

# **Fehleranalysestrategien für digitale Dienste in der Kunden Nutzungsphase**

vorgelegt von

M.Sc.

Yanfu Lu

an der Fakultät V - Verkehrs- und Maschinensysteme

der Technische Universität Berlin

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften

- Dr.-Ing. -

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr.-Ing. Franz Dietrich

Gutachter: Prof. Dr.-Ing. Robert Dust

Gutachter: Prof. Dr. Elmar Bräkling

Tag der wissenschaftlichen Aussprache: 21. Oktober 2019

Berlin 2020



## **Abstract**

Das übergeordnete Ziel dieser Dissertation ist die wissenschaftliche Forschung und Entwicklung der Fehleranalysestrategien für digitale Dienste in der Automobilindustrie.

Im theoretischen Teil dieser Arbeit werden die Grundlagen von Software Qualitätsmanagement, digitalen Diensten sowie Standards des Fehlermanagements in der Automobilindustrie und IT-Branchen beleuchtet. In dieser Arbeit wurde „Fehler“ als die Nichterfüllung von verfügbaren funktionalen Produktmerkmalen sowie die Abweichung von Kundenerwartungen definiert. Die Kundenerwartung für digitale Dienste wurde mittels einer Benchmark Analyse von digitalen Produkten in der Automobilindustrie und Mobilitätsdienstleistungs-Unternehmen untersucht.

Für die Beantwortung der Forschungsfrage wurden neben theoretischen Fundierungen des Untersuchungsgegenstands empirische Erhebungen vorgenommen. Um die Qualitätsaspekte und -Standards in der Früh- und Nutzungsphase der digitalen Dienste weiterzuentwickeln, wurden verschiedene Industrieinterviews und Befragungen mit klassischen Automobilherstellern sowie etablierten neuen globalen Playern, IT-Unternehmen und Start-ups durchgeführt.

Abschließend konnten Handlungsempfehlungen für die Gestaltung von Fehleranalysestrategien im Bereich der digitalen Dienste abgeleitet werden. Einerseits ist die zukunftsgerichtete Selektion passgenauer Kooperationspartner und die Befolgung von Qualitätsstandards zu nennen, andererseits ist die Entwicklung einer agilen Qualitätskultur und -methodik zu empfehlen. Die beschriebenen Aspekte sind maßgeblich, um die zukünftige Wettbewerbsfähigkeit zu gewährleisten. Es geht darum, Entwicklungszeiten zu verkürzen und flexibel auf neue Kundenanforderungen reagieren zu können. Diese Fähigkeiten gelten als wichtige Erfolgsfaktoren für Qualitätsstrategien digitaler Dienste.

## **Abstract (English)**

The main goal of this dissertation is to research and develop issue analysis strategies for digital services in the automotive industry.

In the theoretical section of this thesis, the basics of software quality management and digital services, as well as the standards of issue management in the automotive and IT industries are examined. In this thesis, “issue” was defined as the non-fulfillment of available functional product features and deviation from customer expectations. Customer expectations for digital services were measured using benchmarking analysis of digital products in the automotive industry and mobility service companies.

In addition to exploring the theoretical foundations of the object of investigation, empirical surveys were carried out to answer the research question. In order to further develop the quality aspects and standards in the early and usage phases of the digital services, various industry interviews and questionnaires were completed with classic car manufacturers as well as established new global players, IT companies, and start-ups.

Finally, recommendations for action steps in the design of issue analysis strategies in the field of digital services have been derived. On the one hand, the future-oriented selection of tailor-made quality standards has been set forth as an imperative for cooperation partners; on the other, the development of agile quality culture and methodology is recommended. The skills described are essential to ensure future competitiveness. Shortening development times and being able to react flexibly to new customer requirements are considered important success factors for implementing quality strategies in digital services.

## **Vorwort**

Die vorliegende Arbeit entstand während meiner Tätigkeit als Gastwissenschaftler im Fachgebiet Qualitätsstrategie und Qualitätskompetenz im Institut für Werkzeugmaschinen und Fabrikbetrieb an der Technischen Universität Berlin sowie als Industrie-Doktorand im Bereich Qualitätsmanagement bei Mercedes-Benz Cars in Sindelfingen.

Mein besonderer Dank gilt zunächst meinem Doktorvater Herrn Prof. Dr.-Ing. Robert Dust, der meine Arbeit stets mit viel Verständnis unterstützt und gefördert hat. Er gab mir die Möglichkeit zum selbständigen Arbeiten, regelmäßigen und offenen Diskussionen sowie zu wissenschaftlichen freien Entfaltungen.

Für konstruktive Anregungen danke ich ebenso Prof. Dr. Elmar Bräkling, der meine Doktorarbeit als zweiter Gutachter betreut hat. Weiterhin danke ich Herrn Prof. Dr.-Ing. Franz Dietrich für die Übernahme des Vorsitzes im Prüfungsausschuss.

Für vielfältige Unterstützung bin ich besonders Herrn Arne Ramm, Frau Elke Hammann und meinen Kolleginnen und Kollegen im Fachgebiet Qualitätsstrategie und Qualitätskompetenz der Technischen Universität Berlin verbunden.

Außerdem möchte ich mich bei Herrn Stefan Meyer und Frau Christina Wurz aus der Daimler AG in Sindelfingen bedanken. Sie haben mich vor allem zu Beginn meiner Doktorarbeit durch ihre industriellen Erfahrungen sowie Projektarbeiten unterstützt.

Schließlich bedanke ich mich bei allen Kolleginnen und Kollegen meiner ehemaligen Abteilungen aus dem Bereich Qualitätsmanagement in der Daimler AG in Sindelfingen für die Zusammenarbeit sowie Diskussionsbereitschaft in einer sehr angenehmen, kreativen Atmosphäre.

Mein spezieller Dank gilt meinen Eltern, die mich während meines Studiums und meiner Promotion in Deutschland mit großem Engagement unterstützt haben.

# Inhaltsverzeichnis

1	Einleitung.....	1
2	Motivation, Zielsetzung und Forschungsfragen.....	3
2.1	Unternehmenskultur und Geschäftsmodell.....	3
2.2	Tendenzen und Hypothesen in der Automobilindustrie.....	4
2.3	Chancen und Herausforderungen .....	8
2.4	Forschungsfragen der Dissertation.....	9
2.5	Aufbau der Dissertation.....	10
3	Grundlagen und Stand der Technik .....	11
3.1	Grundlagen Qualitätsmanagement.....	11
3.2	Grundlagen Software .....	13
3.2.1	Entstehung sowie Lebenszyklus eines Software-Produkts .....	14
3.2.2	Automotive SPICE .....	16
3.2.3	Softwaremerkmale .....	17
3.2.4	Qualitätsmerkmale von Software.....	18
3.2.5	Nutzungsqualität von Software.....	20
3.3	Grundlagen und Definition der digitalen Dienste.....	21
3.3.1	Definition digitaler Dienste in dieser Dissertation.....	22
3.3.2	Beispiele von Ökosystemen digitaler Produkte.....	24
3.4	Standards und Definitionen der Fehlerklassifizierung und des Fehlermanagements .....	25
3.4.1	Normen und Definitionen von Fehlern .....	26
3.4.2	Anforderungsmanagement.....	29
4	Stand der Forschung .....	32
4.1	Benchmarking .....	33
4.1.1	Grundlagen und Zielsetzung Benchmarking.....	33
4.1.2	Durchführung des Benchmarkings .....	34
4.1.3	Ergebnisse des Benchmarkings der digitalen Dienste.....	36
4.2	Kundenanforderungen.....	40
4.2.1	Grundlagen der Kundenanforderungen .....	40
4.2.2	Ergebnisse Benchmarking-Analyse und Kundenzufriedenheit bezüglich digitaler Dienste .....	42
4.3	Fehler und Fehlerarten der digitalen Dienste.....	44

