorführmodell (engl.: Mockup) (Abbildung 5.5) verbindet die vorgestellte Suche und einen Filter.

Das Sucheingabefeld ist präsent oberhalb der Filter platziert, wodurch der Benutzer direkt dazu aufgefordert wird, eine Eingabe zu tätigen. Unter der Suche sind Filter sichtbar. Diese sollen sich, wie in Abbildung 5.3 erweitern oder minimieren lassen. Suchergebnisse in Form von Varianten werden unterhalb der Filter angezeigt. Jedes Suchergebnis soll einen Button oder eine Checkbox besitzen, um sie dem Vergleich hinzuzufügen zu können. Es gibt ein seitliches Menü, über das die Nutzer navigieren oder sich abmelden können.

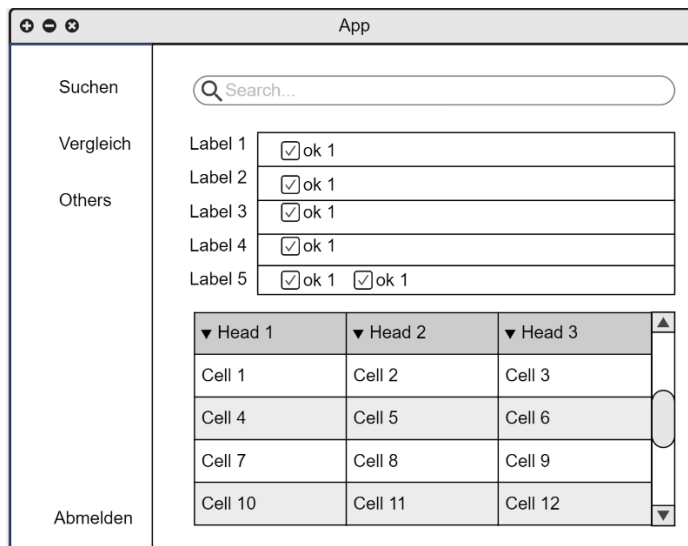


Abbildung 5.5 Mockup Prototyp Suchen / Filtern

Sobald ein oder mehrere Einträge dem Vergleich hinzugefügt wurden, soll dem Anwender dies ebenfalls kenntlich gemacht werden. Neben dem Menü Element „Vergleich“ soll ein Icon auf die Anzahl der Elemente in dem Vergleich hinweisen. Der Vergleich soll sich vom Aufbau nicht sehr von Abbildung 5.4 unterscheiden.

5.3 Architektur

In diesem Abschnitt wird eine mögliche Web Plattform Architektur vorgestellt (5.3.1) und ein Überblick über diese gegeben. In dem weiteren Abschnitt 5.3.2 wird eine mögliche Architektur der Webanwendung vorgestellt. Der letzte Abschnitt 5.3.3 gibt einen Überblick über drei mögliche Datenbanksysteme und deren Vor- und Nachteile.

5.3.1 Web Plattform Architektur

Eines der Anforderungen ist es, dass die Anwendung in einem Webbrowser funktioniert (R37). Ein Ziel dieser Arbeit ist es, die Daten der Excel Variantenmatrix in eine Datenbank zu überführen. Ebenfalls sollen nach Anforderung NR05 die Daten für Benutzer aus unterschiedlichen Regionen verfügbar sein. Dies ist bei der Wahl einer geeigneten Architektur zu berücksichtigen. Stefan Jablonski et al. empfehlen eine vierstufige Web Plattform Architektur (WPA) (Abbildung 5.6). Die vorderste Stufe (*Client Tier*) hat die Möglichkeit, mehrere dahinterliegende Schichten zu gleicher Zeit zu nutzen.

Das Gleiche gilt für die zweite Stufe (*Web Tier*) und der dritten Stufe (*Application Server Tier*). Sie besitzen somit eine *1:n* Beziehung zu benachbarten Stufen. Dadurch wird die Performance, als auch die Skalierbarkeit positiv beeinflusst.

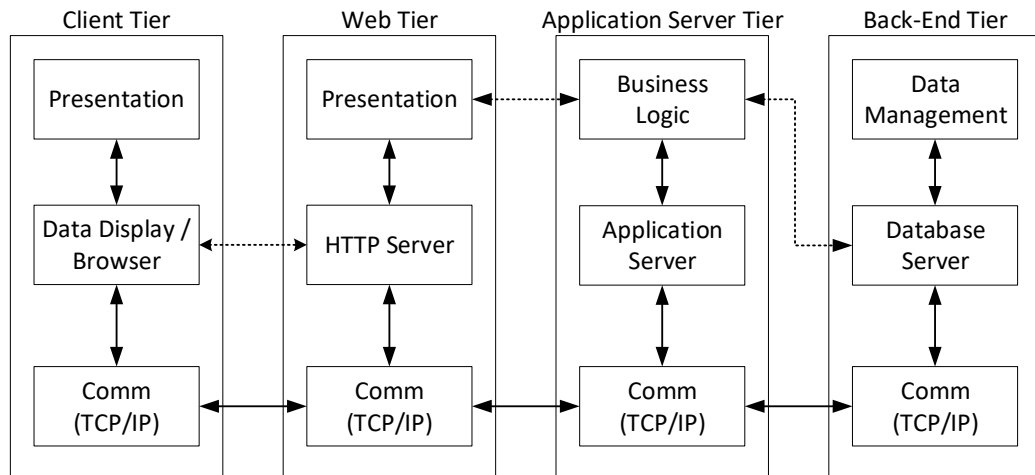


Abbildung 5.6 Vier-Stufen-Architektur - Abbildung in Anlehnung an „Four-tier architecture“ von Stefan Jablonski et al.: Guide to Web Application and Platform Architectures (2004 S. 27)

Jeder der vier Stufen besitzt Vor- und Nachteile. Die Client-Stufe hat ihren Vorteil in der Visualisierung, ist aber für die Datenspeicherung eher ungeeignet. Die Web-Stufe sowie die Server-Applikations-Stufe sind besonders für die Kommunikation als auch für die Ausführung von Business-Logik geeignet, aber ungeeignet für die Visualisierung als auch die Datenspeicherung. Die Back-End-Stufe hat als einzige ihren Vorteil in der Datenspeicherung, ist für die Kommunikation, Ausführung von Business-Logik, als auch Visualisierung eher ungeeignet. Zusammen genutzt bieten sie aber eine große Flexibilität (Jablonski, et al., 2004 S. 28ff.).

5.3.2 Architektur der Webanwendung

Die Webanwendung ist der Teil der Web Plattform Architektur, die von einem Client im Browser ausgeführt wird. Die Architektur der Webanwendung kann einem beliebigen Entwurfsmuster (engl.: Design Pattern) folgen. Die nachfolgende Abbildung 5.7 zeigt ein beispielhaftes Entwurfsmuster anhand eines Model View Controller (MVC) Musters. Das Modell (engl.: *Model*) enthält alle Daten der Anwendung. Es ist unabhängig von der Steuerung (engl.: *Controller*) und der Präsentation (engl.: *View*). Sobald sich das

Modell ändert, wird die abhängige Präsentation benachrichtigt. Die Präsentation zeigt die Inhalte des Modells an. Der Benutzer kann mit ihr interagieren. Bei einer Interaktion benachrichtigt sie die Steuerung, welche das Modell ändert.

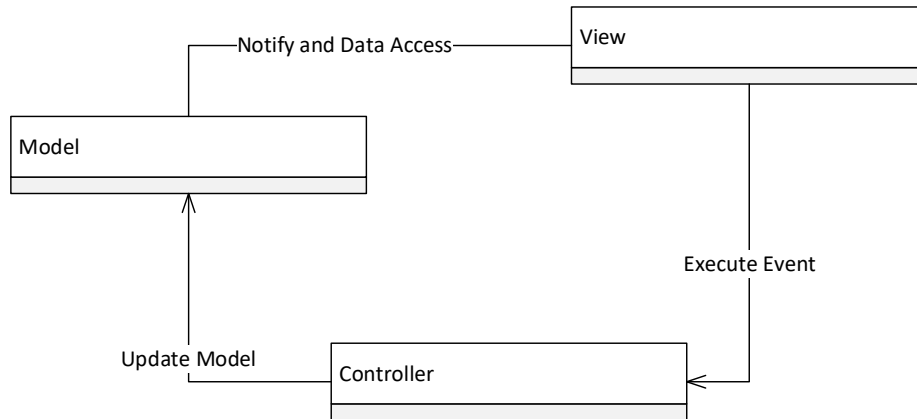


Abbildung 5.7 Model View Controller Entwurfsmuster

5.3.3 Datenbanksysteme

In diesem Abschnitt werden jeweils Vor- und Nachteile der in Abschnitt 2.3 genannten Datenbanksysteme aufgezeigt.

Relationale Datenbanken

Vorteile: Die Mitarbeiter der Entwicklungsabteilung für interne Tools sind mit der Entwicklung von relationalen Datenbanken und der Abfragesprache SQL vertraut. Ein festes Datenbankschema ermöglicht ein einfaches Abfragen der Daten. Wenn ein festes Schema benutzt wird, könnten Heuristiken gefunden werden, die bestimmte Regeln für Attribute der Datenbankobjekte festlegen. Fehlerhafte Einträge könnten so vermieden werden. Die Webanwendung kann aufbauend auf dem Schema mit den verwendeten Entitäten arbeiten. Die Schnittstelle zwischen der Webanwendung und dem Server (siehe Abbildung 5.6) besitzt ebenfalls ein festes Schema, wodurch der Aufbau der übertragenden Daten vorhersehbar ist.

Nachteile: Ein festes Datenbankschema ermöglicht es nicht, sehr einfach Änderungen vorzunehmen. Änderungen am Schema müssten zuerst von Datenbankspezialisten überprüft werden, damit es zum Beispiel nicht zu Redundanzen kommt. Je nach Aufbau der Schnittstellen müssten diese sowie die Webanwendung angepasst werden. Sollte es Heuristiken geben,

müssten diese ebenfalls angepasst oder neu geschrieben werden. Eine Änderung ist somit sehr kritisch und um einiges aufwändiger, als eine Umbenennung einer bestehenden oder Einführung einer neuen Spalte in der Excel-Variantenmatrix. (Parker, et al., 2013)

Document Stores

Vorteile: Die Document Stores gehören zu den nichtrelationalen Datenbanken. Wie in Abschnitt 2.4.1 erläutert, speichern diese zu einem eindeutigen Schlüssel zugeordnete semistrukturierte Datensammlungen, die keinem festen Schema entsprechen müssen. Die Daten können sich somit sehr stark verändern, ohne dass Anpassungen nötig sind. Daher könnte die Variantenverwaltung einer beliebigen Variante neue Attribute vergeben, ohne dass die Anwendung angepasst werden muss oder Datenbankexperten dies autorisieren müssen (Parker, et al., 2013). Die Erstellung einer NoSQL Datenbank ist aufgrund der Schemalosigkeit sehr einfach.

Nachteile: Einer der Vorteile ist auch gleichzeitig ein Nachteil. Da die Document Stores keinem festen Schema folgen müssen, ist darauf zu achten, wie Attribute benannt sind, sowohl beim Abfragen als auch beim Ändern oder Hinzufügen von Varianten. Da kein SQL als Abfragesprache verwendet wird, müssen sich die Mitarbeiter der Entwicklungsabteilung in die benötigte Abfragesprache einarbeiten. Es muss für eine ausreichende Indexierung der Daten gesorgt werden, um diese einfach wieder zu finden. Die Benutzung von Heuristiken ist aufgrund der starken Variabilität der Daten nur sehr eingeschränkt nutzbar.

Graphdatenbanken

Vorteile: Graphdatenbanken gehören wie die Document Stores zu den nichtrelationalen Datenbanksystemen. Wie in Abschnitt 2.4.2 erläutert, basieren Graphdatenbanken auf Knoten und Kanten. Dadurch können relativ einfach Beziehungen zwischen Varianten der Variantenmatrix gezogen werden. Eine Nachverfolgbarkeit bei Änderung oder Vererbung von Varianten könnte eine Kante in dem Graphen darstellen. Knoten, also Varianten, können, müssen aber nicht einem festen Schema folgen. Hier gelten die gleichen Bedingungen wie bei den Document Stores. Relationale

Datenbanken könnten genauso Beziehungen speichern, Graphdatenbanken lassen sich aber im Gegensatz zu relationalen Datenbanken einfacher traversieren (Edlich, et al., 2010 S. 9).

Nachteile: Wenn kein festes Schema gewählt wird, gelten die gleichen Nachteile wie bei den Document Stores. Graphdatenbanken sind noch relativ neu, daher gibt es keine standardisierten Abfragesprachen. Eine Einarbeitung scheint auf dem ersten Blick ebenfalls komplexer gegenüber Document Stores. Es sind ebenfalls im Rahmen dieser Masterarbeit keine Beziehungen zwischen den Varianten hinterlegt, diese müssen von den Mitarbeitern zuerst erstellt werden.

5.4 Vergleich zu existierender Lösung

Es gibt bereits einen Prototyp „Ernie 2.0“, der von einem externen Unternehmen erstellt wurde. Er basiert auf Java, einer relationalen Datenbank und benutzt das Web-Framework Vaadin. Er soll in späterer Ausbaustufe Heuristiken bieten, die vom Benutzer editiert werden können.

Vaadin ist ein Web-Framework, welches serverseitiges Rendering ermöglicht. Ein Großteil der Logik liegt somit auf dem Server. Es werden neu anzuzeigende HTML-Elemente an die Clients gesendet. Dem Benutzer wird nach einer Interaktion die darauffolgende Systemreaktion erst nach der Datenverarbeitung des Servers und Übertragung der neuen HTML-Elemente angezeigt. Dies kann bei einer langsameren Verbindung oder einer hohen Serverlast zu einem trägen User Interface führen. Das Framework ist aktuell nicht sehr häufig in Benutzung²³, weshalb es schwierig werden könnte, komplexere Probleme zu lösen oder die Software zu warten.

Eine relationale Datenbank ermöglicht es nicht so einfach, Attribute von Varianten zu ändern. Dafür müssen wohlmöglich die zugehörigen Heuristiken angepasst werden sowie der komplette Code der Serverapplikation.

Mit der Programmiersprache Java sind einige Mitarbeiter vertraut.

Die Reaktionsgeschwindigkeit sowie ein nicht intuitiv zu bedienendes Benutzerinterface wurden von einigen Mitarbeitern stark kritisiert.

²³ Vaadin wird von aktuell 6.983 Projekten weltweit genutzt. Quelle: <https://github.com/vaadin/framework> - Abgerufen am 11.06.2019 um 22:59 Uhr

6 Implementation

Dieses Kapitel befasst sich mit der Implementation eines Prototyps. Dieser soll die in Abschnitt 4.4.1 erfassten Anforderungen umsetzen sowie der in Kapitel 5 entwickelten Konzepte implementieren.

Die Webanwendung als auch die Webserveranwendung wurde in TypeScript geschrieben. TypeScript wird von Microsoft entwickelt und bietet die Möglichkeit, objektorientierten und stark typisierten Code zu schreiben, der durch einen Compiler in JavaScript-Code kompiliert wird. JavaScript ist eine Skriptsprache, die für die Generierung und Veränderung von HTML und CSS entwickelt wurde. JavaScript kann auch außerhalb eines Browsers in einer geeigneten Laufzeitumgebung genutzt werden. Außer der Namensgebung hat JavaScript keine Gemeinsamkeiten mit Java. Jeder JavaScript-Code kann auch von TypeScript verwendet werden.

Der Abschnitt 6.1 befasst sich mit der Implementierung und Beschreibung verwendeter Frameworks der Webanwendung. In Abschnitt 6.2 wird eine der in Abschnitt 5.3.3 aufgeführten Datenbanksysteme gewählt und der Aufbau der Webserveranwendung beschrieben. Das gewählte Datenbanksystem sowie die Anforderungen aus Abschnitt 4.4.1 schränken die Nutzung von heuristischen Regeln sehr stark ein. Dies wird ebenfalls in Abschnitt 6.2 genauer erläutert.

6.1 Webanwendung

Wie zuvor geschrieben, wurde für die Webanwendung eine TypeScript-Applikation geschrieben. Diese nutzt die populären Frameworks React und Redux. React wird für die View-Ebene genutzt (vgl. Abbildung 5.7), Redux gibt eine MVC-ähnliches Entwurfsmuster vor (siehe Abschnitt 5.3.2). React ermöglicht es hierarchische View Komponenten zu erstellen und zu verwalten. Zusätzlich ermöglicht es in selbst großen Anwendungen eine performante Darstellung durch ein sog. virtuelles DOM (Stefanov, et al., 2017). Das Framework Redux wird in dem Abschnitt 6.1.1 genauer erläutert.

Der nachfolgende Absatz präsentiert und erklärt einige Screenshots des Prototyps. Umgesetzte Anforderungen werden ebenfalls genannt. In den Screenshots sind prozessinterne Informationen unkenntlich gemacht. Die Webanwendung (R37) des Prototyps besitzt die Fähigkeit der Authentifizierung (siehe Abbildung 6.1) von Nutzern (SR04). Nutzer können sich an- und wieder abmelden (SR05). Der Prototyp besitzt keine Benutzerverwaltung. Es sind Demodaten hinterlegt, mit denen sich der Nutzer authentifizieren kann und lesenden Zugriff auf die Daten erhält (R12, R19).

Abbildung 6.1 Prototyp Authentifizierung

Formularfelder werden bei falscher Eingabe mit einem roten Rahmen umrandet, es gibt eine zusätzliche Fehlermeldung unterhalb eines Formulars. Der Benutzer bekommt somit eine direkte, dem jeweiligen Formularfeld zugeordnete, Fehlermeldung. Dadurch wird bei Formularen die Wahrscheinlichkeit des korrekten Ausfüllens erhöht (Jacobsen, 2014 S. 166). Nach erfolgreichem Login wird der Nutzer zur Suche weitergeleitet (siehe Abbildung 6.2). Dem Benutzer wird auf der linken Seite ein Menü präsentiert, mit dem er schnell zwischen möglichen Hauptfunktionen wechseln oder sich wieder abmelden kann. Der Prototyp besitzt lediglich die Funktionen „Suchen“ (R03), „Vergleichen“ (R21) und „Abmelden“ (SR05). Die Hauptseite „Suchen“ bietet ein Sucheingabefeld und einen Filter (R07), der sich unter dem Button „Filter öffnen“ versteckt. Unterhalb der Such- und Filtermaske werden die Informationen, in diesem Fall die Varianten, angezeigt (R02). Es gibt ebenfalls eine Export-Funktion, die über den gleichnamigen Button erreichbar ist (R21). Der Export erstellt eine Excel-Datei (.xlsx), welche alle sichtbaren und von dem Filter oder der Suche eingegrenzten Varianten enthält. Ist kein Filter oder keine Suche aktiv, werden somit über diese Funktion

alle Varianten aus der zugrundeliegenden Datenbank in eine Excel Datei exportiert. Der Aufbau der Excel Datei ähnelt dem der Variantenmatrix. Bei einem Klick auf eine der Varianten in den Suchergebnissen gelangt der Nutzer auf eine Detailseite, die alle Informationen der Variante aufschlüsselt.

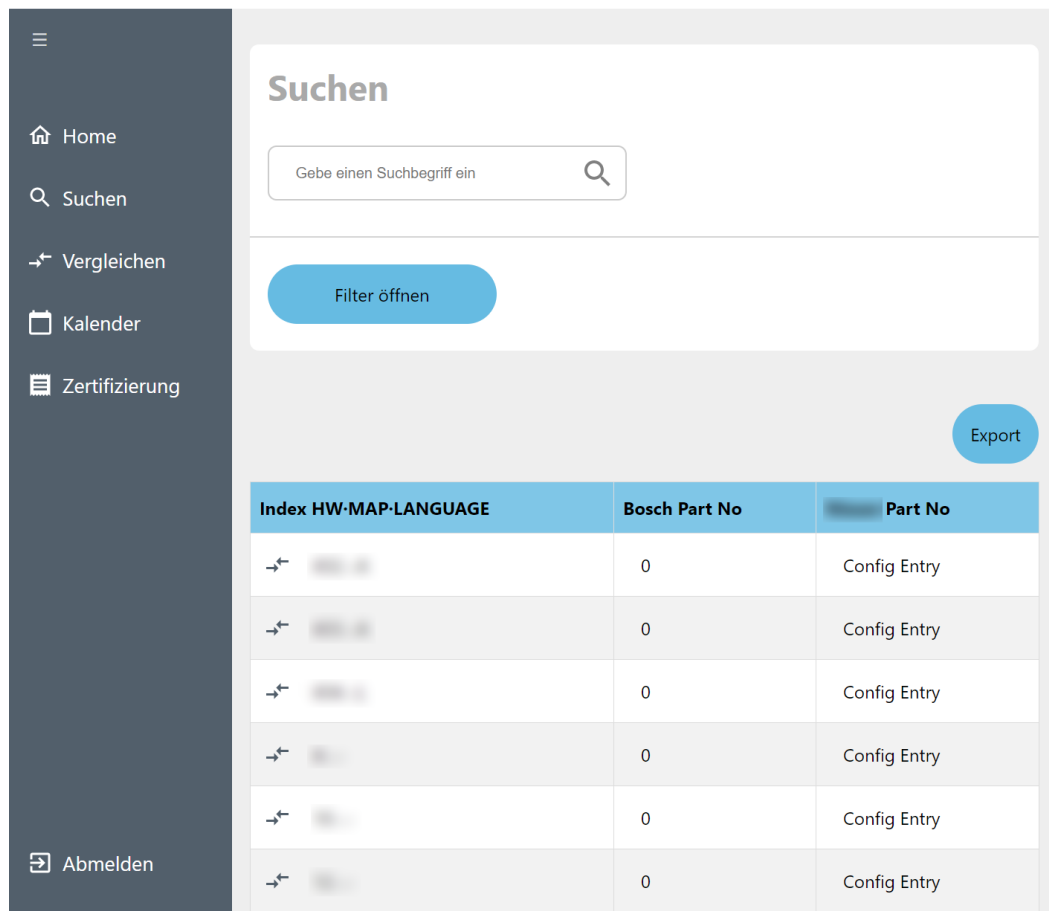


Abbildung 6.2 Prototyp Suche und Filter Startseite

Jede Variante in den Ergebnissen (siehe Abbildung 6.2) besitzt ein „Vergleichen“ Icon, das ein beliebiges Element zu einem Vergleich (R21) hinzufügt. Dabei verändert sich das Icon in einen rot-orangefarbenen Ton.

Alle sichtbaren Inhalte der Anwendung sind markierbar und können anschließend kopiert werden(R10). Der Prototyp ist in Englisch und Deutsch übersetzt (R01). Er richtet sich nach der vom Browser vorgegebenen Spracheinstellung.

Die nachfolgende Abbildung zeigt eine aktive Suche und einen ausgewählten Filter.



Abbildung 6.3 Prototyp Suche und Filter aktiv

Es können, wie in Abbildung 6.3 sichtbar, mehrere Suchen gleichzeitig durchgeführt werden (R05). Sucheingaben können einzeln über ein „X“ gelöscht werden (R06). Dies gilt ebenfalls für die Filter (R08). Sollte es für eine Sucheingabe und/oder einen Filter keine Ergebnisse geben, wird dies unterhalb der Such- und Filtermaske angezeigt (R04, R09), ähnlich zu der Information in Abbildung 6.1. In Abbildung 6.3 besitzen die Filterkategorien eine Checkbox, welche bei Aktivierung eine weitere Spalte in den Ergebnissen sichtbar macht. Es lassen sich dadurch schon in den Suchergebnissen genauere Details der Varianten anzeigen.



Abbildung 6.4 Prototyp Suchvorschläge

In Abbildung 6.4 werden zu einer Eingabe Suchvorschläge gemacht. Diese können ausgewählt werden und werden einer Suche direkt hinzugefügt (R36). Nach Eingabe einer Suche oder Änderung des Filters, dauert es nicht länger als 200ms, bis ein Ergebnis sichtbar wird (NR02). Dabei ist zu

beachten, dass der Prototyp vollständig auf einem ausreichend schnellen Gerät läuft und keine Netzwerklatenz vorhanden ist.

Die nachfolgende Abbildung 6.5 zeigt das Vergleichen von Varianten.

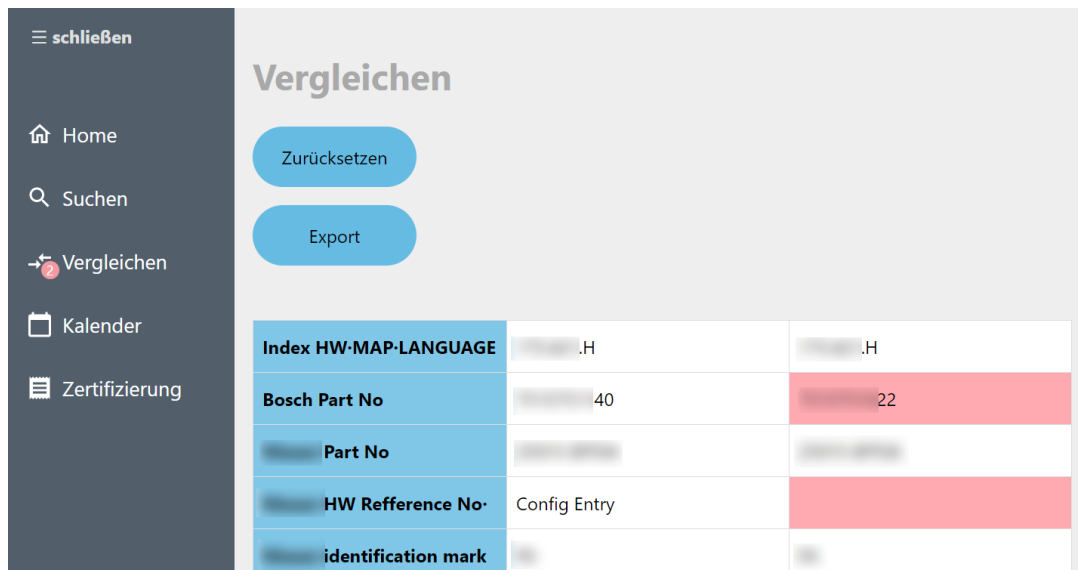


Abbildung 6.5 Prototyp Vergleichen von Varianten

Die Unterschiede von zwei oder mehreren Varianten (R21) werden durch ein helles Rot sichtbar gemacht (R22). Verglichen wird immer gegenüber dem ersten Element. Das Menü enthält ebenfalls einen Indikator, wie viele Elemente dem aktuellen Vergleich hinzugefügt wurden. Die Elemente in dem Vergleich lassen sich ebenfalls exportieren. Der Vergleich kann komplett zurückgesetzt werden. Der Vergleich enthält alle Felder der Varianten. Sollte eine Variante ein Feld nicht besitzen, wird dies ebenfalls als Unterschied dargestellt.

Die Webapplikation enthält einige weitere Features, welche die Usability verbessern. Diese zu erläutern, würde den Rahmen der vorliegenden Arbeit überschreiten. Die nachfolgende Abschnitt geht genauer auf das verwendete Framework Redux ein, das in den Grundzügen dem MVC Entwurfsmuster (siehe Abschnitt 5.3.2) entspricht.

6.1.1 Redux

„Redux ist ein vorhersehbarer Zustandscontainer für JavaScript Anwendungen“²⁴. Genauso wird Redux in der offiziellen Dokumentation beschrieben.

²⁴ Quelle: <https://redux.js.org/introduction/getting-started> - 02.06.2019 - Übersetzt

Hauptziel von Redux ist es, für Konsistenz und Vorhersehbarkeit von Daten in Anwendungen zu sorgen. Eine mit Redux entwickelte Anwendung gleicht einem großen Zustandsautomaten. Redux trennt die Verantwortlichkeiten der Zustandsverwaltung in kleinere Einheiten (Garreau, et al., 2018 S. 5):

- Der *Store* hält den Zustand der Anwendung in einem Objekt. Der Zustand besitzt die Struktur eines Baums.
- Der *Store* kann nur mit *Actions* verändert werden. *Actions* sind Objekte, die ein Ereignis beschreiben.
- *Reducer* sind Funktionen, die einen bestimmten Teil (Blatt) des *Stores* und eine *Action* nehmen, diesen dann durch einen neuen Zustand ersetzen.

Nachfolgende Abbildung 6.6 gibt einen etwas genaueren Überblick auf die in dieser Arbeit verwendete Redux Architektur.