4.3.1	Status quo des Fehlerabstellprozesses .....	45
4.3.2	Fehlermanagement-Methoden .....	47
4.3.3	Kritik an Fehlermanagement-Methoden .....	50
4.4	Umgang mit Reklamationen .....	52
4.4.1	Beschwerdemanagement (BM) .....	52
4.4.2	Ablauf 8D-Methode .....	53
4.4.3	Umgang mit Reklamationen in der Software-Branche .....	58
4.4.4	Standardisierter Reklamationsprozess in der Automobilindustrie .....	59
4.5	Neue Begriffe rund um die Fehleranalysestrategien digitaler Dienste.....	61
4.6	Agilität .....	63
4.6.1	Agiles Qualitätsmanagement nach DGQ .....	63
4.6.2	Werte und Prinzipien agiler Methoden.....	65
4.6.3	Agile Methode Scrum.....	66
4.6.4	DevOps.....	67
5	Anpassung der Rolle des Qualitätsmanagements im Hinblick auf agile Methoden im Bereich Connectivity .....	70
5.1	Zielsetzung und Forschungsfragen .....	70
5.2	Zusammenfassung der Analyse .....	71
5.2.1	Rahmenbedingungen für die Rolle des Qualitätsmanagers .....	74
5.2.2	Technische Rolle des Qualitätsmanagers .....	75
5.2.3	Konzeptionelle Rolle des Qualitätsmanagers 2.0 .....	77
5.3	Agile Organisationsstruktur schaffen .....	79
5.4	Zusammenfassung und Fazit .....	82
5.5	Ausblick .....	89
6	Strategien für die Frühphase der zukünftigen Zusammenarbeit digitale Dienste..	91
6.1	Zielsetzung.....	91
6.2	Vorstellung bisheriger Studien sowie Umfragen .....	92
6.2.1	Beschreibung des Datenerhebungsinstruments und des Erhebungsablaufs .....	93
6.2.2	Datenerhebung und Auswertung der Ergebnisse .....	94
6.2.3	Diskussion der Ergebnisse .....	98
6.3	Industrie-Interviews .....	99
6.4	Ergebnisse und Ergebnisdiskussion.....	100
6.5	Kritische Einordnung, Ausblick und Zusammenfassung .....	102

7	Agiler Fehlermanagementprozess für digitale Dienste .....	104
7.1	Ergebnisse der Interviews Leitfaden Fehlermanagement .....	104
7.2	Fehlerklassifizierung für digitale Dienste .....	107
7.3	Modelldarstellung des Fehlerbeseitigungsprozesses .....	110
7.3.1	Integration des Fehlerbeseitigungsmodells in den Produktlebenszyklus ... .....	110
7.3.2	Fehlererfassungsmaßnahmen in der IT-Branche .....	112
7.3.3	Auswertung von Kundenrezensionen .....	114
7.4	Weiterentwicklung des agilen Fehlermanagement-prozesses .....	116
7.4.1	Entwurf eines agilen Fehlermanagementprozesses .....	116
7.4.2	Allgemeine Ideen, Hypothesen und Ansätze .....	118
7.5	Weiterentwicklung Reklamationsmanagement als Teil des gesamten Fehleranalyseprozesses .....	120
7.5.1	Teilprozess Initialisierung .....	121
7.5.2	Teilprozess 4D+1-Methode .....	124
7.5.3	Teilprozess Verifikation/Abschluss .....	126
7.5.4	Teilprozess Beanstandung ablehnen .....	129
7.5.5	Teilprozess Produktrückruf.....	129
7.5.6	Ergebnisse und Zusammenfassung .....	130
7.6	Herausforderungen und Risiken .....	131
7.7	Kritische Reflexion .....	133
8	Zusammenfassung und Zukunftsausblick .....	134
9	Literatur .....	137
10	Anhang .....	146

# Abbildungen Verzeichnis

Abbildung 1.1: Allgemeine Darstellung Qualitätsmanagements digitale Dienste in der Automobilindustrie	2
Abbildung 2.1: The five core areas of the new software organization	4
Abbildung 2.2: Änderung der Wettbewerber und des Geschäftsmodells in der Automobilindustrie	6
Abbildung 2.3: Fahrzeuge mit Internetverbindung weltweit von 2012–2020 in Milliarden	7
Abbildung 2.4: Anteil vernetzter Fahrzeuge bis 2025	7
Abbildung 2.5: Zukünftiges Zusammenarbeitsmodell im Feld Connectivity	8
Abbildung 3.1: Anforderung, Ergebnis, Qualität	11
Abbildung 3.2: Entwicklungsstufen des Qualitätsmanagements	12
Abbildung 3.3: Lebenszyklus eines Software-Produkts	14
Abbildung 3.4: Zusammenhang zwischen innerer sowie äußerer Software-Qualität und Nutzungsqualität	21
Abbildung 3.5: Matching der Eigenschaften mit den Schnittstellen von digitalen Diensten	22
Abbildung 3.6: Neue Auto Mobilität	24
Abbildung 3.7: Relative Kosten für die Beseitigung eines Software-Fehlers	30
Abbildung 3.8: Eingeführte Software-Fehler in der Anforderungsentwicklung	30
Abbildung 4.1: Benchmarking-Arten im Vergleich	33
Abbildung 4.2: Ranking nach Leistungsvergleich	38
Abbildung 4.3: Spannungsdreieck nach Masing	41
Abbildung 4.4: Auszug der Funktionsübersicht nach Herstellern	42
Abbildung 4.5: Vergleich Kundenbewertungen Mercedes-me- und Tesla-App	43
Abbildung 4.6: Ranking nach Automobilhersteller	43
Abbildung 4.7: Fehlerabstellprozess im Aachener Qualitätsmanagementmodell	45
Abbildung 4.8: VDA Blauer Band, Schadteilanalyseprozess	46
Abbildung 4.9: Vergleich des ganzheitlichen Schadenteile-Rückführungsprozesses und desjenigen digitaler ‚Komponenten‘	47
Abbildung 4.10: Six-Sigma-DMAIC-Zyklus	49
Abbildung 4.11: Fehler Klassifizierungsprozess für Software in der Automobilindustrie	50
Abbildung 4.12: Integration des Fehler-Analyseprozesses für digitale Dienste in der Automobilindustrie	51
Abbildung 4.13: Quellen der Felddaten klassische Komponenten	52
Abbildung 4.14: Partner im Produktentstehungsprozess	53
Abbildung 4.15: Prozessdarstellung 8D	54

Abbildung 4.16: Fehleranalyse Pyramide & Quality-Time-Feature Dreieck	62
Abbildung 4.17: Rolle des Fehlermanagements im gesamten Produktlebenszyklus	62
Abbildung 4.18: Ablauf des Scrum-Entwicklungsprozesses	67
Abbildung 5.1: Einbindung der Rolle QM 2.0 in die Entwicklungsarbeit	74
Abbildung 5.2: Technische Rolle des Qualitätsmanager 2.0	76
Abbildung 5.3: Konzeptionelle Rolle des Qualitätsmanagers 2.0	77
Abbildung 5.4: Produktzelle und zellulare Organisation	80
Abbildung 6.1: Bisheriger Kooperationsstand von Start-ups	95
Abbildung 6.2: QM-Zertifizierung in Start-ups	95
Abbildung 6.3: QM-Stellen in Start-ups	96
Abbildung 6.4: QM-Rollen in Entwicklungsteams	96
Abbildung 6.5: Referenzmodelle in Start-ups	97
Abbildung 6.6: Vorgehensmodelle in Start-ups	97
Abbildung 6.7: Evaluation des Vorgehens in Start-ups	98
Abbildung 7.1: Fehler-ID Phasenmodell	108
Abbildung 7.2: Entwurf eine agilen Fehlermanagementprozesses	118
Abbildung 7.3: Übermittlung der Beanstandungsdaten an den OEM	122
Abbildung 7.4: Datenübermittlung an den Lieferanten	122
Abbildung 7.5: Prüfung auf sicherheitsrelevante Folgen	123
Abbildung 7.6: Vorprüfung der Fehlerdaten	123
Abbildung 7.7: Kategorisierung und Priorisierung der Fehlerdaten	123
Abbildung 7.8: D3: Sofortmaßnahmen	124
Abbildung 7.9: D4: Ursachenanalyse	124
Abbildung 7.10: Prüfung der Beanstandung	125
Abbildung 7.11: D5+D6: Erarbeitung und Realisierung von Abstellmaßnahmen	125
Abbildung 7.12: Testen der Abstellmaßnahmen	126
Abbildung 7.13: Prüfung der Stellungnahme	127
Abbildung 7.14: Zurückweisung der Stellungnahme	127
Abbildung 7.15: Abschluss der Beanstandung	128
Abbildung 7.16: Prüfung auf Stornierung der Beanstandung	129
Abbildung 7.17: Stornierung der Beanstandung	129
Abbildung 8.1: Erfolgskriterien des weiterentwickelten Reklamationsprozesses	136