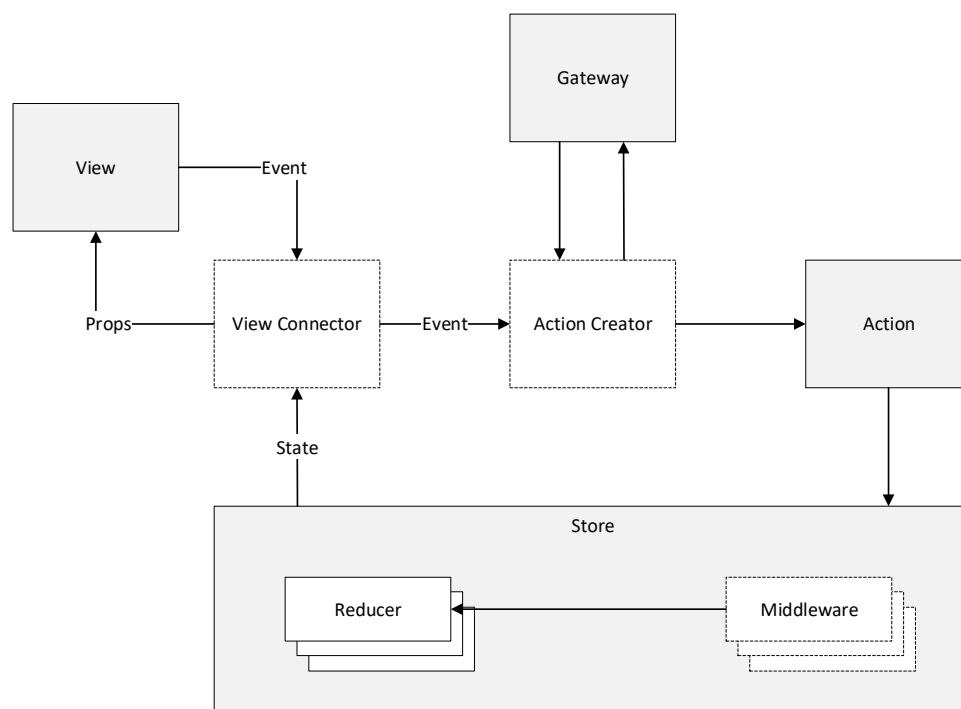


Abbildung 6.6 Redux Architektur - In Anlehnung an Garreau, et al.: Redux in Action (2018 S. 40)

View Konnektoren erhalten den Zustand (*State*) der Anwendung und erstellen daraus Eigenschaften (*Props*) für eine (React-)View Komponente. Sobald der Benutzer eine Interaktion (*Event*) ausführt, wird über den *View Connector* und den *Action Creator* eine *Action* an den *Store* geleitet. *Action Creator* können ggf. eine Nebenläufigkeit besitzen und sogar mehrere *Actions* nacheinander an den *Store* weiterleiten. Es kann sog. *Middlewares*

geben, die auf das *Action* Objekt und den *Store* zugreifen können, bevor die *Action* an die *Reducer* weitergeleitet wird. *Reducer* verändern den Zustand der Anwendung, wodurch der Kreislauf erneut beginnt.

Die Nebenläufigkeit der Action Creator erfolgt über Gateways Interfaces. Diese Interfaces bieten die Möglichkeit, den Prototyp im Laufe seiner Entstehung zuerst mit einem Demo-Gateway auszurüsten, der nur Demo-Daten liefert. Bei fortgeschrittener Entwicklung können andere Implementierungen genutzt werden, die zum Beispiel die Verbindung zu einem HTTP-Server aufbauen. Dadurch erhält der Stakeholder ein noch schnelleres Feedback bei der Entwicklung neuer Features, ohne dass dafür die komplette Web Plattform Architektur (Abschnitt 5.3.1) angepasst werden muss.

6.2 Wahl der Datenbank und heuristische Regeln

6.2.1 Heuristische Regeln

Ein weiteres Ziel dieser Arbeit ist es, die Konsistenz von Einträgen durch heuristische Regeln überprüfen zu können. Dies widerspricht den Anforderungen R28, R29 und R30 in Verbindung mit NR03. Es wird ein sehr variables Datenbankschema gefordert, das sich schwer mit fest definierten Plausibilitätsprüfungen untersuchen lässt.

Ein festgelegtes Datenbankschema erlaubt es, nur vorher definierte Attributtypen zu speichern. Die Anwendung weiß in diesem Fall, welche Werte sie erwartet und benötigt und könnte Benutzereingaben über festgelegte Heuristiken validieren. Eine Änderung des Schemas bedeutet oftmals eine gleichzeitige Änderung der Softwareanwendung, mindestens aber eine Änderung der Heuristiken. Ein variables Datenbankschema legt, wenn nötig, nur grundlegende Attribute eines Objektes fest. Anwender erhalten so die Möglichkeit, Attribute von einzelnen Objekten zu verändern, ohne Schemaanpassungen durchführen zu müssen. Heuristische Regeln lassen sich aber nicht mehr so einfach festlegen, denn sie müssen sich ebenfalls anpassen können. Mitarbeiter müssten die heuristischen Regeln selber definieren und ändern können, sobald sie Änderungen am Schema der Datenbank vornehmen. Da die Mitarbeiter bei den Anpassungen von Attributen oftmals wenig Zeit haben werden, kann es hier schnell zu

Vernachlässigungen kommen. Ebenfalls könnten falsche Heuristiken Fehlalarme auslösen, welche die Nutzbarkeit einschränken würde. Die Mitarbeiter würden ebenfalls nur einfache Heuristiken implementieren können, die eventuell aufwendig festzulegen sind und einen sehr geringen Mehrwert bieten. In den Interviews wurde festgestellt, dass ein variables Schema und eine schnelle Anpassungsfähigkeit den Mitarbeitern wichtiger ist als einfache Heuristiken, da es häufiger zu Anpassungen kommt (siehe Abschnitt 3.5.1).

6.2.2 Wahl der Datenbank

Aufgrund der im vorherigen Abschnitt genannten Anforderungen wurde eine nichtrelationale Datenbank gewählt. Konkret handelt es sich um MongoDB. MongoDB ist ein Document Store (siehe Abschnitt 2.4.1 und 5.3.3), also eine dokumentenorientierte NoSQL-Datenbank. Sie verwaltet die Dokumente in dem Dateiformat BSON (Binary JavaScript Object Notation). Abfragen werden durch objektspezifische Methoden in JavaScript realisiert. Bosch benutzt MongoDB ebenfalls für mehrere IoT²⁵-Projekte²⁶.

6.2.3 Tools und Kommunikation

Die Serverapplikation ist ebenfalls in TypeScript geschrieben. Es wird ExpressJS als serverseitiges Webframework verwendet, das den Webserver startet sowie die Anfragen und Antworten verwaltet. Für die Kommunikation mit der lokalen MongoDB Datenbank wird mongoose verwendet. Mongoose ist eine einfache Bibliothek zur Modellierung und dem Designen von Daten für MongoDB. ExpressJS sowie mongoose bieten typisierte Interfaces, so dass sie ohne Typeinschränkungen in TypeScript verwendet werden können. ExpressJS bietet die Möglichkeit, sog. Middlewares zu benutzen. Diese prüfen vor einer Weiterleitung der Anfrage an die Business Logik (siehe Abbildung 5.6) den Inhalt. So können falsche oder fehlerhafte Anfragen sowie unautorisierte Zugriffe entsprechend beantwortet werden.

²⁵ Internet of Things: Vernetzung und Zusammenarbeit vieler physischer und virtueller Gegenstände

²⁶ Quelle: <https://www.mongodb.com/customers/bosch> - Abgerufen am 04.06.2019 18:55 Uhr

Die Middlewares bieten einen hohen Grad an Wiederverwendbarkeit.

Die Kommunikation zwischen dem Webserver und der Webanwendung erfolgt über eine REST²⁷ API. Je nach Konfiguration des Servers, auf dem die Anwendung später läuft, sollte es keine Einschränkung bei Anforderung NR05 geben. Es kann je nach Verbindung zu höheren Latenzen (bis zu 250ms) zwischen Ländern Europas und des asiatischen Kontinents kommen²⁸. Die Latenz sollte sich bei der geringen benötigten Datenmenge nicht bemerkbar machen.

6.2.4 Sicherheit

Die Passwörter in der Datenbank werden mit bcrypt gehasht (SR01). Er basiert auf dem Blowfish-Algorithmus (Schneier, 1994) und ist speziell für die Speicherung von Passwörtern entwickelt (Provos, et al., 1999).

Der Prototyp setzt die Anforderung SR02 nicht um, denn er läuft lokal auf einem Gerät. Bei der Implementierung ist darauf zu achten, dass eine verschlüsselte Verbindung (HTTPS/TLS) für die Kommunikation zwischen Client und Server verwendet wird.

Die Serverapplikation erstellt einen signierten Cookie, der maximal eine Stunde lang gültig ist (SR03). Der Cookie muss bei einer Anfrage mitgesendet werden und wird von dem Server überprüft. Der Cookie ermöglicht es, Zugriffe zeitlich einzuschränken. Er bietet aber keine weiteren Sicherheiten gegen Man-in-the-Middle²⁹ Angriffe.

6.2.5 Datentransformation

Die Daten der Variantenmatrix werden über einen Algorithmus aus der Excel-Tabelle gelesen und in MongoDB-Datenbankobjekte übersetzt. Diese werden anschließend in die Datenbank geschrieben.

Beim Auslesen der Variantenmatrix wird der größte Nachteil von Excel deutlich. Der Inhalt der Variantenmatrix ist oftmals unformatiert. Er kann

²⁷ Einheitliche Programmierschnittstelle für die Kommunikation zwischen Client und Server

²⁸ Quelle: <https://enterprise.verizon.com/terms/latency/> - Abgerufen am 04.06.2019 22:14 Uhr

²⁹ Angriff, bei dem der Angreifer physisch oder logisch zwischen den Kommunikationspartnern sitzt und Daten lesen und/oder verändern kann.

Webseiten-Links sowie viele Sonderzeichen und Umbrüche enthalten. Diese müssen gesondert gefiltert und ersetzt werden. Ebenfalls muss sichergestellt werden, dass der Algorithmus die blauen Linien (siehe Abbildung 3.3), sowie weitere Reihen der Tabelle ignoriert. Da in der Variantenmatrix viele Informationen redundant gehalten werden, müssen diese für die Speicherung in der Datenbank bereinigt und zusammengefasst werden. So gibt es für jede Automobilbaureihe eine eigene Spalte, in der ein Kreuz die Zugehörigkeit festlegt. Die Datenbank hingegen sollte ein Attribut „Carline“ besitzen, welches die Automobilbaureihen aufschlüsselt.

Der verwendete Algorithmus funktioniert somit nur mit einer festen Version einer Variantenmatrix und muss ggf. bei einer anderen Version neu konfiguriert werden. Er ist aber ausreichend, um die Daten für eine Evaluation zu nutzen.

Es muss darauf geachtet werden, dass der Algorithmus sehr gut dokumentiert wird, denn die Export-Funktion der Webanwendung sollte die Daten wieder in das ursprüngliche Format zurückwandeln. Dies wurde durch den Prototyp nicht vollständig umgesetzt. Daten können zwar exportiert werden, der Aufbau der Tabelle ist jedoch nicht vollständig identisch gegenüber der Excel-Variantenmatrix.

7 Evaluation

In diesem Teil der Arbeit geht es um die Evaluation des entwickelten Prototyps. Dieser wurde mehreren Mitarbeitern präsentiert, die Aufgaben durchführten und einen anschließenden Fragebogen ausfüllen sollten. Während der Evaluation wurden Notizen über die von den Mitarbeitern ausgeführten Schritte und deren Äußerungen gemacht. Ziel der Evaluation ist es unter anderem, Usability-Probleme (siehe Abschnitt 2.2) zu finden und mögliche Erweiterungen der Anwendung aufzudecken. Des Weiteren soll die Akzeptanz bei den Mitarbeitern überprüft werden.

Der erste Abschnitt 7.1 wird die Planung und das Studiendesign näher beschreiben. Im darauffolgenden Abschnitt 7.2 werden die Ergebnisse der Evaluation erläutert und gezeigt. Es wurden weitere nötige und mögliche Erweiterungen erkannt, von denen einige in Abschnitt 7.3 genannt werden.

7.1 Studiendesign

7.1.1 Ziele der Studie

Wie eingangs erwähnt, sind die Ziele der Studie das Finden von Usability-Problemen des Prototyps, das Erkennen weiterer Anforderungen und das Prüfen der Akzeptanz von einer nicht Excel-basierten Anwendung. Die Studie findet während der Arbeitszeit der Mitarbeiter statt. Der Fokus wurde auf eine quantitative Untersuchung der Anwendung gelegt, mit dem Ziel, so viele neue Erkenntnisse wie möglich zu erhalten.

7.1.2 Evaluationsumgebung

Die Studie wurde an den Arbeitsplätzen der Mitarbeiter durchgeführt, da Besprechungsräume nicht immer verfügbar waren. Die Studie wurde an einem Laptop durchgeführt, der mit einer externen Maus bedient werden konnte. Der Laptop verfügt über eine Auflösung von 2.560 x 1.440 Pixel, so dass nicht es zu Einschränkungen bei der Bedienung von großen Excel-Tabellen kommt. Die Helligkeit des Displays ist manuell auf die höchste Stufe festgelegt worden. Die Usability-Tests wurden moderiert, so dass auch Anschlussfragen gestellt werden konnten oder erkenntnisreiche

Diskussionen nach dem Usability-Test geführt werden konnten. Die Tests wurden persönlich durchgeführt. Das heißt, der Moderator saß neben dem Probanden und konnte somit die Testperson leichter beobachten. Es wurde die Think-aloud-Methode eingesetzt, bei der die Probanden dazu aufgefordert wurden, ihre Gedanken, Empfindungen und Bewertungen während der Durchführung der Aufgaben auszusprechen.

Bei der Studie handelt es sich um ein Within-Group Design, bei dem jeder Proband alle Aufgaben bearbeiten soll.

7.1.3 Zu bearbeitende Aufgaben

Den Probanden wurden sechs realitätsnahe Aufgaben gestellt (Siehe Anhang Abbildung 2). Diese wurden zuvor zusammen mit einem nicht zur Probandengruppe gehörenden Mitarbeiter erstellt. Die Aufgaben sollen von den Probanden innerhalb von ungefähr 20 Minuten gelöst werden können, so dass eine Evaluation nicht länger als 30 Minuten in Anspruch nimmt. Es gibt drei Aufgaben, die mit der bisher genutzten Excel-Tabelle gelöst werden sollen sowie drei Aufgaben, die mit dem Prototyp (in den Aufgaben mit *WEB* markiert) gelöst werden sollen. Dabei sind je eine Excel- sowie Prototyp-Aufgabe in der Aufgabenstellung identisch. Die nachfolgende Abbildung 7.1 veranschaulicht den Ablauf der Aufgaben.

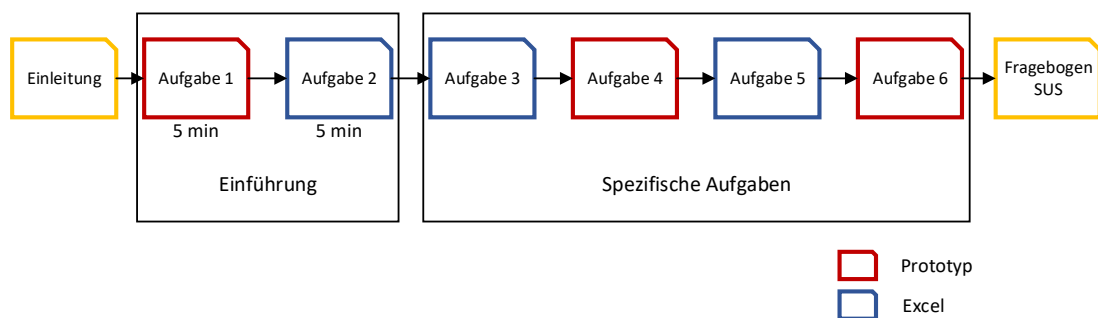


Abbildung 7.1 Ablauf zu bearbeitender Aufgaben

Die in der Abbildung rot markierten Aufgaben sind mit dem Prototyp, blau markierte mit der Excel-Tabelle zu lösende Aufgaben. Aufgabe 1 und 2 sind für die Einführung in den Prototyp sowie die Excel-Tabelle gedacht. Dabei machen sich die Probanden innerhalb von je fünf Minuten mit der Anwendung vertraut und erhalten einen auffrischenden Einblick in die Excel-

Tabelle. Der Moderator führt die Benutzer durch die Funktionalitäten und die Probanden erhalten, falls nötig, Hilfestellung. Ziel der Aufgaben 1 und 2 ist es, dass der Proband die Suche und den Filter nutzt sowie mindestens zwei Varianten miteinander vergleicht.

Die Aufgaben 3 bis 6 sind spezifische Aufgaben, während diesen wurden die Bearbeitungszeiten mit Hilfe einer Stoppuhr gemessen. In den Aufgaben 3 und 4 muss der Proband eine identische Aufgabe lösen, einmal in Excel, anschließend mit dem Prototyp. Der Proband soll hier vor allem die Suche und den Filter benutzen, um eine bestimmte Teilenummer zu finden. Damit der Proband keinen Vorteil in der vierten Aufgabe hat, muss er dort eine andere Teilenummer als in Aufgabe 3 suchen. In Aufgabe 5 und 6 sollen Varianten gesucht und anschließend verglichen werden. Hier wird vor allem die Vergleichsfunktion von Varianten untersucht. Ein Lerneffekt bei der Durchführung der Aufgaben kann somit vorhanden sein. Dieser ist beabsichtigt, denn Ziel der Aufgaben 5 und 6 ist es, eine erweiterte Suche nach zwei Teilenummern zu verwenden, um die Varianten anschließend zu vergleichen. Die Benutzer sollen daher eine Suche nach einer Teilenummer wie in Aufgabe 3 und 4 mindestens einmal durchgeführt haben.

Die Aufgaben 5 und 6 enthalten eine weitere Teilaufgabe „Finden Sie alle Geräte der Carline XYZ“. Nachdem in einem ersten Testdurchlauf festgestellt wurde, dass diese zusätzliche Aufgabe zu viel Zeit bei einigen Probanden in Anspruch nahm, wurde sie nicht weiter durchgeführt. Die Studie sollte bei keinem der Teilnehmer länger als 30 Minuten dauern.

7.1.4 Metriken

Bei der Durchführung der Aufgabe 3, 4, 5 und 6 werden die Metriken Erfolgsrate und der Zeitaufwand erhoben, die im Folgenden definiert werden (Jacobsen, et al., 2017 S. 185):

Erfolgsrate: Anzahl der Probanden, die eine Aufgabe erfolgreich abgeschlossen haben.

Zeitaufwand: Zeit, die ein Proband benötigt, um eine Aufgabe erfolgreich durchzuführen.

Der Zeitaufwand wird hier vor allem als Richtwert genutzt, um eine Differenz bei identischen Aufgabentypen in Excel oder dem Prototyp festzustellen.

7.1.5 System Usability Scale Fragebogen

Nach der Durchführung der Aufgaben musste jeder Proband einen System Usability Scale (SUS) Fragebogen (siehe Abbildung 7.1 und Anhang Abbildung 3) ausfüllen. Der System Usability Scale ist von John Brooke (1996) entwickelt worden und ein etablierter Fragebogen (Lewis, 2018), um die Usability einer Anwendung zu ermitteln. Dabei handelt es sich um einen standardisierten Fragebogen mit zehn Aussagen, welchen der Proband auf einer Skala von 1 bis 5 gar nicht zustimmen kann oder voll zustimmen kann. Der SUS ist eine Likert-Skala. Um das Ergebnis eines SUS Fragebogens zu berechnen, werden die Ergebnisse jeder Antwort miteinander addiert. Jede Antwort kann zwischen 0 und 4 Punkten erreichen. Die Antworten zu den Aussagen 1, 3, 5, 7 und 9 werden durch Skala Position minus 1 berechnet. Die Antworten zu den Aussagen 2, 4, 6, 8, und 10 werden durch 5 minus Skala Position berechnet. Die Ergebnisse der einzelnen Antworten werden dann addiert und mit dem Faktor 2,5 multipliziert. Der SUS ist eine Perzentil Interpretation mit einem Wert zwischen 0 und 100.

7.1.6 Hypothesen

Ausgehend von den Zielen der Studie werden Hypothesen aufgestellt, die eine Grundlage für die statistische Analyse sind. Ziel ist es, auf Basis der Stichproben eine Aussage über die Gesamtheit treffen zu können. Dazu werden die gesammelten Daten untersucht und es werden Nullhypothesen (H_0) bei statistisch signifikanten Ergebnissen abgelehnt. Bei Ablehnung der Nullhypothese (H_0) wird die Alternativhypothese (H_1) angenommen (Wohlin, et al., 2012).

Es gibt beim Testen von Hypothesen zwei unterschiedliche Arten von möglichen Fehlern. Der Fehler 1. Art ist das Ablehnen der Nullhypothese und Annehmen der Alternativhypothese, obwohl diese richtig ist. Es wird eine Irrtumswahrscheinlichkeit, auch Signifikanzniveau α genannt, mit dem Wert von 0,05 gewählt. Der Fehler 2. Art ist das Annehmen der Nullhypothese, obwohl die Alternativhypothese richtig ist. Die Wahrscheinlichkeit, dass dieser Fehler eintritt wird β genannt. Je kleiner α , desto größer β . Die Wahrscheinlichkeit β hängt ebenfalls von der Art des Tests ab.

Es werden nachfolgend die Hypothesen genannt. Ebenfalls die verwendeten Metriken (fett markiert).

H1₀: Es gibt keinen Zeitunterschied beim Finden von Varianteninformationen.

H1₁: Es gibt einen Zeitunterschied beim Finden von Varianteninformationen.

Hierfür wird die Metrik **Zeitaufwand** verwendet.

H2₀: Es gibt keinen Zeitunterschied beim Feststellen von Unterschieden zwischen Varianten.

H2₁: Es gibt einen Zeitunterschied beim Feststellen von Unterschieden zwischen Varianten.

Hier wird ebenfalls die Metrik **Zeitaufwand** verwendet.

H3₀: Es gibt keinen Unterschied in der Usability der Anwendung gegenüber der Usability von Excel.

H3₁: Es gibt einen Unterschied in der Usability der Anwendung gegenüber der Usability von Excel.

Die Usability wird mit Hilfe des **SUS** gemessen.

7.1.7 Studiensubjekte

Insgesamt wurden acht Mitarbeiter bei dem Usability-Test befragt und beobachtet. Nach Sarodnick und Brau (2011 S. 174) werden bei fünf Probanden mindestens 55 %, durchschnittlich 85,55 % der Usability-Probleme bei einer Standardabweichung von 9,30 gefunden. Bei zehn Probanden werden mindestens 82 % und durchschnittlich 94,69 % der Usability-Fehler bei einer Standardabweichung von 3,22 gefunden. Es wurden nach dem sechsten Probanden keine weiteren Usability-Fehler sowie neue mögliche Ergänzungen und Erweiterungen erkannt. In Anbetracht des zeitlichen Aufwands für einen Mitarbeiter sowie der Wahrscheinlichkeit, neue unbekannte Usability-Fehler als auch mögliche Ergänzungen der Anwendung zu finden, wurde die Studie nach dem achten Probanden abgebrochen. Es wurde darauf geachtet, dass sowohl Probanden der Gruppe 1 (siehe Abschnitt 3.9

sowie nachfolgende Tabelle 3) als auch Probanden der anderen Gruppen gewählt wurden.

Proband	Nutzung	Gruppe
P5	alle zwei Tage	Gruppe 2
P8	mehrfach täglich	Gruppe 1
P2	mehrfach täglich	Gruppe 1
P6	alle drei Tage	Gruppe 2
P3	mehrfach täglich	Gruppe 2
P7	alle drei Tage	Gruppe 3
P4	monatlich	Gruppe 3
P1	monatlich	Gruppe 3

Tabelle 3 Nutzung der Excel Variantenmatrix der Probanden

7.2 Auswertung/Ergebnisse der Evaluation

7.2.1 Beobachtungen und Äußerungen während der Studie

Es werden in diesem Abschnitt Beobachtungen beschrieben, wie die Probanden an die Aufgaben herangingen sind und die Weboberfläche des Prototyps empfanden.

Alle Probanden haben die Webanwendung mit sehr großem Interesse angenommen, erkennbar durch deren Äußerungen, nachdem sie den Prototyp einige Minuten ausprobierten. Da einige Probanden großes technisches Verständnis im Bereich Softwareentwicklung mitbrachten, wurden viele Fragen über die Architektur sowie der verwendeten Bibliotheken und Frameworks gestellt. Zwei Probanden war es wichtig gewesen, dass der Code der Anwendung von den Mitarbeitern wartbar ist.

Von drei Probanden kam die Frage, warum das Seitenmenü weitere Möglichkeiten listet, obwohl diese noch nicht implementiert wurden. Die Seitenmenü-Elemente sollten hier nur weitere mögliche Anwendungsfeatures darstellen. Einem Probanden fiel eine mögliche Inkonsistenz auf. Der Prototyp besitzt in dem Filter „Customer“ ein Attribut „HD“. Dieser Fehler ist nach nachfolgender Recherche darauf zurückzuführen, dass die Excel-Variantenmatrix zwei Spalten mit den Namen „Customer“ besitzt. Dies ist weder dem Probanden bewusst gewesen, noch bei der Datenüberführung der riesigen Excel-Tabelle aufgefallen. Sieben Probanden wünschten sich eine andere Sortierung der Filter. Diese sollten in ihrer Positionierung angepasst werden können. Drei Probanden benutzten die Tastenkombination

„Strg + F“, um sowohl die Excel-Variantenmatrix als auch den Web Prototyp zu durchsuchen. Alle Probanden empfanden die Möglichkeit, Varianten zu vergleichen und dabei Unterschiede markiert zu bekommen als äußerst hilfreich. Ein Proband wünschte sich in dieser Ansicht nur eine Darstellung der Unterschiede.

Es gab zwei Probanden, die gerne eine Übersicht, ähnlich wie in Excel hätten. Ihnen böte die Webapplikation gefühlt zu wenig Informationen auf dem ersten Blick, ein Proband wünschte sogar eine Ansicht wie in der Excel-Tabelle. Ein Proband der Gruppe 1, er benutzt die Variantenmatrix also täglich, boten die Suchergebnisse zu wenig Informationen. Er benötigt immer das sichtbare Attribut „Sample Phase“ in der Ergebnisliste. Wiederum gab es drei Probanden, denen die Übersicht in Excel zu verwirrend war. Ihnen reichen die zuerst sichtbaren Informationen der Suchergebnisse des Prototyps.

Es gab zwei Probanden, die sich eine weitere Variantenansicht wünschten. In dieser Ansicht sollten Variantenfamilien und deren „Vererbung“ sichtbar sein. Fünf Probanden wünschen sich eine sortierbare Suchergebnisliste. Dies wurde nicht von allen Probanden direkt geäußert, doch es wurde daran erkannt, dass sie mit einem Klick auf den Tabellenkopf der Suchergebnisse versuchten, diese zu sortieren.

Die Probanden mussten während der Studie mehrmals daran erinnert werden, dass es sich um eine Think-aloud-Studie handelt und Gedanken und Handlungen ausgesprochen werden sollen.

Vier Probanden versuchten, die Suche als Volltextsuche zu benutzen und gaben mehrere Suchbegriffe, durch ein Leerzeichen getrennt, ein. Diese Möglichkeit bietet der Prototyp nicht, weshalb keine Suchergebnisse gefunden wurden.

Zur allgemeinen Vorgehensweise beim Suchen von Varianten sowohl im Prototyp als auch der Excel-Tabelle lässt sich sagen, dass alle Probanden unterschiedlich agierten. So gaben fünf Mitarbeiter die vollständige Bosch Teilenummer ein, drei Mitarbeiter lediglich die letzten drei oder vier Stellen, um Ergebnisse anzeigen zu lassen. Proband 6 versuchte zunächst das Gerät über ausschließlich den Filter zu finden, anstatt die Suche zu benutzen. Dies spiegelt sich in den Ergebnissen der Bearbeitungszeit der Aufgaben

wider. Ebenfalls gibt es drei Möglichkeiten, die Carline einer Variante in dem Prototyp zu finden. Sie kann durch Aktivierung der Suchergebnisspalte direkt in den Suchergebnisse angezeigt werden. Sie ist in der Detailansicht eines Suchergebnisses sowie in der von den Suchergebnissen generierten Excel-Datei sichtbar. Jede dieser Möglichkeiten wurde von den Mitarbeitern während der Studie durchgeführt und führte zu unterschiedlichen Ergebnissen in der Bearbeitungszeit (siehe Abschnitt 7.2.2).