## Tabellenverzeichnis

Tabelle 3.1: Vergleich Fehleranalyse IT-Branche mit Automobilbranche .....	17
Tabelle 3.2: Qualitätsaspekte einer Software nach ISO/IEC 25010:2011 .....	18
Tabelle 3.3: Vergleich der Eigenschaften zwischen Hardware und Software .....	19
Tabelle 4.1: Funktionsanalyse Automobilhersteller, Anzahl Funktionen .....	36
Tabelle 4.2: Anzahl der Bewertung .....	39
Tabelle 4.3: Grundsätze des agilen QM.....	64
Tabelle 5.1: Zusammenfassung der Analyseergebnisse .....	73
Tabelle 6.1: Qualitätsmanagement in Start-ups .....	93
Tabelle 7.1: Übersicht der Interviewpartner .....	99

# Abkürzung

Abb.	Abbildung
AG	Aktien Gesellschaft
AGB	Allgemeine Geschäftsbedingungen
App	Application
AR	Augmented Reality
B2B	Business to business
B2C	Business to customer
BMVI	Bundesministerium für Verkehr und digitale Infrastruktur
Bspw.	Beispielweise
BVDS	Bundesverband Deutsche Startups e.V.
BVDS	Bundesverband Deutsche Startups e.V.
bzw.	beziehungsweise
CASE	Connected, Autonomous, Shared, Electric
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
CRM	Customer-Relationship-Management
DevOps	Developer Operations
DGQ	Deutsche Gesellschaft für Qualität
dh.	Das heißt
DIN	Deutsches Institut für Normung
DL	Dienstleister
DMAIC	Define – Measure – Analyse – Improve – Control
dpmo	defects per million opportunities
E/E	Elektrik/Elektronik
E2E	End-to-End
E-Drive	Electric Drive
EN	Europäische Norm
Etc.	et cetera (auf Deutsch: und die übrigen [Dinge])
FMEA	Fehlermöglichkeits- und Einflussanalyse
FTA	Fehlerbaumanalyse
GPS	Global Positioning System
GUI-Test	Graphical user interface Test
HTML	Hyper Text Markup Language
HW	Hardware
i.d.R	In der Regel
IAA	Internationale Automobil-Ausstellung
IaC	Infrastructure as Code
ID	Identifikator
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
iOS	Internetwork Operating System (Betriebs system)

ISO	International Organization of Standardization
J.D. Power	James David Power
KI	künstliche Intelligenz
KIEFA	Karosserie, Innenraum, Elektrik/Elektronik, Fahrwerk und Antrieb
KPI	Leistungsindikator
KPI	Key Performance Indicator
Lft.	Lieferant
LoC	Lines of Code
MAO	Mitarbeiterorientierung
MVP	Minimum Viable Product
NFZ	Nutzfahrzeug
NTF	No Trouble Found
OEM	Original Equipment Manufaktur
OTA	Over - the - air
PDCA	plan - do - check - act
PEP	Produktentwicklungsprozess / Produktentwicklungsphase
PKW	Personal Kraft Wagen
PO	Product Owner
Q	Qualität
QFD	Quality Function Deployment
QM	Qualitätsmanagement
QSK	Fachgebiet Qualitätsstrategie und Qualitätskompetenz, TU Berlin
R&D	Research and development
RGA	Reifegrad Absicherung
RPZ	Risikoprioritätszahl
S.	Siehe
SOP	Start of production
SPC	statistical process control
SPICE	Software Process Improvement and Capability Determination
SRE	Site Reliability Engineering
SW	Software
TQM	Total Quality Management
TU	Technische Universität
TV	television
UMTS	Universal Mobile Telecommunications System
UX	User Experience
VAN	Auf Deutsch: Großraumlimousine
VDA	Verband der Automobilindustrie
Vgl.	vergleich
V-Modell	Vorgehensmodell
XOR	eXclusive OR, exklusives Oder, entweder oder
z.B.	zum Beispiel

# 1 Einleitung

Connectivity, Autonomes Fahren, Sharing und E-Drive sind sowohl neue Herausforderungen als auch künftige Megatrends in der Automobilindustrie. Die OEMs (Original Equipment Manufacturer) investieren seit Jahren viel in diese neuen Themengebiete. Die intelligente Vernetzung von Daten eröffnet fundamental neue Möglichkeiten für die Automobilindustrie. Mit einer digitalen Prozesskette von der Forschung und Entwicklung über die Produktion bis hin zum Vertrieb sowie After-Sales ist die Automobilindustrie bereits in die digitale Ära aufgebrochen.<sup>[1]</sup> Die Welt ändert sich in rasanter Geschwindigkeit; Innovationen und damit die Anforderungen und Erwartungen der Kunden sind zunehmend IT-basiert.

Der Fahrzeugbau wird immer komplexer und im Fahrzeug werden immer mehr Bauteile mit Daten, Software und Apps kombiniert. Trotz hoher Anstrengungen in den Entwicklungs- und Testprozessen zur Übergabe von reifen, robusten und fehlerfreien Produkten an den Kunden treten während der Nutzungsphase beim Kunden Abweichungen vom erwarteten Zustand auf. Somit stellt sich die Frage, wie die Automobilindustrie zukünftig darauf reagieren soll. Innovation, Premiumqualität und Partnerschaft sind seit Jahren wesentliche Erfolgsfaktoren für die Zusammenarbeit zwischen OEMs und Lieferanten. Die Zusammenarbeit der verschiedenen Standorte sollte unter Berücksichtigung der Entwicklung, Produktion, Lokalisierung, Lieferantenbetreuung und Logistik ökonomisch optimiert werden. Hierbei sind nicht nur die Kosten sowie der Zeitaufwand zu berücksichtigen, sondern auch die Nachhaltigkeit der zukünftigen Qualitätsanalysestrategien und die Kundenzufriedenheit.

Mit der Tendenz des steigenden Software-Komponentenanteils im Fahrzeug spielen heute die neuen Player und Start-ups aus der Unterhaltungselektronik und der IT-Branche eine immer bedeutendere Rolle in der Automobilindustrie. Die neuen Unternehmen und Forschungseinrichtungen entwickeln neuartige Werte und Ideen. Immer kürzere Innovationszyklen und der steigende Anteil an Software-Komponenten stellen Automobilhersteller, Zulieferer und Dienstleister permanent vor neue Herausforderungen.<sup>[2]</sup> Der Entwicklungsprozess sowie der Lebenszyklus von digitalen Diensten sind dabei kürzer als der Hardware-Entwicklungsprozess.

Qualität ist der Kern der deutschen OEMs und steht im Mittelpunkt. In diesem Zusammenhang ist noch unklar, wie die Automobilhersteller das Thema Kundenzufriedenheit in Bezug auf zukünftige Services definieren müssen. Damit einhergehend muss auch geprüft werden, inwieweit die aktuelle Qualitätsmanagement-Methodik angepasst werden muss. Qualität bleibt auch in Software-Projekten der relevanteste Treiber für Kostensenkungen und Zeitreduzierung. Die Beseitigung von Fehlern ist in fortgeschrittenen Lebenszyklusphasen, ähnlich wie bei Hardware, mit mehr Zeitaufwand und Kosten verbunden als frühzeitige Fehleridentifikation oder -prävention.<sup>[3]</sup> Das Qualitätsmanagement der OEMs steht nun also vor der Herausforderung, die Qualität von Software sicherzustellen. Doch während sich die Produktstruktur und die Anforderungen der Zusammenarbeit in der Entwicklung

geändert haben, bleibt das Qualitätsmanagement noch immer das traditionelle nach der International Organization of Standardization (ISO) 9000.

Vor diesem Hintergrund ist zu klären, ob das etablierte Qualitätsmanagement bei den OEMs in seiner jetzigen Form weiterhin optimal die Qualität in den Produktentstehungsprozess einbringen kann oder ob es neuer Modelle von Qualitätssystemen bedarf (siehe Abb. 1.1).