Alle Probanden waren der Meinung, dass die Software mit weiteren nötigen Features die Arbeit erleichtern kann. Es gab drei Probanden, die nach der Studie fragten, wann sie die Anwendung nutzen könnten. Sobald die Probanden Kritik übten, fehlten weitere Funktionen. Es gab keinen Probanden, dem der Prototyp überhaupt nicht gefiel. Es ist daher anzunehmen, dass eine Akzeptanz für eine Anwendung dieser Art bei den Mitarbeitern vorhanden ist.

7.2.2 Auswertung der Bearbeitungszeiten

Alle Probanden schlossen die Aufgaben 3 und 4 erfolgreich ab. Proband 6 führte die Aufgaben 5 und 6 nicht durch, alle anderen Probanden führten die Aufgaben 5 und 6 ohne die Zusatzfrage durch.

Die Aufgaben 1 und 2 sind Einstiegsaufgaben, bei denen keine Bearbeitungszeiten gemessen wurden. In diesem Abschnitt werden die Bearbeitungszeiten der Aufgaben 3, 4, 5 und 6 verglichen und Unterschiede dargestellt. In der nachfolgenden Abbildung 7.2 wird die Bearbeitungszeit der Aufgaben 3 und 4 verglichen.

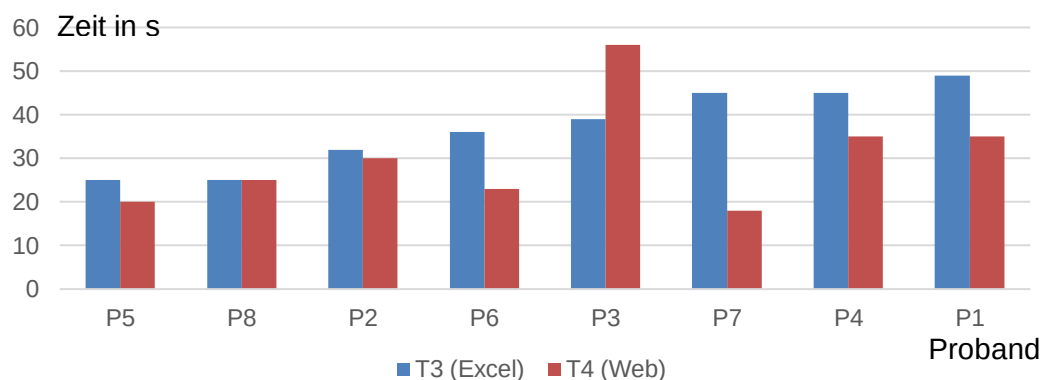


Abbildung 7.2 Diagramm Bearbeitungszeit der Aufgaben 3 und 4 im Vergleich

In Aufgabe 3 sollten die Probanden eine Variante mit Hilfe einer Teilenummer in der Excel-Tabelle finden und zeigen, zu welcher Automobilbaureihe diese Variante gehört. In Aufgabe 4 sollten die Probanden dies mit Hilfe des Prototyps, aber einer anderen Teilenummer durchführen. Die benötigte Bearbeitungszeit in der Excel-Tabelle hängt sehr wahrscheinlich von der Benutzungsfrequenz der Variantenmatrix sowie der Kenntnis von Excel und deren Funktionalitäten zusammen. Proband 1 und Proband 4 benutzen die Variantenmatrix sehr selten (siehe Tabelle 3 auf Seite 80).

Die unterschiedlichen Bearbeitungszeiten der Aufgabe 4 hängen sehr wahrscheinlich mit der kurzen Einarbeitungsphase von fünf Minuten sowie der unterschiedlichen Herangehensweise und Lösungsfindung zusammen. Es gab fünf Probanden, die nach dem Test der Meinung gewesen waren, dass eine fünf-minütige Einführung eventuell nicht ausreichend gewesen sei und sie deshalb langsamer waren eigentlich möglich. Proband 3 brauchte bei der Bearbeitung der Aufgabe 4 sehr viel länger als alle anderen Probanden. Der Proband entschied sich, anstatt die Variante mit Hilfe der Teilenummer über die Suche zu finden, zuerst einige Filter einzustellen, die nicht zielführend waren. Dies tat er, da er die Variante vermeintlich zu kennen glaubte.

Es folgen Tests zur Berechnung der statistischen Signifikanz durch die einfaktorielle Varianzanalyse ANOVA. Es wurde ein Signifikanzniveau von 5% gewählt. Um eine statistische Signifikanz zu erreichen, müssen der F-Wert > 1 und der p-Wert < 0.05 sein.

In nachfolgender Tabelle 4 werden der Mittelwert und die Standardabweichung der gemessenen Zeiten von Aufgabe 3 und 4 gezeigt.

Aufgabe	Mittelwert in s	Standardabweichung
Aufgabe 3 (Excel)	37.00	9.18
Aufgabe 4 (Prototyp)	30.25	12.21

Tabelle 4 Mittelwert, Standardabweichung Aufgabe 3 und 4

Aufgrund der hohen Standardabweichung der Aufgabe 4 ergibt sich ein p-Wert von 0.2319, bei einem F-Wert von 1.5620, weshalb kein statistisch signifikanter Unterschied zwischen den Bearbeitungszeiten der Aufgabe 3 und 4 existiert.

Es ist dennoch zu erkennen, dass die Bearbeitung der Aufgabe mit dem Prototyp häufiger weniger Zeit in Anspruch nahm, als die Bearbeitung der gleichen Aufgabe in Excel. Die Probanden hatten durch Aussagen bestätigt, das Gefühl zu haben, mit dem Prototyp schneller zu sein, obwohl sie nach eigener Aussage eine sehr kurze Einarbeitungsphase hatten. Somit wird die Nullhypothese H_{10} mit dem Hinweis angenommen, dass der Stichprobenumfang wahrscheinlich nicht groß genug gewählt wurde.

Die nachfolgende Abbildung 7.3 zeigt die Bearbeitungszeiten der Probanden für die Aufgaben 5 und 6. Da aufgrund einer Unterbrechung der Proband 6 die Aufgabe 5 nicht durchführen konnte, wird er im Vergleich der Aufgaben 5 und 6 nicht aufgeführt. Eine erneute Durchführung mit dem Probanden hätte das Gesamtergebnis verfälschen können. Da die vorangegangenen Aufgaben 3 und 4 von Proband 6 vollständig gelöst wurden, hat dies keine Auswirkungen auf das Ergebnis sowie die nachfolgenden Aufgaben.

Wie in Aufgabe 3 und 4 sollten die Probanden in Aufgabe 5 und 6 jeweils eine identische Aufgabe in Excel sowie mit Hilfe des Prototyps durchführen. Jeder Proband sollte zwei Varianten mit Hilfe unterschiedlicher Teilenummern suchen, Unterschiede zwischen den Varianten feststellen und diese dem Moderator zeigen. Dabei waren die Teilenummern in beiden Aufgaben (Excel und Prototyp), anders als in den vorherigen Aufgaben 3 und 4, identisch.

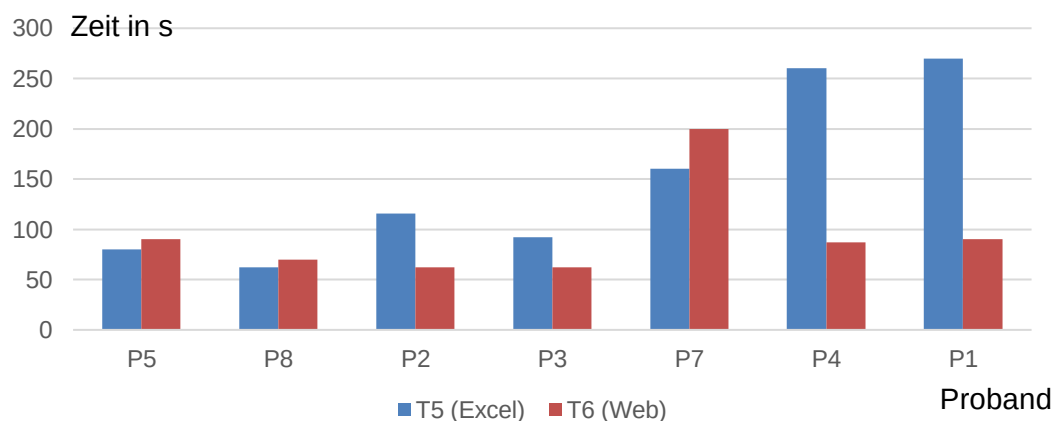


Abbildung 7.3 Diagramm Bearbeitungszeit der Aufgaben 5 und 6 im Vergleich (ohne P6)

Wie in der vorherigen Aufgabe sind die Bearbeitungszeiten der Aufgabe in Excel sehr wahrscheinlich von der Benutzungsfrequenz der

Variantenmatrix sowie der Kenntnis von Excel und deren Funktionalitäten abhängig. Um mehrere Varianten in Excel einem Filter hinzuzufügen, muss ein Haken „*Dem Filter die aktuelle Auswahl hinzufügen*“ beim Auswählen eines zweiten Elementes gesetzt werden. Dieser wurde von vier Probanden fälschlicherweise vergessen, so dass sie eine erneute Filtersuche starten mussten. Sobald die Probanden beide Varianten untereinander gelistet hatten, gingen sie ebenfalls unterschiedlich beim Vergleichen innerhalb der Excel-Tabelle vor. So benutzten zwei Probanden die Cursor-Tasten, um zwischen den Feldern hin und her zu springen und einen visuellen Anhaltspunkt zu haben, die anderen Probanden benutzten die Scroll-Funktion von Excel und verglichen die Attribute der Varianten ohne visuellen Anhaltspunkt. Alle Probanden gaben an, dass das Vergleichen mit dem bloßen Auge fehleranfällig und zeitaufwendig sei.

Die Bearbeitungszeit der Aufgabe 6 variiert durch unterschiedliche Vorgänge beim Suchen von Varianten. So suchten vier Probanden direkt hintereinander nach den unterschiedlichen Varianten, ohne die vorherige Suche zu löschen. Drei Probanden löschten die vorherige Suche. Ebenfalls gaben fünf Probanden die Teilenummer vollständig ein, zwei gaben lediglich die letzten drei bis vier Ziffern ein und nutzten die Suchvorschläge, wodurch sie insgesamt schneller waren. Proband 7 benötigte für das Suchen der Varianten besonders viel Zeit. Nachdem er die erste Teilenummer eingegeben und die Variante dem Vergleich hinzugefügt hatte, wusste der Proband für eine kurze Zeit nicht, wie er eine weitere Variante suchen kann. Durch selbstständiges Probieren erkannte er, dass er nach einer weiteren Variante suchen kann, ohne dass der Vergleich oder die vorherige Suche gelöscht wird. Der Proband gab anschließend an, dass er wohl zu kompliziert dachte. Dennoch sollte dies als möglicher Usability-Fehler bewertet werden, da die Anwendung auch ohne größere Erklärung intuitiv benutzbar sein sollte.

Es folgen weitere Tests zur Berechnung der statistischen Signifikanz. In nachfolgender Tabelle 5 werden Mittelwert und Standardabweichung der Aufgaben 5 und 6 dargestellt.

Aufgabe	Mittelwert in s	Standardabweichung
Aufgabe 5 (Excel)	148.57	85.39
Aufgabe 6 (Prototyp)	94.43	48.20

Tabelle 5 Mittelwert, Standardabweichung Aufgabe 5 und 6

Es ergibt sich bei der Berechnung der Signifikanz ein F-Wert von 2.1342 und ein p-Wert von 0.1697, weshalb die ebenfalls kein statistisch signifikanter Unterschied in den Bearbeitungszeiten zwischen den Aufgaben 5 und 6 existierte. Die Hypothese H_{20} wird daher angenommen.

Auch wenn sich die folgende Aussage nicht durch statistische Signifikanz belegen lässt, gaben alle Probanden an, dass sie die Aufgaben mit Hilfe des Prototyps gefühlt und meistens auch messbar in gleicher Geschwindigkeit, teilweise sogar schneller durchführen konnten. Für ein möglicherweise statistisch signifikanteres Ergebnis müssten mehr Mitarbeiter befragt und in die Studie einbezogen werden sowie möglicherweise hindernde Usability Fehler behoben werden. Der Unterschied in den Bearbeitungszeiten der Aufgaben 5 und 6 ist bei viel Variantenmatrix nutzenden Probanden sehr gering.

Hingegen waren Mitarbeiter, welche die Variantenmatrix selten nutzen, kaum langsamer, als die häufig nutzenden. Sechs von sieben Mitarbeitern benötigten zwischen 60 und 90 Sekunden für die Lösung der Aufgaben, unabhängig von der Nutzungshäufigkeit der Excel-Variantenmatrix.

Die Mitarbeiter der Abteilung Variantenverwaltung (Abschnitt 3.5) vergleichen sehr häufig Varianten für andere Mitarbeiter oder Abteilungen. Diese Aufgaben könnten durch das neue Feature von den Abteilungen oder Mitarbeitern selbständig durchgeführt werden, wodurch Kommunikation minimiert und Arbeitszeit eingespart würde.

7.2.3 Auswertung des System Usability Scores

Der System Usability Score (SUS) Fragebogen (siehe Anhang Abbildung 3) wurde von allen Probanden ausgefüllt, doch der Proband 4 wollte keine Antwort zu der Aussage 6 des Fragebogens wählen, da er die Inkonsistenzen des Systems nicht bewerten wollte. Der Proband 4 wird daher nicht in die Berechnung des Usability Scores einbezogen.

Der System Usability Score ohne Proband 4 beträgt 80.0 (siehe Anhang Tabelle 2).

Bangor, et al. (2008) verglichen die SUS Ergebnisse von 2.324 Studien und stellten einen Mittelwert von 70,14 sowie einen Median von 75,00 fest. Sie teilten den SUS ebenfalls in Quartile ein und legten eine Akzeptanz- sowie Bewertungsskala fest (Abbildung 7.4). Demnach ist die Usability des Prototyps mindestens mit „Gut“ zu bewerten und liegt über dem Durchschnitt. Der Mittelwert für Webanwendungen liegt nach Bangor, et al. (2008) sogar nur bei 68,05 Punkten bei einer Standardabweichung von 21,56 und 1.180 untersuchten Studien. Die Usability scheint nach Aussagen der Mitarbeiter sowie anhand des SUS Scores besser als die von Excel zu sein. Es ist somit ein Unterschied feststellbar, und es kann die Hypothese H3₀ verworfen werden. Dennoch zu beachten, dass es sich in dieser Studie um einen sehr kleineren Stichprobenumfang handelt.

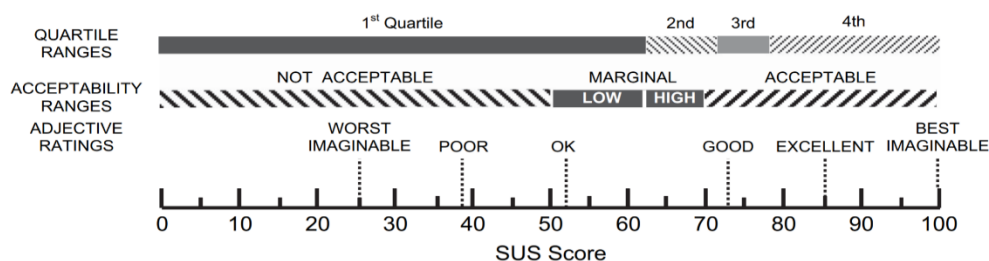


Abbildung 7.4 Vergleich SUS Werte nach Quartil, Akzeptanz und Bewertung
(Bangor, et al., 2008 S. 592)

7.2.4 Validität

Die Probanden erhielten eine jeweils fünfminütige Einführung in den Prototyp sowie der Excel-Tabelle. Die Einführung erfolgte durch Erklärung der Features, welche für die Lösung der Aufgaben nötig sind. Die Probanden mussten dabei selbstständig das Programm benutzen. Dabei kann es zu Abweichungen in den Erklärungen der Features gekommen sein, da Probanden unterschiedliche Rückfragen gestellt haben. Ebenfalls könnten Probanden die Funktionen innerhalb der fünfminütigen Einführung unterschiedlich aufgenommen und verinnerlicht haben.

Die Bearbeitungszeiten wurden manuell mit einer Stoppuhr gemessen. Die durch die menschliche Reaktionsgeschwindigkeit bedingten Abweichungen sind so gering, dass keine Ergebnisänderung zu erwarten ist.

Es wurde darauf geachtet, dass möglichst repräsentative Aufgaben sowie Probanden gewählt wurden. Es wurde sichergestellt, dass Probanden mit hoher, mittlerer als auch geringer Erfahrung und der Erprobung der Anwendung teilnahmen.

Da bei der Studie nur acht Probanden unterschiedlicher Benutzergruppen befragt wurden, können die Ergebnisse einzelner Benutzergruppen nicht verallgemeinert werden. Dazu müsste eine größere Stichprobe gewählt werden. Genauso müssen für einen aussagekräftigeren SUS mehr Probanden gewählt werden. Der SUS ist zwar etabliert und ausreichend für eine Tendenz, doch es handelt sich hierbei nur um ein subjektives Maß. Der SUS sollte während einer nachfolgenden Weiterentwicklung mehrmals überprüft werden (Bangor, et al., 2008 S. 588f.).

7.3 Zusätzliche Erweiterungen an die Anwendung

Bei der Durchführung der Studie wurden Usability-Probleme durch Beobachtung sowie Aussagen von Mitarbeitern festgestellt, welche durch die nachfolgenden Erweiterungen behoben werden sollen.

Ebenfalls wurden von den Mitarbeitern weitere mögliche und hilfreiche Features genannt, die durch neue mögliche Erweiterungen festgehalten werden. Einige von ihnen werden nachfolgend genannt. Aufgrund der Menge und Komplexität an Erweiterungen können nicht alle im Rahmen dieser Arbeit aufgeführt werden. Die neuen Erweiterungen werden mit nummeriert.

E1: In Anforderung R05 wird lediglich das Suchen mehrerer Begriffe festgelegt, aber nicht der Umstand, dass diese auch als Volltext eingegeben werden können. Während der Evaluation versuchten mehrere Probanden die Suche als Volltextsuche zu benutzen. Daher muss die Anforderung genauer spezifiziert werden.

E2: Wie bereits erwähnt, war einem Probanden während der Suche nicht bekannt, dass er weitere Suchbegriffe eingeben kann und somit seine Suche erweitern kann. Daher muss die Anwendung dem Benutzer einen

Hinweis geben, dass er weitere Suchbegriffe in die Suche eingeben kann, sobald er schon nach einem Begriff gesucht hat.

E3: Oft kommt es bei den Mitarbeitern vor, dass immer dieselben Filter verwendet werden. Es ist daher für die Mitarbeiter eine Arbeitserleichterung, wenn Filter gespeichert werden können.

E4: Gespeicherte Suchen oder Filtereingaben aus E3 sollen wiederverwendet werden können. Dazu müssen die gespeicherten Filter- und Suchkombinationen geladen und angewendet werden können.

E5: Während der Usability-Tests wurde festgestellt, dass jeder Mitarbeiter anders beim Suchen und Filtern vorgeht. Ebenfalls wurde erkannt, dass jeder Mitarbeiter andere Prioritäten bei den Filterelementen setzt. Daher muss das System dem Benutzer die Möglichkeit bieten, die vorausgewählten Filter sowohl in der Reihenfolge als auch in der Menge der sichtbaren Filter beim Öffnen der einzelnen Filterebenen anpassen zu können.

E6: Bisher werden den Benutzern nur drei festgelegte Spalten in den Suchergebnissen angezeigt, sofern keine weiteren in dem Filterbereich ausgewählt wurden. Hier wünschen sich einige Probanden eine Anpassungsmöglichkeit in Reihenfolge und Menge der vorgewählten Spalten.

E7: Es gab zwei Probanden, die von der Vergleichsseite ein einzelnes Element aus dem Vergleich entfernen wollten. Daher sollte die Anwendung den Benutzern die Möglichkeit bieten, ein einzelnes Element aus dem Vergleich zu entfernen, ohne dass sich die Reihenfolge der anderen Elemente ändert. Ebenfalls sollte der Benutzer ein Element im Vergleich auswählen können, gegen das verglichen wird.

E8: Ein Proband wünschte sich die ausschließliche Visualisierung der Unterschiede ohne längeres vertikales Scrollen. Daher sollte die Anwendung die Möglichkeit bieten, nur Unterschiede im Vergleich sichtbar zu machen.

8 Verwandte Arbeiten

H. Schumann, et al. (2010) evaluierten in ihrer Arbeit Vor- und Nachteile für die Nutzung von Excel-basierten Anwendungen, die in der Luftfahrtbranche häufig verwendet werden. Sie stellten unter anderem fest, dass Excel die Arbeit durch eine nicht ausreichende Versionierung erschwere. Dies gilt vor allem, wenn die Mitarbeiter von unterschiedlichen Orten zur gleichen Zeit auf dieselbe Datei zugreifen. Fehleingaben oder Änderungen an der Tabelle können zu falschen Formelergebnissen führen. Als Problematisch erkannten sie ebenfalls die Abhängigkeit von Excel zum ausführenden Betriebssystem. Es wurden zwei Prototypen entwickelt, welche die Nachteile minimierten, sowie weitere benötigte Features bereitstellen sollten. Die größte Herausforderung sahen sie in der Nachbildung der Excel Formeln, da diese sehr viele Freiheiten bieten.

Z. Parker, et al. (2013) verglichen in ihrer Arbeit NoSQL MongoDB mit Microsoft SQL Server Express hinsichtlich ihrer Performance und gaben an, für welche Einsatzgebiete sich welche Lösung besser eignen würde. Sie stellten fest, dass MongoDB beim Einfügen, Verändern und Abfragen einfacher Eigenschaften oftmals sehr viel performanter ist. SQL aber besitzt Vorteile hinsichtlich der Geschwindigkeit beim Verändern und Abfragen von non-key Attributen, sowie komplexerer Aggregationen. Sie empfehlen MongoDB für größere Datenmengen, bei denen sich das Schema häufig ändert oder Abfragen nicht sehr komplex sind. Die Autoren merkten an, dass MongoDB nicht stark verbreitet ist. Im Gegensatz dazu wird SQL in der Industrie sehr viel häufiger verwendet.

Bianchi, et al. (2003) befassten sich mit dem iterativen Reengineering-Prozess eines Softwaresystems. Sie stellten dafür ein Modell auf und wandten es an ein aktiv-benutztes System an. Sie versuchten damit zu bestätigen, dass der untersuchte Prozess vorherige Anforderungen erfüllt, sowie die Wirksamkeit des Modells zu messen. Das dafür verwendete System eignete sich aufgrund der Architektur für eine iterative Implementierung von neuem Code, während das alte System weiterhin voll funktionsfähig blieb. Sie

stellten fest, dass durch das Reengineering vor allem die Code-Verunreinigung, -Kopplung als auch Schichtenarchitekturen verbessert oder gar entfernt werden konnten. Sie gaben ebenfalls die für das Reengineering benötigten Arbeitszeiten an.

D. Amalfitano, et al. (2014) untersuchten in ihrer Arbeit einen Spreadsheet-basierten Prozess bei dem Automobilhersteller Fiat, mit dem Ziel, den Prozess ebenfalls durch eine Web Anwendung zu ersetzen. Die Webanwendung basiert ebenfalls auf einer MVC-Architektur. Sie stellten fest, dass sehr viele redundante Daten und Informationen über mehrere hundert Excel Dateien verteilt sind. Dies begründen sie durch die einfache Zugänglichkeit zu Excel, welches in vielen Unternehmen immer mehr als Business-Tool verwendet wird. Die Autoren geben ebenfalls an, dass Excel Spreadsheets schwierig zu warten sind. Fehler werden dort schnell übersehen und Formeln und Skripte sind schlecht dokumentiert, sowie schwierig zu verstehen. Sie rekonstruierten das Datenmodell, den Prozess sowie Code der Excel-basierten VBA-Anwendungen und migrierten die Logik in ein MVC-basierte Web-Anwendung, entwickelt mit Hilfe von .NET Framework sowie Microsoft SQL Server. Anschließend validierten sie die Web-Anwendung und stellten fest, dass die Fehlerrate beim Ausführen von Aufgaben um 100% gesunken ist. Die Aufgaben konnten außerdem bis zu 33% schneller durchgeführt werden. Zudem wurden 41% weniger Schritte benötigt.

9 Fazit und Ausblick

In diesem und dem letzten Kapitel werden die Vorgehensweise, Ergebnisse und Erkenntnisse zusammengefasst. Es wird ebenfalls einen sehr ausführlichen Ausblick geben. Es konnten nicht alle nötigen Anforderungen mit dem Prototyp umgesetzt werden, daher gibt dieser Ausblick konstruktive Ideen und nötige weitere Schritte, um den Prototyp entsprechend erweitern zu können und die Anforderungen umzusetzen. Es werden ebenfalls zusätzliche mögliche Erweiterungen der Anwendung genannt.

9.1 Fazit

Das Ziel dieser Arbeit war es, die Verwaltung von Produktlinieninformation in Form einer Excel-basierten Variantenmatrix durch eine geeignete Anwendung zu optimieren und deren Akzeptanz und Nützlichkeit zu prüfen. Dazu wurde zunächst der gesamte Prozess der Erstellung und Verwaltung von Produktvarianten untersucht und Zusammenhänge mit der Variantenmatrix analysiert. Bei der Untersuchung konnten viele mögliche Probleme und kritische Abläufe herausgefunden werden, wobei die Variantenmatrix nur einen Teil der Problematik darstellt. Es wurden Anforderungen vieler Mitarbeiter gesammelt, untersucht und sortiert. Danach wurden Konzepte hinsichtlich der Architektur, Features der Anwendung und zu verwendenden Datenbanken analysiert. Aufbauend auf den Konzepten und den Anforderungen wurde ein Prototyp entwickelt. Der Prototyp beinhaltet grundlegende Funktionen des Suchens und Filterns, um Varianten zu finden und diese anschließend miteinander zu vergleichen. Dabei wurden die Bearbeitungszeiten gleicher Aufgaben in Excel und dem Prototyp gemessen sowie Äußerungen und Verhalten der Probanden notiert. Ein anschließender System Usability Score Fragebogen wurde von den Probanden ausgefüllt. Es stellte sich heraus, dass die Mitarbeiter bei der Durchführung der Aufgaben mit dem Prototyp in den meisten Fällen schneller waren, als mit der Excel-Tabelle. Vor allem Mitarbeiter, die wenig mit der Excel-Tabelle arbeiten, sind bis zu 300 % schneller (siehe Abbildung 7.3). Das Vergleichen von Varianten dauert bei sechs von sieben Probanden mit der Nutzung des Prototyps

zwischen 60 und 90 Sekunden, unabhängig der Nutzungshäufigkeit der Excel Variantenmatrix. Es ist zu beachten, dass sich bei Erhöhung der sehr kurzen Einführungszeit von fünf Minuten, die Bearbeitungszeiten nochmals verbessern könnten und eine Befragung von acht Personen für eine Verallgemeinerung nicht ausreichend ist. Die Ergebnisse somit nicht statistisch signifikant. Dennoch ist die klare Tendenz des Geschwindigkeitsvorteils sichtbar. Dies spiegelt sich auch im Ergebnis des System Usability Score Fragebogens wider. Dieser ergibt ein überdurchschnittlich gutes Ergebnis von 80.0.

Bei der Durchführung der Evaluation waren die Mitarbeiter sehr aufgeschlossen gegenüber der Anwendung und zeigten großes Interesse, damit in Zukunft zu arbeiten. Es konnte durch die Evaluation sogar ein skeptischer Mitarbeiter davon überzeugt werden, dass er seine Arbeit mindestens in gleicher Geschwindigkeit durchführen kann.

Eine Akzeptanz ist bei den Probanden vorhanden. Die Anwendung wird von den Mitarbeitern weiterhin akzeptiert werden, wenn deren Interessen und Bedürfnisse bei der Entwicklung berücksichtigt werden. Doch dazu müssen noch zusätzliche Funktionalitäten implementiert sowie weitere Anforderungen der Mitarbeiter erfasst und umgesetzt werden. Dies und weitere zu untersuchende Punkte werden in dem nachfolgenden Kapitelabschnitt genauer genannt.