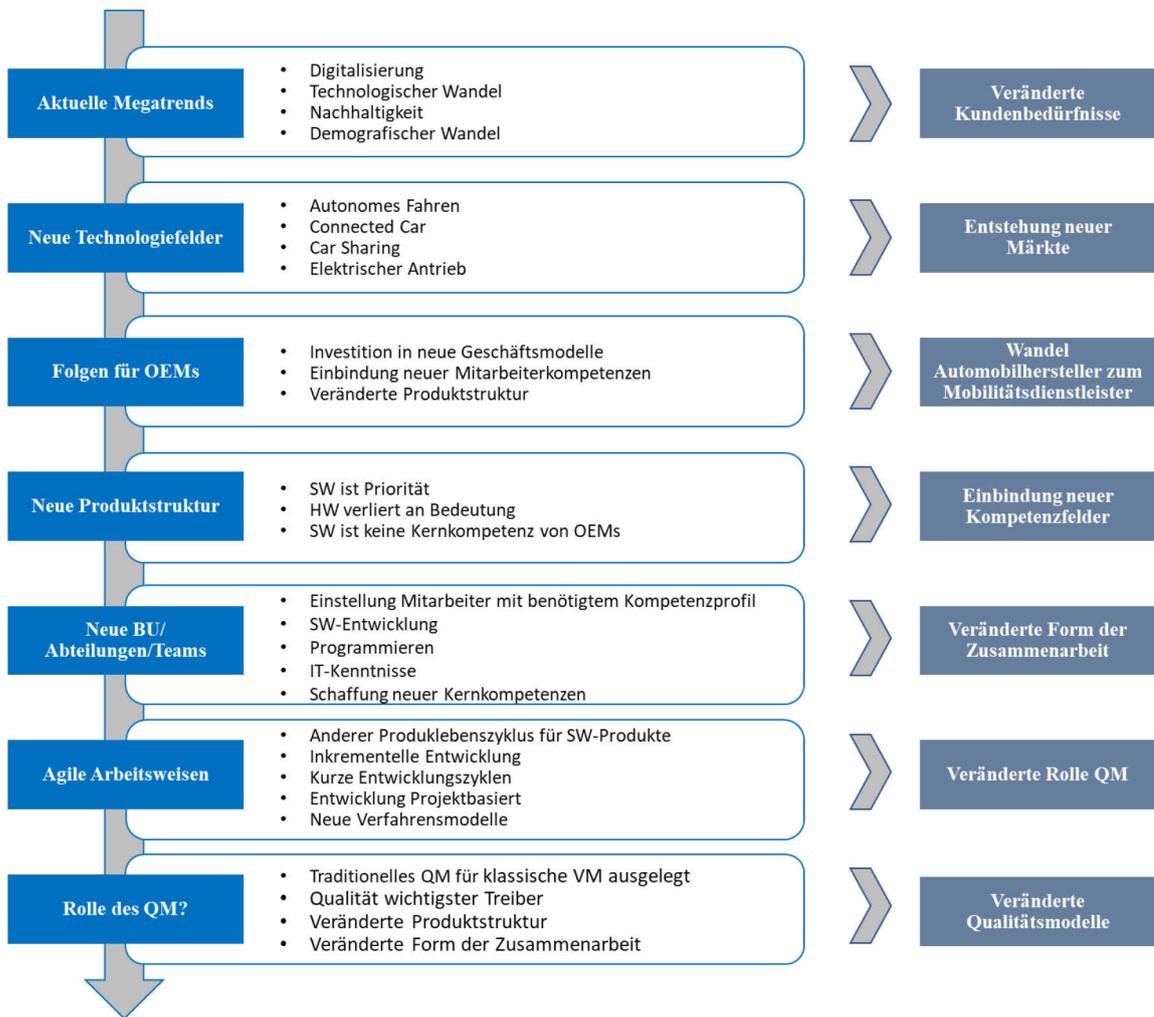


Abbildung 1.1: Allgemeine Darstellung Qualitätsmanagements digitale Dienste in der Automobilindustrie (Quelle: Eigene Darstellung)

Digitalisierung und Konnektivität beeinflussen auch das Rollenverhältnis zwischen OEMs und Zulieferern.<sup>[4]</sup> Die Autohersteller beziehen heute nicht nur Hardware-Komponenten von traditionellen Zulieferern, sondern auch digitale Dienste von globalen IT-Unternehmen sowie Start-ups.

## 2 Motivation, Zielsetzung und Forschungsfragen

Die Automobilindustrie befindet sich in einem dynamischen Wandel, der durch verschiedenste Trends eingeläutet wurde. Diese Innovationen wirken sich essenziell auf die Gestaltung der zukünftigen Strategie der Automobilkonzerne aus: Durch sie werden neue Geschäftsbereiche sowie -modelle entwickelt und alte erfahren teilweise große Veränderungen. Neben technologischen Neuheiten wird die Dynamik auch durch den demografischen Wandel, die Urbanisierung, hohe Wachstumschancen auf den asiatischen Märkten und den Einstieg neuer Wettbewerber aus der IT-Welt mit ihren Geschäftsmodellen in den Automobilsektor vorangetrieben.<sup>[5]</sup> In dieser Arbeit liegt das Hauptaugenmerk auf der Weiterentwicklung von Fehleranalysestrategien digitaler Services in der Automobilindustrie in der Nutzungsphase.

### 2.1 Unternehmenskultur und Geschäftsmodell

Im folgenden Abschnitt werden die bestehende Organisationsstruktur und das traditionelle Geschäftsmodell klassischer Automobilhersteller beleuchtet. Der Auf- und Ausbau von neuen Geschäftsfeldern erfordert nicht nur Investitionen in Form von Forschung und Entwicklung, sondern auch die unterstützenden Prozesse müssen sich entwickeln und andere Formen annehmen.

Der Aufbau eines Automobils weckt zunächst Assoziationen mit Motoren, Karosserie, Achsen und anderen Hardware-Bauteilen. Traditionell ist die Automobilindustrie auch eng mit der Produktion von Komponenten mittels Maschinen verbunden. Gemeinsam ist diesen Automobilteilen, dass ihre Hauptfunktion ebenfalls hardwarebasiert ist und sie dadurch ihren Kundennutzen entfalten. Die Produkte der traditionellen Automobilhersteller werden zunehmend von digitalen Produkteigenschaften geprägt. Die Digitalisierung der Automobilindustrie erfolgt heute entlang der klassischen Kernbereiche, der modularen KIEFA-Produkt- oder Modulstruktur. Die KIEFA-Struktur setzt sich aus den Komponenten Karosserie, Innenraum, Elektrik/Elektronik, Fahrwerk und Antrieb zusammen. Die digitalen Dienstleistungen im Fahrzeug und die entsprechenden mobilen Software-Anwendungen sowie der neuartige Verbindungskanal zum Kunden werden in naher Zukunft einen wesentlichen Erfolgsfaktor für die Automobilunternehmen darstellen. Demnach muss der Fortschritt über die Elektrifizierung des Antriebsstrangs oder die Digitalisierung entlang der KIEFA-Strukturen hinausgehen.<sup>[6]</sup>

Die Vernetzung mit und die Dienstleistung von disruptiven Technologien stellt die gesamte Branche vor neue Herausforderungen und fordert Veränderungen beim bewährten Geschäftsmodell. Die zentrale Strategie für digitale Dienstleistungen im Fahrzeug und die dazugehörigen Geschäftsmodelle müssen auf die digitale Expertise der Wertschöpfungsstufen aus der IT-Branche angepasst werden.

Der ehemalige Vorstandsvorsitzende der Daimler AG Dieter Zetsche sprach in diesem Zusammenhang von einer „Kultur der Offenheit“ und meinte damit: „Um all das

erfolgreich voranzutreiben, erfinden wir auch unsere Unternehmenskultur neu“. Neue Geschäftsmodelle bedeuten zum einen neue Kunden, auf deren Anforderungen möglichst schnell und flexibel eingegangen werden sollte, und zum anderen die Einbindung von neuen Kompetenzen in Form von Mitarbeitern oder Kooperationen.<sup>[7]</sup>

Im Jahr 2015 wurde die neue Struktur in der deutschen Automobilindustrie neben KIEFA als Connectivity oder Software-Bereich benannt. Wie z. B. bei Daimler als CASE (Connected, Autonomous, Shared, Electric), wurde von Volkswagen im Jahr 2019 der neue Bereich als „Digital Car and Services“ bezeichnet.<sup>[8]</sup>



Abbildung 2.1: The five core areas of the new software organization (Quelle: Volkswagen <sup>[8]</sup>)

## 2.2 Tendenzen und Hypothesen in der Automobilindustrie

Diese Trends wurden bereits von einer Vielzahl großer deutscher Automobilhersteller und Konzerne identifiziert und als Strategie formuliert.