9.2 Ausblick

Die Arbeit hat gezeigt, dass die Akzeptanz und ein großes Interesse an einer arbeitsunterstützenden Software bei den Mitarbeitern vorhanden sind. Auf Basis der gewonnenen Erkenntnisse bietet diese Arbeit eine Vielzahl weiterer Punkte, an denen sich durch Forschungen, Projekte sowie weitere Arbeiten anknüpfen lässt.

9.2.1 Variantenmatrix und Anwendung

Der Prototyp setzt zwar viele der in Kapitel 4 genannten Anforderungen um, doch wird das Erstellen, Verändern, Löschen und anderweitige Verwalten

von Varianten bisher nicht abgebildet. In dieser Arbeit werden einige der dafür benötigten Anforderungen genannt, die aber vollständig sein müssen. Das Hinzufügen von Varianten wird aktuell über ein Skript ausgeführt (Abschnitt 3.5.1). Die Vielzahl der dabei verwendeten Informationsquellen sind weder genauer analysiert noch dokumentiert worden. Es müssen ebenfalls Schnittstellen für das Hinzufügen von Varianten bereitgestellt werden. Die Datenbank muss in diesem Fall ebenfalls dafür vorbereitet werden. Durch die verwendete schemalose Datenbanklösung muss besonders auf die Benennung der Attribute geachtet werden (Abschnitt 2.4.1). Ein besonderer Modus, der es beim Hinzufügen von Varianten erlaubt, das Schema zu ändern, wäre denkbar. Dieser könnte bei der Übertragung der Daten angestoßen werden, sobald es Schemaänderungen gibt. So könnte sichergestellt werden, dass es bei neuen Varianten nicht zu falsch benannten Attributen kommen kann, sondern nur dann, wenn dies auch gefordert wird.

Das Bearbeiten von vorhandenen Varianten muss ebenfalls durch Eingabemasken gelöst werden. Formulareingaben und deren Überprüfungen können durch verschiedene Mechanismen ausgeführt werden. Formulare sind ein ungeliebtes Mittel, oft aber von zentraler Bedeutung (Jacobsen, 2014 S. 166).

Das Bearbeiten von Varianten sollte wie das Hinzufügen von Varianten nur durch bestimmte Benutzergruppen erfolgen. Eine Benutzerverwaltung ist daher elementar und wichtig, um Berechtigungen für bestimmte Mitarbeiter einzuschränken. So sollen die meisten Nutzer nur eine Leseberechtigung besitzen (Abschnitt 3.9). Die Benutzerverwaltung kann dabei der Einstellung von Berechtigungen, zeitbeschränkten Zugriffen oder ähnlichem dienen.

Es wurden ebenfalls keine erweiterten Anforderungen für das Anmelden, Registrieren, etc. verfasst und umgesetzt. Die Anwendung sollte dem Benutzer das Ändern bzw. Zurücksetzen des Passworts ermöglichen. Neu registrierte Benutzer müssen freigeschaltet werden. Der Aufwand dieser Mechanismen darf dabei nicht unterschätzt werden.

Darüber hinaus wünschen sich alle Probanden der Studie eine Anpassbarkeit der sichtbaren Filter, Suchergebnisse sowie anderer sichtbarer Informationen. Es wurde sowohl während der Evaluation als auch der

vorhergegangenen Interviews festgestellt, dass sich die Anforderungen an die sichtbaren Informationen nicht auf Benutzergruppen pauschalisieren lassen. Jeder Mitarbeiter nutzt andere Informationen, sucht und filtert anders. Ob die Mitarbeiter die Suche, den Filter, die Suchergebnisse oder andere Ansichten der Anwendung selber anpassen können oder ein gar lernendes System verwendet werden könnte, kann untersucht werden. Eine Anpassbarkeit der Ansichten würde die Produktivität und Nutzungsgeschwindigkeit sehr wahrscheinlich erhöhen.

Die Suche nach Varianten kann ebenfalls mit vielen weiteren Funktionen erweitert werden. Eine Volltextsuche würde es ermöglichen, die Suche zu erweitern oder einzugrenzen. Die Suche nach z.B. einer Varianten-ID könnte durch die Eingabe von „id:“ und einer nachfolgenden ID genauere Ergebnisse liefern. Aktuell werden nur übereintreffende Ergebnisse gezeigt. Es könnte ebenfalls sinnvoll sein, ähnliche Ergebnisse nach Relevanz sortiert anzuzeigen. Die Suche wird sehr wahrscheinlich eines der zentralen Funktionen der Anwendung sein.

Bei der Umsetzung des Prototyps hat sich gezeigt, dass es einige Filtergruppen mit sehr vielen Elementen gibt. Der Filter wirkt bei sehr vielen Elementen unübersichtlich. Es sollte untersucht werden, ob eine andere Anordnung oder die Verwendung eines anderen visuellen Elementes diese Filtergruppen besser darstellt.

Einigen Mitarbeitern ist ebenfalls die Nachverfolgbarkeit von Änderungen wichtig. Gleiches gilt für die Darstellung von Produktfamilien oder die „Stammbäume“ der Komponenten. Diese Informationen sind nach aktuellem Stand weder dokumentiert, noch den meisten Mitarbeitern bekannt. Hier sollte eine Nutzung der Graph-Datenbanken genauer analysiert werden, die Beziehung und Nachverfolgbarkeit von Änderungen einfacher abbilden kann (Abschnitt 2.4.2). Mit Hilfe dieser Informationen könnten heuristische Regeln möglich sein. In der aktuellen Form der Variantenmatrix fehlen zum einen zu viele Informationen, zum anderen müssen die Daten und das Schema so variabel sein. Die Nutzung von heuristischen Regeln würde die Mitarbeiter laut eigenen Aussagen eher hindern, anstatt sie zu unterstützen.

Es wurden bisher nur die in dieser Arbeit genannten Abteilungen und Benutzergruppen analysiert und deren Zusammenhänge mit der Variantenmatrix aufgedeckt. Wie in Abschnitt 3.9 erkannt, wird die Variantenmatrix von sehr vielen weiteren Mitarbeitern benutzt. Die Variantenmatrix ist ein wichtiges Dokument und Informationsquelle für sehr viele weitere Prozesse. Es ist nicht auszuschließen, dass es weitere Abteilungen gibt, die mit Hilfe von Skripten oder auf manuelle Weise Informationen aus der Variantenmatrix kopieren und diese in, eigens gepflegten Excel-Dateien weiterverarbeiten. Dies muss untersucht werden. Eventuell müssen weitere Schnittstellen bereitgestellt werden und deren Anforderungen in die der Anwendung aufgenommen werden. Es müssen weitere Mitarbeiter in die Entwicklung der Anwendung miteinbezogen werden.

Ein weiteres bisher ungelöstes Problem ist der Zugriff durch den Kunden. Der Kunde sollte sich mithilfe der Anwendung eine Variante „bauen“ können, ähnlich einem modularen Baukasten. Er sollte direkt Feedback von der Anwendung erhalten, welche Funktionen er integrieren und kombinieren kann. Dazu bedarf es der Definition von anpassbaren Regeln, denn der Kunde soll auch Bestellungen von Varianten aufnehmen können, die vielleicht noch gar nicht existieren (oder erst durch ein CRQ dazu kommen). Das heißt, der Baukasten muss beliebig erweitert werden können. Da die Kunden weltweit auf die Anwendung zugreifen, muss sie mehrere Sprachen unterstützen sowie möglicherweise ein right-to-left (RTL) Layout anbieten.

9.2.2 Der Prozess

Die Gefahr, dass falsche Informationen kopiert, oder die Informationen falsch kopiert werden, ist weiterhin vorhanden und wird durch die in dieser Arbeit vorgestellten Lösung nicht behoben. Es sollte weiter untersucht werden, inwiefern sich Informationen aus anderen Excel-Dateien und Tabellen, als auch Teilprozessen in die Anwendung integrieren lassen, sodass es weniger redundante Informationsquellen gibt. Dabei müssen ebenfalls benötigte Features der einzelnen Abteilungen integriert werden sowie Schnittstellen zu weiteren, bisher unbekannten Prozessen bereitgestellt werden. Durch die Integration müssten Informationen nicht mehr manuell zwischen unterschiedlichen Dokumenten ausgetauscht werden. Eine sinkende

Fehlerrate beim Kopieren von Informationen würde Kosten einsparen und gleichzeitig die Produktivität der Mitarbeiter steigern.

Ebenfalls ist es sinnvoll, die in dieser Arbeit erkannten möglichen Probleme genauer zu analysieren und zu untersuchen. Einige dieser sind als kritisch anzusehen, da sie Gefahren für den Gesamtprozess bergen.

Es wurde ebenfalls nicht analysiert, inwieweit das Problem der vielen vernetzten Excel-Sheets in anderen Unternehmensbereichen vorhanden ist und wie diese damit umgehen. So könnte es identische Probleme innerhalb des Unternehmens geben und deren Lösungen auf andere Bereiche übertragen werden.

9.2.3 Studie und Tests

Bei der Evaluation der Usability wurden bisher nur die Metriken Erfolgsrate und Zeitaufwand gemessen. Es konnte bisher keine statistische Signifikanz festgestellt werden. Dazu müssten weitere Tests durchgeführt werden. Eine Auswirkung der Einarbeitungszeit auf die Ergebnisse ist ebenfalls nicht auszuschließen und sollte genauer betrachtet werden.

Es könnten ebenfalls Studien durchgeführt werden, welche die Veränderung der Fehlerrate bei Vergleichen von Varianten messbar machen.

Im Laufe der Entwicklung der Anwendung sollten Usability-Tests durchgeführt werden, um Kontrolle über die Qualität und ein aktives Feedback der Anwender zu erhalten.

Der Prototyp wurde bisher nicht durch Unit³⁰-Tests, Integrationstests oder User Interface (UI)-Tests überprüft. Testverfahren sollten aber bei der Entwicklung der Anwendung eingesetzt werden, vor allem bei kritischen Codeabschnitten wie dem Import/Export von Excel Dateien, möglichen Schnittstellen oder anderen komplexen Funktionen.

Zusammenfassend lässt sich sagen, dass es äußerst kritisch ist, nur einen Teil eines höchst komplexen Business Prozesses zu ersetzen. Es wurden bisher weder alle Stakeholder des international agierenden Prozesses analysiert, noch wurden wichtige Fragen wie der Zugriff des Kunden geklärt.

³⁰ Test eines Codeabschnitts / Softwaremoduls.

Außerdem wurden weder weitere wichtige Funktionen getestet, noch wurden alle benötigten Anforderungen dafür erfasst. Doch lässt sich sagen, dass viele Mitarbeiter eine unterstützende Anwendung dringend fordern und benötigen. Das Feedback der Mitarbeiter und die Ergebnisse der Studie haben gezeigt, dass eine Werkzeugunterstützung durch eine geeignete Anwendung akzeptiert und gewünscht wird.

Die Variantenmatrix ist aber nur ein Teil des Prozesses und verursacht auch nur einen Teil der Probleme. Durch eine Softwareunterstützung des Gesamtprozesses könnten vor allem die Probleme der mehrfach vorhandenen Informationsquellen, sowie die damit verbundene Fehlerrate und nötige Kommunikation minimiert, als auch die Produktivität gesteigert werden.

Verzeichnisse

Quellen- und Literaturverzeichnis

Amalfitano Domenico [et al.] Migrating legacy spreadsheets-based systems to Web MVC architecture: An industrial case study [Konferenz] // 2014 Software Evolution Week - IEEE Conference on Software Maintenance, Reengineering, and Reverse Engineering (CSMR-WCRE). - Antwerpen, Belgien : IEEE, 2014. - S. 387-390.

Bangor Aaron, Kortum Philip T. und Miller James T. An Empirical Evaluation of the System Usability Scale [Artikel] // International Journal of Human-Computer Interaction. - [s.l.] : Taylor & Francis, 2008. - 6.

Bianchi A. [et al.] Iterative Reengineering of Legacy Systems [Buchabschnitt] // IEEE Transactions on Software Engineering. - 2003.

Brooke John SUS: a "quick and dirty" usability scale [Buchabschnitt] // Usability Evaluation In Industry / Buchverf. Jordan Patrick W., Thomas B. und McClelland Ian Lyall. - London : Taylor and Francis, 1996.

Chevez Bernaldo de Quiros A. [et al.] Human Reliability as a Source of Error in Research [Konferenz] // 24th Annual Conference of ARCOM. - Cardiff, UK : [s.n.], 2008. - S. 687-696.

Cockburn Alistair Writing Effective Use Cases [Buch]. - University of Michigan : Addison-Wesley, 2001. - ISBN: 978-0201702255.

Codd Dr. Edgar F. The Relational Model for Database Management [Buch]. - [s.l.] : Addison-Wesley Publishing Company, Inc., 1990. - 2..

Ebert Christof Systematisches Requirements Engineering und Management [Buch]. - Heidelberg : dpunkt.verlag GmbH, 2008. - ISBN: 978-3898645461.

Edlich Stefan [et al.] NoSQL Einstieg in die Welt nichtrelationaler Web 2.0 Datenbanken [Buch]. - München : Carl Hanser Verlag, 2010.

Garreau Marc und Faurot Will Redux in Action [Buch]. - Shelter Island, New York : Manning Publications Co., 2018.