Den Wandel hat auch Dieter Zetsche erkannt und weiß um die Chancen und Risiken, die er birgt. „Um die Zukunft der Mobilität von der Spitze aus zu gestalten, haben wir im besten Jahr unserer Firmengeschichte den größten Wandel angestoßen. Denn die Automobilindustrie steht vor fundamentalen Umbrüchen. Connectivity, autonomes Fahren, Sharing und Elektromobilität – jedes dieser Themen hat das Potential, unsere Branche auf den Kopf zu stellen...“<sup>[9]</sup> **CASE:** Autonomes Fahren, Vernetztes Fahrzeug, Elektrischer Antrieb, Sharing

### 1) Autonomes Fahren

Der Wagen meistert im selbstfahrenden Modus Situationen eigenständig, also ohne einen aktiv beteiligten Fahrer. Dadurch ergeben sich Möglichkeiten hinsichtlich fließender Verkehrsströme, flexibler Logistikprozesse und komfortabler Fahrerlebnisse. Vorstufen eines komplett autonom fahrenden Autos sind selbstständiges Abstandhalten zu vorausfahrenden Fahrzeugen, Unterstützung bei Spurwechseln sowie das

automatische Ausweichen und Bremsen. Ein Beispiel einer weiteren Innovation ist das automatische Einparken per Smartphone.<sup>[9]</sup>

## 2) Connected Car

Das vernetzte Fahrzeug unterstützt den Fahrer mit lernfähigen Systemen und Connect-Diensten, die mit der Umgebung kommunizieren. Zudem bietet es den Zugang zu verschiedenen Services, den Abruf von Fahrzeuginformationen per App oder die Warentransportorganisation. Konkrete Formen der Vernetzung äußern sich bspw. in einem Austausch zwischen den einzelnen Fahrzeugen bei Stau oder Glätte. Weiterhin ermöglicht die Vernetzung Funktionen, die bei der Parkplatzsuche unterstützen. Der Fahrer ist über sein Smartphone ständig mit dem Auto verbunden und kann somit Funktionen wie ferngesteuertes Parken, die Verwendung des Smartphones als Fahrzeugschlüssel oder die Fernabfrage diverser Fahrzeuginformationen nutzen. In gewerblicher Hinsicht sind vernetzte Transportsysteme, wie etwa im Verbund fahrende Lkw-Flotten, interessant. Aber auch an vollautomatisierten Laderaumkonzepten und Lieferdrohnen wird gearbeitet.<sup>[9]</sup>

## 3) Electric Mobility

Die Luft in den Städten wird durch das hohe Verkehrsaufkommen belastet, was gesundheitliche Schäden für Mensch und Natur in der direkten Umgebung hervorruft.<sup>[10]</sup> Auch politisch wurden in den letzten Jahren immer engere Grenzen für Emissionen gesetzt.<sup>[11]</sup>

Damit ergibt sich für die Automobilkonzerne dringender Handlungsbedarf und es werden neue Technologien erschlossen, um diesen Entwicklungen entgegenzuwirken. Durch die Elektrifizierung von Antrieben ist ein emissionsfreies Fahren innerhalb und außerhalb von Städten möglich. Erste Pkw-Serienmodelle gibt es bereits seit längerer Zeit auf dem Markt und immer mehr Anbieter bringen Modelle heraus, weil sie deren aktuelles und zukünftiges Potenzial erkannt haben. An der Elektrifizierung von gewerblich genutzten Fahrzeugen wie Trucks, Vans und Bussen wird ebenfalls mit Hochdruck gearbeitet.<sup>[9]</sup>

Mit den elektrischen Antrieben wachsen allerdings auch die Investitionen in Forschungs- und Entwicklungsfelder wie Batterie-Servicesysteme, stationäre Energiespeicher, Ladestationen und Recyclingmöglichkeiten. Neben der Elektrifizierung wird auch an Hybridkonzepten sowie Brennstoffzellen gearbeitet.<sup>[9]</sup>

## 4) Mobilitätsdienstleistungen – Sharing and Service

Eine Fülle an Mobilitätsdienstleistungen ergänzt das Angebot der zukünftigen Mobilität. Applikationen wie Car Sharing, Routenfinder mit zusätzlicher Buchungsoption des jeweiligen Verkehrsmittels bieten den Anwendern eine Vielzahl von Möglichkeiten, ohne dass sie selbst im Besitz eines Autos sein müssen. Die Option, Fahrzeuge flexibel und bei Bedarf für eine kurz- oder mittelfristige Dauer zu mieten, ist seit Jahren unter Car Sharing bekannt.<sup>[9]</sup> Viele der Automobilhersteller, aber auch Autovermietungen, wissen bereits um das Potenzial, das dieser Markt aktuell hat und in Zukunft haben wird.<sup>[9]</sup>

Neue Technologien und entsprechende Produkte verlangen ein angepasstes Mitarbeiterprofil. Wo seit jeher traditionelle Ingenieursberufe wie Maschinenbau und Elektrotechnik in der Forschung und Entwicklung gefragt waren, werden nun Kompetenzen wie Programmieren und Software-Entwicklung immer relevanter. Damit geht die Forderung nach neuen Formen der Zusammenarbeit einher, die eher an erfolgreiche Unternehmen der IT-Branche erinnern als an die Automobilindustrie. Das Stichwort des agilen Arbeitens ist eine dieser neu etablierten Arbeitsweisen, die in der Software-Branche bereits seit vielen Jahren erfolgreich angewendet werden. Dementsprechend fallen auch flachere Hierarchien und kürzere Entscheidungswege unter den Wandel der Unternehmenskultur und werden von OEMs angestrebt.<sup>[13]</sup> Das Thema wird im Lauf dieser Arbeit erforscht und analysiert.



Abbildung 2.2: Änderung der Wettbewerber und des Geschäftsmodells in der Automobilindustrie<sup>[12]</sup>

Wie in Abbildung 2.3 dargestellt, expandiert die Zahl der mit dem Internet und miteinander verbundenen Neufahrzeuge jährlich um etwa 25 %. Heute sind etwa 17 % der Fahrzeuge weltweit mit dem Internet verbunden. Von Experten wird prognostiziert, dass im Jahr 2020 etwa 22 % aller auf dem Markt befindlichen Fahrzeuge mit dem Internet verbunden sind. In den darauffolgenden zwei Jahren steigt die Zahl der mit dem Internet verbundenen Fahrzeuge um 83 Millionen Einheiten auf insgesamt 290,4 Millionen Fahrzeuge an.<sup>[14]</sup>

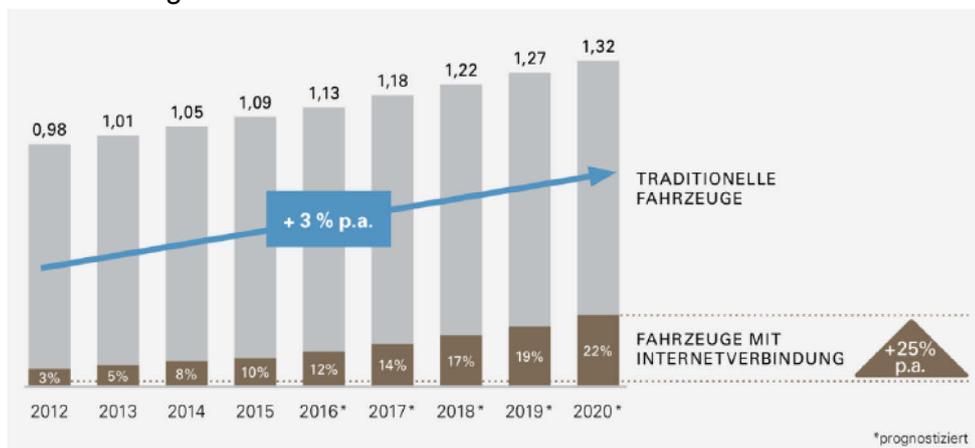


Abbildung 2.3: Fahrzeuge mit Internetverbindung weltweit von 2012–2020 in Milliarden<sup>[14]</sup>

Das exponentielle Wachstum sowie die absolute Zahl von heute 207,4 Millionen Einheiten mit dem Internet verbundener Fahrzeuge sind Indikatoren für die Bedeutung von digital Services in der Automobilindustrie. Aufgrund seines exponentiellen Wachstums gewinnt das Digitalgeschäft jährlich an wirtschaftlicher Bedeutung. Intelligente Nachrüstlösungen für Gebrauchtfahrzeuge können die Wachstumskurven und die Anzahl der mit dem Internet verbundenen Fahrzeuge weiter steigen lassen. Ebenso wird der Ausbau von digitalen Dienstleistungen gefördert. Die Ausbreitung der digitalen Geschäftsmodelle in den After-Sale erschließt weitere Einnahmequellen für den Gesamtumsatz.<sup>[14]</sup>

Laut Statistik werden bis zum Jahr 2020 ca. 98 % der neu zugelassenen Fahrzeuge über verschiedene Plattformen mit Nutzern vernetzt sein, die in das On-Board-System im Fahrzeug integriert sind.<sup>[15]</sup> Dies ist eine kostengünstige Alternative zum eingebetteten System. Zudem bietet es dem Fahrer ein bekanntes Ökosystem und erleichtert so die Bedienung.

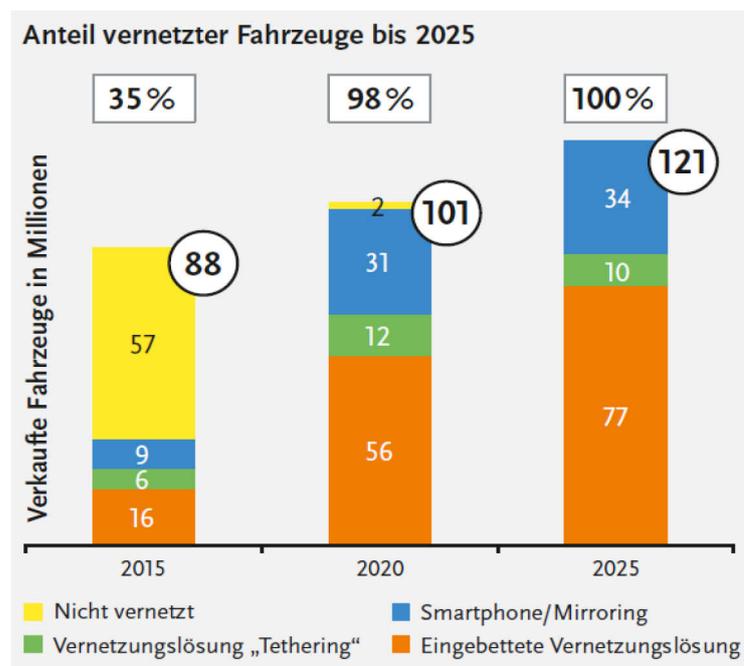


Abbildung 2.4: Anteil vernetzter Fahrzeuge bis 2025 <sup>[15]</sup>

Das traditionelle Geschäftsmodell und dessen Wertschöpfungskette stehen vor einem Umbruch mit tiefgreifenden Veränderungen: Diese Veränderungen beziehen sich auf die Dauer der Entwicklungszyklen, die Produkte, die Kundenbeziehungen, die Vertriebsstrukturen und die Technologien bis hin zum Angebot digitaler Dienstleistungen. Auch der Kundenkontakt ändert sich im Zuge der Digitalisierung: Der Kunde steht mit der eingeforderten digitalen Dienstleistung im Zentrum anstatt mit dem ursprünglichen Produkt – dem Fahrzeug.

Nach einer Befragung von Capgemini im Jahr 2017 mit über 8.000 Teilnehmern aus acht Ländern sind 57 % der befragten Personen bereit, in Zukunft ein Auto von einem technischen Rookie zu erwerben, wie z. B. Apple oder Google. Der Anteil in China

beträgt bereits mehr als 75 %. Über 30 % der Teilnehmer, die heute keinen Connectivity-Service im Auto haben, möchten diese Funktionen für den nächsten Wagen nutzen.<sup>[16]</sup> Die Kundenerwartungen ändern sich heute und werden sich bis zum Jahr 2022 noch schneller ändern. Das stellt eine große Herausforderung für die klassischen Automobilhersteller dar.

## 2.3 Chancen und Herausforderungen

Infolge der Einführung von digitalen Dienstleistungen und den damit verbundenen digitalen Herausforderungen steht die gesamte Automobilindustrie im Umbruch. Die Erweiterung des Geschäftsmodells bietet den Automobilherstellern zahlreiche neue Chancen und Möglichkeiten.

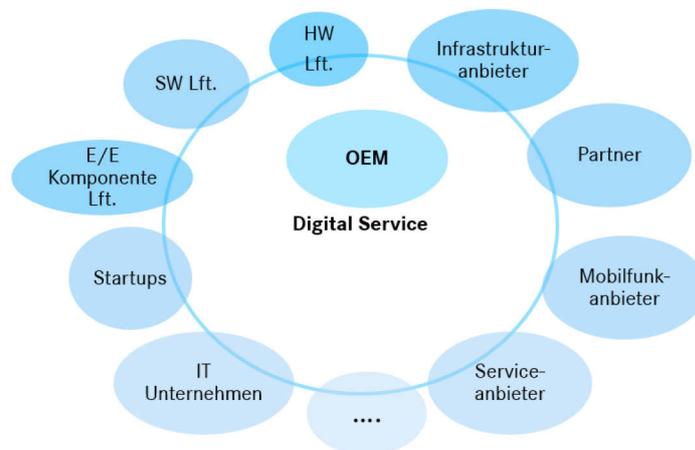


Abbildung 2.5: Zukünftiges Zusammenarbeitsmodell im Feld Connectivity (eigene Darstellung)

Die Automobilhersteller konkurrieren im Wettbewerbskampf um digitale Produkte mit neuen Konkurrenten aus der IT-Branche. Die neuen Wettbewerber aus Silicon Valley sind Rookies, Unternehmen wie Apple und Google. Die Erweiterung der etablierten Produktpalette um digitale Dienstleistungen erfordert eine Verknüpfung der traditionellen und digitalen Wertschöpfungsstufen. Entwicklungszyklen und Innovationsfrequenzen müssen infolge der Zusammenführung hinsichtlich ihrer Synchronität überprüft werden. Die Erweiterung um digitale Kompetenzen, Wertschöpfungsstufen und Geschäftsprozesse erfordert eine Handlungsempfehlung zur Absicherung und Steigerung der Kundenzufriedenheit.

Neue Anforderungen an die Qualitätsanalyse vor dem Hintergrund zunehmender Digitalisierung in der Automobilindustrie sind die folgenden:

- 1) Neue Geschäftsmodelle sowie neue Player im Feld Connectivity
- 2) Erkennen der Wandlung der Kundenerwartungen und der Kundenzufriedenheit
- 3) Aufbau von neuem Know-how entsprechend der Digitalisierungstendenz in anderen Fachbereichen

- 4) Vernetzung der Informationen aus verschiedenen Wertschöpfungsphasen im Fahrzeugentstehungsprozess und gesamten Produkt-Life-Cycles (transparenter & schneller Datenaustausch und Data Mining für verschiedene Bereiche) für Fehleranalysen
- 5) Adressieren der notwendigen Einkaufsrahmenbedingungen sowie Vertragsgrundlagen der Qualitätsanalyse entsprechend der Digitalisierungstendenz
- 6) Vernetzung der Informationen mit internen sowie externen Partnern

Der seit Jahren durchgeführte Qualitätsanalyseprozess in der Automobilindustrie sollte unter Berücksichtigung der wachsenden Komplexität, Digitalisierung und Globalisierung entwickelt werden. Digitale Dienste in der Automobilindustrie umfassen viele IT-übergreifende Themen.

Die Automobilhersteller arbeiten heute eng mit globalen IT-Unternehmen und Start-ups zusammen, um neue Produkte und Services im Markt zu etablieren. Die erfolgreichen Start-ups sowie IT-Unternehmen liefern bereits Produkte und Ideen an Automobilunternehmen. So kaufte Daimler bspw. das Start-up Mytaxi und kooperiert im Jahr 2019 mit BMW unter dem neuen Namen Free Now mit Hauptsitz in Berlin. Daimler, Audi und BMW arbeiten in China mit Alibaba im Feld Artificial Intelligence für digitale Services zusammen.<sup>[17]</sup> Neue Geschäftsmodelle, verschiedene Kooperationspartner und agile Entwicklungsprozesse existieren heute schon in der Automobilindustrie.

In der IT-Branche ist es essenziell, schnell neue Features zu entwickeln.<sup>[18]</sup> Es ist die Aufgabe eines guten Ingenieurs, sich zu überlegen, was vom zukünftigen Qualitätsprozess in den agilen Entwicklungsprozess der digitalen Dienste integriert werden muss.