- Glewe Paul** Verbesserung Spreadsheet-basierter Ansätze zum Variantenmanagement mit Hilfe von Datenbanktechnologien. - Hildesheim : Universität Hildesheim, 2019. - Bachelorarbeit, Bereich Intelligente Informationssysteme - Prof. Dr. Klaus-Dieter Althoff.
- Jablonski Stefan [et al.]** Guide to Web Application and Platform Architectures [Buch]. - Erlangen : Springer-Verlag, 2004.
- Jacobsen Jens und Meyer Lorena** Praxisbuch Usability und UX [Buch]. - Bonn : Rheinwerk Verlag GmbH, 2017. - 1. Auflage. - ISBN: 978-3-83624423-7.
- Jacobsen Jens** Website-Konzeption [Buch]. - Heidelberg : dpunkt.verlag GmbH, 2014.
- Kortum Philip T. und Bangor Aaron** Usability ratings for everyday products measured with the System Usability Scale [Buchabschnitt] // International Journal of Human-Computer Interaction. - [s.l.] : Taylor & Francis, 2013.
- Lewis James R.** The System Usability Scale: Past, Present, and Future // International Journal of Human–Computer Interaction. - [s.l.] : Taylor & Francis, 2018. - Bd. 34.
- Nielsen Jakob und Mack Robert L.** Usability Inspection Methods [Buch]. - New York : John Wiley & Sons, 1994. - ISBN: 0-471-01877-5.
- OMG (Object Management Group)** OMG [Online] = Business Process Model And Notation // BPMN Specification 2.0. - Object Management Group, 12 2011. - 2.0. - 12. 05 2019. - <https://www.omg.org/spec/BPMN/2.0/PDF>.
- Parker Zachary, Poe Scott und Vrbsky Susan V.** Comparing NoSQL MongoDB to an SQL DB [Konferenz] // Proceedings of the 51st ACM Southeast Conference. - Savannah, Georgia : [s.n.], 2013.
- Provos Niels und Mazières David** A future-adaptable password scheme [Konferenz] // USENIX Annual Technical Conference. - Monterey, California, USA : [s.n.], 1999. - S. 81-92.
- Rubin Jeffrey** Handbook of Usability Testing [Buch]. - Kanada : John Wiley & Sons, Inc., 1994. - ISBN: 978-0471594031.

- Rupp Chris** Requirements-Engineering und -Management [Buch]. - [s.l.] : Carl Hanser Verlag GmbH & Co. KG, 2014. - Bd. 6. Auflage.
- Sarodnick Florian und Brau Henning** Methoden der Usability Evaluation: Wissenschaftliche Grundlagen und praktische Anwendung [Buch]. - [s.l.] : Hogrefe AG, 2011.
- Schneider Prof. Dr. Kurt** Requirements-Engineering (Vorlesung) - Leibniz Universität Hannover. - Hannover : [s.n.], 2018.
- Schneier Bruce** Fast Software Encryption [Konferenz] // Cambridge Security Workshop Proceedings. - [s.l.] : Springer-Verlag, 1994.
- Schumann H. [et al.]** Concurrent Systems Engineering in Aerospace: From Excel-based to Model Driven Design [Konferenz] // 8th Conference on Systems Engineering Research. - Hoboken, New Jersey : [s.n.], 2010.
- Silver Bruce** BPMN Method and Style [Buch]. - [s.l.] : Cody-Cassidy Press, 2011. - ISBN: 978-0982368114.
- Stefanov S. und Demmig T.** Durchstarten mit React: Web-Apps einfach und modular entwickeln [Buch]. - [s.l.] : O'Reilly, 2017.
- Wohlin Claes [et al.]** Experimentation in Software Engineering [Buch]. - [s.l.] : Springer-Verlag Berlin Heidelberg, 2012.

Tabellen- und Abbildungsverzeichnis

Abbildung 2.1 BPMN Basiselemente	3
Abbildung 2.2 Document Store Beispiel	7
Abbildung 2.3 Graphdatenbank Beispiel	8
Abbildung 3.1 Prozessübersicht	10
Abbildung 3.2 Erweiterter Prozessablauf.....	12
Abbildung 3.3 Variantenmatrix: Index, Teilenummer, Scope, Region	14
Abbildung 3.4 Variantenmatrix: Status, SOP, Kunde, Typ.....	15
Abbildung 3.5 Variantenmatrix: Front, Sample Phase, PCB, Tuner	16
Abbildung 3.6 Variantenmatrix: Speicher, Externe Anschlüsse, Software.....	17
Abbildung 3.7 Spezifikation Prozess mit Kunden	20
Abbildung 3.8 Ankunft der Spezifikation	21
Abbildung 3.9 Spezifikation überprüfen	22
Abbildung 3.10 Variantenverwaltung: Überblick	26
Abbildung 3.11 Variantenverwaltung: Erstellung / Änderung einer Variante	26
Tabelle 1 Zugriffsstatistik Excel Variantenmatrix	34
Tabelle 2 Anforderung an den Prototyp, sortiert.....	51
Use Case 1: Suchen und Filtern.....	52
Use Case 2: Vergleichen.....	53
Abbildung 5.1 UML Informationsfluss zwischen Dateien der Abteilungen	56
Abbildung 5.2 Suchvervollständigung Google	57
Abbildung 5.3 Filterfunktion Preisvergleich Geizhals.....	58
Abbildung 5.4 Vergleichsfunktion Webshop Notebooksbilliger	59
Abbildung 5.5 Mockup Prototyp Suchen / Filtern.....	60
Abbildung 5.6 Vier-Stufen-Architektur	61
Abbildung 5.7 Model View Controller Entwurfsmuster	62
Abbildung 6.1 Prototyp Authentifizierung.....	66
Abbildung 6.2 Prototyp Suche und Filter Startseite	67
Abbildung 6.3 Prototyp Suche und Filter aktiv	68
Abbildung 6.4 Prototyp Suchvorschläge.....	68
Abbildung 6.5 Prototyp Vergleichen von Varianten	69
Abbildung 6.6 Redux Architektur	70
Abbildung 7.1 Ablauf zu bearbeitender Aufgaben	76

Tabelle 3 Nutzung der Excel Variantenmatrix der Probanden	80
Tabelle 4 Mittelwert, Standardabweichung Aufgabe 3 und 4	83
Abbildung 7.3 Diagramm Bearbeitungszeit der Aufgaben 5 und 6 im Vergleich (ohne P6)	84
Tabelle 5 Mittelwert, Standardabweichung Aufgabe 5 und 6	86
Abbildung 7.4 Vergleich SUS Werte nach Quartil, Akzeptanz und Bewertung	87
Anhang Abbildung 1 Einverständniserklärung	107
Anhang Abbildung 2 Aufgaben	108
Anhang Abbildung 3 Fragebogen Seite 1	109
Anhang Abbildung 4 Fragebogen Seite 2	110
Anhang Tabelle 1 Bearbeitungszeiten Aufgaben 3, 4, 5 und 6	111
Anhang Tabelle 2 SUS Auswertung	112

Anhang

Einverständniserklärung und Fragebogen

Einverständniserklärung Testperson Usability-Test

Vielen Dank, dass Sie mir helfen, das Variantenmatrix-Tool, welches ich Ihnen gleich zeigen werde, zu verbessern.

Um mir die Arbeit zu erleichtern, werde ich den Test schriftlich dokumentieren. Die Aufzeichnung wird nur intern zugänglich gemacht, um die Testergebnisse zu demonstrieren. Die Aufzeichnungen werden nicht veröffentlicht. Die Ergebnisse werden anonymisiert.

Bitte lesen Sie die folgende Erklärung und unterschreiben Sie darunter. Vielen Dank.

Ich gestatte es ausdrücklich, die schriftlichen Aufzeichnungen zu Zwecken der Untersuchung und Demonstration der Testergebnisse intern zu verwenden.

Name, Vorname (bitte in DRUCKBUCHSTABEN): _____

Datum, Ort: _____

Unterschrift: _____

Unterschrift: _____

Anhang Abbildung 1 Einverständniserklärung

Usability-Test Variantenmatrix – Aufgaben

Mit Ihrem Test helfen Sie mir, die Web Version der Variantenmatrix zu verbessern. Im Folgenden bekommen Sie Aufgaben, welche Sie während des Tests ausführen sollen.

Es wird Aufgaben geben, welche mit der Excel Variantenmatrix zu lösen sind. Diese sind mit (EXCEL) markiert. Ebenfalls folgen Aufgaben, die mit der Web-Variante zu lösen sind. Diese sind mit (WEB) markiert.

Aufgaben müssen nicht vollständig gelöst werden. Bei Bedarf kann der Test auch abgebrochen werden – teilen Sie dies dem Protokollanten mit.

Während der Tests werden Zeiten gemessen. Diese werden nicht verwendet, um Leistungen von Ihnen zu testen.

Ziel des Tests ist es, Fehler oder Probleme zu finden. Versuchen Sie während des Tests laut zu denken.

Aufgabe 1 (WEB):

Machen Sie sich mit der Web-Version vertraut. Suchen sie mindestens zwei Varianten und stellen Sie Unterschiede zwischen diesen fest.

Aufgabe 2 (EXCEL):

Suchen Sie mindestens zwei Varianten und stellen Sie Unterschiede zwischen diesen fest.

Aufgabe 3 (EXCEL):

Sie haben ein Gerät mit der Bosch-Teilenummer 7515750970 auf ihrem Tisch liegen. Zu welcher Carline gehört dieses Gerät.

Aufgabe 4 (WEB):

Sie haben ein Gerät mit der Bosch-Teilenummer 7515751100 auf ihrem Tisch liegen. Zu welcher Carline gehört dieses Gerät.

Aufgabe 5 (EXCEL):

Erkennen Sie Unterschiede zwischen dem Gerät mit der Bosch Teilenummer 7515750970 und 7515751100 und zeigen diese.

Finden Sie alle Geräte der Carline P32R.

Aufgabe 6 (WEB):

Erkennen Sie Unterschiede zwischen dem Gerät mit der Bosch Teilenummer 7515750970 und 7515751100 und zeigen diese.

Finden Sie alle Geräte der Carline P32R und exportieren Sie diese anschließend.

Anhang Abbildung 2 Aufgaben

Usability-Test Variantenmatrix – Abschließender Fragebogen

Mit Ihrem Test helfen Sie mir, die Web Version der Variantenmatrix zu verbessern, die Sie soeben besucht haben. Zum Abschluss bitte ich Sie, diesen Fragebogen auszufüllen, um mir mitzuteilen, wie Ihnen die Version gefallen hat.

Bitte Kreuzen sie die an:

Aussage	STIMME GAR NICHT ZU					STIMME VOLL ZU
Ich kann mir sehr gut vorstellen, das System regelmäßig zu nutzen.	☐	☐	☐	☐	☐	☐
Ich empfinde das System als unnötig komplex.	☐	☐	☐	☐	☐	☐
Ich empfinde das System als einfach zu nutzen.	☐	☐	☐	☐	☐	☐
Ich denke, dass ich technischen Support brauchen würde, um das System zu nutzen.	☐	☐	☐	☐	☐	☐
Ich finde, dass die verschiedenen Funktionen des Systems gut integriert sind.	☐	☐	☐	☐	☐	☐
Ich finde, dass es im System zu viele Inkonsistenzen gibt.	☐	☐	☐	☐	☐	☐
Ich kann mir vorstellen, dass die meisten Leute das System schnell zu beherrschen lernen.	☐	☐	☐	☐	☐	☐
Ich empfinde die Bedienung als sehr umständlich.	☐	☐	☐	☐	☐	☐
Ich habe mich bei der Nutzung des Systems sehr sicher gefühlt.	☐	☐	☐	☐	☐	☐
Ich musste eine Menge Dinge lernen, bevor ich mit dem System arbeiten konnte.	☐	☐	☐	☐	☐	☐

Anhang Abbildung 3 Fragebogen Seite 1

Weitere Anmerkungen / Ideen oder Kommentare:

Ergebnisdaten

Aufgabe	P5	P8	P2	P6	P3	P7	P4	P1
T3 (Excel)	25	25	32	36	39	45	45	49
T4 (Web)	20	25	30	23	56	18	35	35
T5 (Excel)	80	62	116		92	160	260	270
T6 (Web)	90	70	62		62	200	87	90
Nutzung	alle zwei Tage	täglich	täglich	alle drei Tage	täglich	alle drei Tage	monatlich	monatlich

Farbzuordnung:

Gruppe 1 Gruppe 2 Gruppe 3

Anhang Tabelle 1 Bearbeitungszeiten Aufgaben 3, 4, 5 und 6

	P1	P2	P3	P5	P6	P7	P8	Ø
Q1	4	2	2	4	3	4	4	3,285714
Q2	4	4	4	3	2	3	3	3,285714
Q3	4	3	3	3	2	4	3	3,142857
Q4	4	2	3	3	3	3	3	3
Q5	4	4	1	4	2	3	4	3,142857
Q6	4	3	2	3	2	3	4	3
Q7	4	3	0	3	4	3	1	2,571429
Q8	4	3	4	4	3	4	3	3,571429
Q9	4	3	4	3	3	3	3	3,285714
Q10	4	4	4	4	3	3	4	3,714286
Σ	40	31	27	34	27	33	32	32
SUS	100	77,5	67,5	85	67,5	82,5	80	80

Anhang Tabelle 2 SUS Auswertung

Inhalt der beiliegenden CD

Die beiliegende CD beinhaltet folgende Materialien:

- Masterarbeit in digitaler Form (PDF)
- Die in dieser Arbeit verwendeten Abbildungen, Screenshots und Diagramme
- Quellcode des Prototyps
- Evaluationsdaten:
 - Fragebögen
 - Protokolle
 - Ergebnisse
- Beispiel der Variantenmatrix als Excel Tabelle

Danksagungen

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während dieser Masterarbeit unterstützt haben.

Zuerst möchte ich mich bei allen Mitarbeitern der Robert Bosch Car Multimedia GmbH in Hildesheim bedanken, die mich während der Masterarbeit betreut sowie an den vielen Interviews und zahlreichen Usability-Tests teilgenommen haben. Besonderen Dank gilt José, Uwe und Thomas für die Organisation sowie den hilfreichen Anregungen und konstruktiven Kritiken.

Ebenfalls möchte ich mich bei meinen Betreuern Melanie und Wasja bedanken, ihr habt euch bei den wöchentlichen Treffen immer sehr viel Zeit genommen. Eure konstruktiven Kritiken, Ideen und Anregungen haben mir bei der Durchführung der Arbeit sehr geholfen.

Ich möchte mich bei meiner Freundin Claudia bedanken, die auch in schwierigen Situationen immer für mich da war. Trotz eigener Abschlussarbeit hat sie mich großartig unterstützt.

Besonders möchte ich mich bei allen meinen Freunden bedanken. Danke für die Partys, Festivals, Kanutouren, Motorradtouren und lustigen Abende. Die Ablenkung tat mir sehr gut.

Außerdem möchte ich Jürgen für das Korrekturlesen meiner Masterarbeit danken.

Vielen Dank an alle Mitarbeiter sowie ehemaligen Mitarbeiter von Smartjobr. Ohne eure Unterstützung wäre eine Arbeit in diesem Umfang nicht möglich geworden.

Abschließend möchte ich mich bei meinen Eltern und meiner Familie bedanken, die mir mein Studium ermöglicht haben und mich immer unterstützen.