## 2.4 Forschungsfragen der Dissertation

Das übergeordnete Ziel dieser Arbeit ist die wissenschaftliche Forschung und Entwicklung der Fehleranalysestrategien für digitale Dienste in der Automobilindustrie. Die grundlegenden Forschungsfragen lauten somit:

1. Wie beeinflusst und verändert Digitalisierung die Kundenerwartung?
2. Was sind die Konflikte und Herausforderungen der agilen Produktentwicklung für Qualität?
3. Wie kann die Rolle des Qualitätsmanagements weiterentwickelt und in Zukunft in agile Vorgehensmodelle für digitale Dienste integriert werden?
4. Wie sollte der Fehleranalyseprozess der Automobilindustrie zukünftig mit Blick auf die Entwicklung und Betrieb angepasst werden?
5. Worin bestehen die Herausforderungen der Kultur und Organisation für das zukünftige Connectivity-Feld in der Automobilindustrie?
6. Gilt „Quality first“ auch für die Kooperation von Automobilherstellern mit IT-Firmen und Start-ups?

## **2.5 Aufbau der Dissertation**

Digitale Dienste in der Automobilindustrie sind eine Kombination von klassischen Automobilherstellern und IT-Firmen. Im Verlauf dieser Dissertation werden stets die Bereiche Automobil und IT berücksichtigt bzw. verglichen.

Die vorliegende Dissertation unterteilt sich in zehn Kapitel. Die ersten zwei Kapitel umfassen das Motivationsschreiben, die Trendanalyse, die Problemstellung, die Forschungsfragen sowie die Zielsetzung dieser Arbeit. Im Anschluss an diese Punkte wird die Situation in der Automobilindustrie beschrieben.

In Kapitel 3 werden die Grundlagen sowie relevanten Theorien dieser Dissertation vorgestellt. Der theoretische Teil beleuchtet die Grundlagen von Qualitätsmanagement, Software und digitalen Diensten sowie Standards des Fehlermanagements.

In Kapitel 4 zum Stand der Forschung und zur Gap-Analyse werden die Kernthemen der Arbeit analysiert und erforscht.

Kapitel 5 bis 7 umfassen den praktischen Teil. Darin werden die Forschungsfragen analysiert und durch Literaturrecherche, Industrie-Interviews, Umfragen und praktische Projektarbeit beantwortet. Anschließend werden eigene Konzepte sowie die Ergebnisse der Dissertation vorgestellt.

In Kapitel 8 erfolgt eine Zusammenfassung sowie der Ausblick dieser Arbeit.

### 3 Grundlagen und Stand der Technik

Dieses Kapitel beschreibt zunächst die Grundlagen des Qualitätsmanagements sowie der Fehleranalyse in den Automobil- und IT-Branchen und vergleicht sowohl diese beiden Branchen als auch die jeweiligen Fehleranalysemethoden miteinander. Darüber hinaus wird ein Überblick über den Begriff der digitalen Dienste gegeben.

#### 3.1 Grundlagen Qualitätsmanagement

Gemäß EN ISO 9000 ff. ist Qualität der Grad, in dem ein Satz inhärenter Merkmale eines Objekts Anforderungen erfüllt (S. Abb. 3.1). Management indessen beschreibt aufeinander abgestimmte Tätigkeiten zum Führen und Steuern einer Organisation. Daraus lässt sich ableiten, dass Qualitätsmanagement abgestimmte Tätigkeiten zum Führen einer Organisation umfasst, die das Ziel haben, die Anforderungen an ein Objekt mit dem höchstmöglichen Grad zu erfüllen. Dabei kann ein Objekt sowohl als Produkt als auch als Prozess verstanden werden.<sup>[19]</sup>

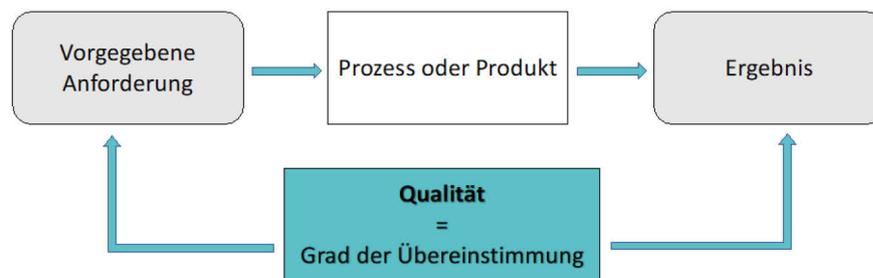


Abbildung 3.1: Anforderung, Ergebnis, Qualität (Eigene Darstellung, interne Forschungsstudie Fachgebiet Qualitätsstrategien und -kompetenz an der TU Berlin)<sup>[20]</sup>

Die zentralen Aufgaben des Qualitätsmanagements umfassen die Qualitätsplanung, die Qualitätslenkung, die Qualitätssicherung sowie die Qualitätsverbesserung. Im Zuge der Qualitätsplanung werden die Qualitätsziele sowie die Prozesse und Ressourcen festgelegt, die zum Erreichen der entsprechenden Ziele benötigt werden. Die Qualitätslenkung hat das Ziel, die Qualitätsanforderungen zu erfüllen. Die Qualitätsverbesserung befasst sich mit Verbesserungsmöglichkeiten in Bezug auf die Wirksamkeit, Effizienz und Rückverfolgbarkeit. Die Qualitätssicherung umfasst Tätigkeiten zur Erzeugung von Vertrauen darin, dass Qualitätsanforderungen erfüllt werden, und umfasst drei Zuständigkeiten: Messen und Prüfen, Fehlermanagement und Anforderungsmanagement.<sup>[21]</sup>

In vielen Produktgruppen, wie bspw. der Unterhaltungselektronik oder Smartphones, sind traditionelle Qualitätsmerkmale wie eine möglichst lange Lebensdauer und Robustheit keine zentralen Anforderungen mehr an das Produkt. Aufgrund der kurzen Innovations- und damit auch Produktlebenszyklen sowie durch ein überzeugendes Marketing werden die Konsumenten dazu angehalten, Produkte in immer schnelleren

Rhythmen zu ersetzen. Vor allem das deutsche Verhältnis zu Qualität entspricht eher dem traditionellen Verständnis, was sicherlich im Hinblick auf Ressourcenverbrauch und Umweltbelastung positiv zu bewerten ist. In den Industriebereichen, in denen besonders schnelle Innovationszyklen herrschen, sind deutsche Unternehmen jedoch eher weniger vertreten.

### Die Historie des Qualitätsmanagements <sup>[22]</sup>

Bis zum Jahr 1900 blieben einzelne Fertigungsschritte in der industriellen Produktion im Verantwortungsbereich eines einzelnen Arbeiters, der simultan die Qualität seiner Arbeit prüfte. Von 1900 bis 1950 veränderte die steigende Nachfrage nach Gütern die Produktionsstrategie nachhaltig. Fertigungsvorgänge wurden entlang der Fließbandfertigung in einzelne Arbeitsschritte zerlegt, die von gering qualifiziertem Personal ausgeführt wurden. Diese Arbeitsorganisation wird als Funktionsmeisterprinzip bezeichnet. Die geringe Qualifikation der Arbeiter hatte zur Folge, dass die Qualitätskontrolle nicht länger vom Arbeiter selbst, sondern stattdessen von Qualitätsprüfabteilungen verantwortet wurden.

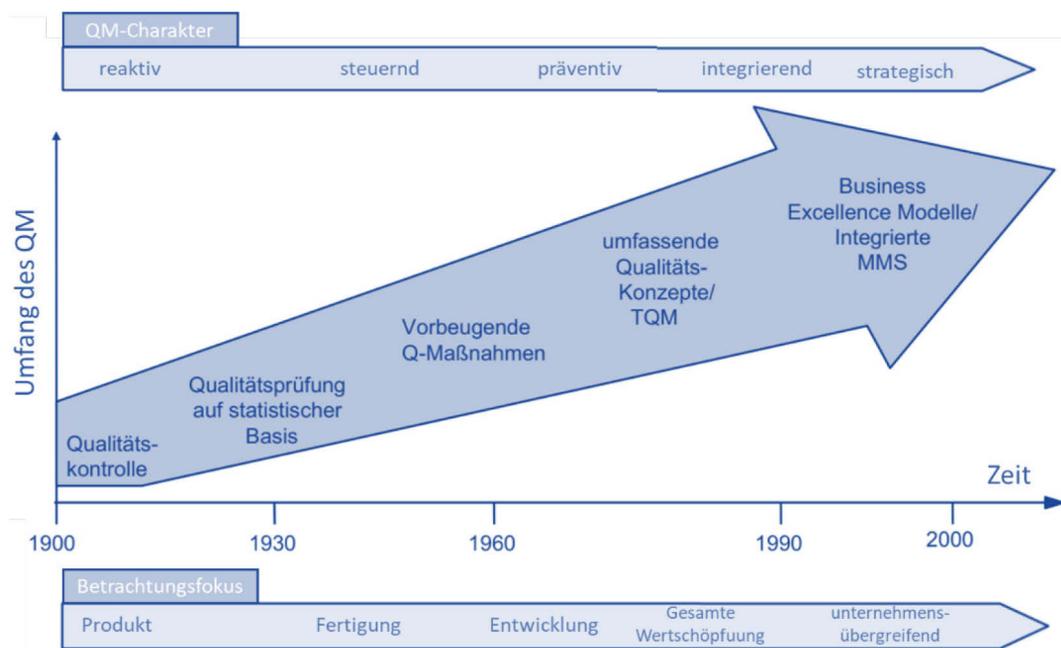


Abbildung 3.2: Entwicklungsstufen des Qualitätsmanagements (Quelle: Eigene Darstellung in Anlehnung an Brüggemann/Bremer 2015<sup>[22]</sup>)

1924: Methoden zur kontinuierlichen Prozessbeobachtung und -bewertung auf statistischer Basis werden entwickelt.

1930: Mit der aufkommenden Massenproduktion wurde die vollständige Kontrolle in der Produktion zu aufwändig und deshalb von der Teilkontrolle auf Basis statistischer Verfahren abgelöst.

Ab 1960: Die zunehmende Komplexität der Produkte verlangte eine intensivierete Integration des Qualitätsmanagements (QM) in die Produktentwicklung. Fehler sollten

nicht erst nach Auftreten beseitigt, sondern vor der Entstehung erkannt und verhindert werden. Infolgedessen verloren reine Kontrollmaßnahmen an Bedeutung.

Ab 1990: Japanische Qualitätsüberlegungen wie Kaizen und das Toyota-Produktionssystem prägten zunehmend die westlichen Industrien. Die Konzepte wurden im Total Quality Management umgesetzt, was die kontinuierliche Qualitätsverbesserung zur Aufgabe und Ziel der Geschäftsleitung des Unternehmens machte. Qualität wurde somit ein strategisches Unternehmensziel.

Mitte 1990: Die meisten deutschen Industrieunternehmen führten die weltweit verbreitete QM-Norm DIN EN ISO 9001 ein. Weitere Normen hinsichtlich Umweltstandards, Arbeitssicherheit u.a. wurden entwickelt und folglich das System der Qualitätsnormen weiter ausgebaut. Unternehmen führten zwecks Vergleichbarkeit der Strukturen von Managementsystemen das integrierte Managementsystem ein, das die Qualitätsnormen zusammenfasst. Unternehmenspreise (Business Excellence Awards) wurden eingeführt, die auf Basis von Bewertungsmodellen (Business Excellence Models) an exzellente Unternehmen vergeben werden.<sup>[22]</sup>

## 3.2 Grundlagen Software

Unter Software wird ein immaterielles Produkt verstanden, das seine Funktion primär durch den Programmcode erhält. Diese Programme lassen sich auf Hardware implementieren und bilden zusammen ein Computersystem, das dem Nutzer verschiedenste Lösungsmöglichkeiten bietet. Im Zusammenhang mit Software wird auch von der Software-Entwicklung gesprochen, die sich von der Anforderungsdefinition bis hin zur Implementierung des Quellcodes in die Hardware-Struktur erstreckt. Die Software-Dokumentation umfasst Anforderungsspezifikationen, Entwurfsdokumente, den Programmcode, die Testdokumentation und Wartungsinstruktionen und ist somit ein bedeutender Teil der Software-Erstellung.

Nach Hohler<sup>[23]</sup> lassen sich folgende Begriffsdefinitionen aus der Literatur ableiten:

Eine Software ist ein geistiges Produkt, das aus Programmen, Verfahren und allen dazugehörigen Beschreibungen besteht, die zur Arbeit mit einem Datenverarbeitungssystem gehören; Software ist unabhängig von dem Medium, auf dem sie gespeichert ist.

Die aktuelle ISO/IEC-Norm 24765 ersetzt die DIN-Norm 44300 und enthält für Software folgende Definitionen:<sup>[24]</sup>

- Software ist ein Programm oder eine Menge von Programmen, die dazu dienen, einen Computer zu betreiben.
- Software sind Programme sowie die zugehörige Dokumentation.
- Software sind Programme und ggf. die zugehörige Dokumentation und weitere Daten, die zum Betrieb eines Computers notwendig sind.

### 3.2.1 Entstehung sowie Lebenszyklus eines Software-Produkts

Um der Entwicklung von Software-Projekten eine Struktur zu geben, lassen sich Software-Prozessmodelle oder -Vorgehensmodelle anwenden. Diese beinhalten ein einheitliches Rahmenwerk, auf dessen Grundlage sich der Ablauf von Projekten planen und umsetzen lässt. Vorgehensmodelle sind eng mit dem Konzept des Lebenszyklus verknüpft. Dieses Konzept lässt sich analog zu physischen Produkten auch auf Software-Produkte anwenden.

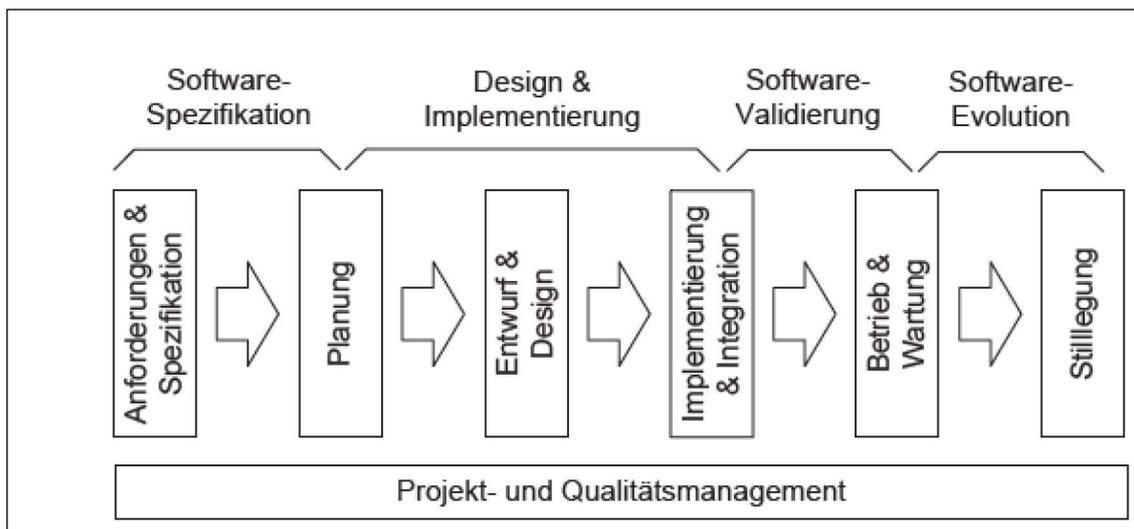


Abbildung 3.3: Lebenszyklus eines Software-Produkts <sup>[25]</sup>

Der Lebenszyklus lässt sich grob in vier aufeinanderfolgende Phasen einteilen, die wiederum in feinere Phasen untergliedert werden. Das Projekt- und Qualitätsmanagement unterstützt ein Software-Projekt phasenübergreifend. Hierfür existieren softwarespezifische Qualitätskriterien. An dieser Stelle ist anzumerken, dass konkrete Qualitätsmaßnahmen projektspezifisch geplant werden sollten. Dazu gehört auch die adäquate Auswahl des Vorgehensmodells. Grundsätzlich lassen sich projektspezifische Qualitätsmaßnahmen in agilen Vorgehensmodellen (s. Kap. 4.7.1) umsetzen. Das liegt einerseits am iterativen Charakter, der die umgesetzten Anforderungen nach jeder Entwicklungsphase überprüfbar macht. Andererseits bieten viele agile Vorgehensmodelle genügend Freiheitsgrade, um projektspezifische Qualitätsaktivitäten innerhalb des gewählten Vorgehens zu integrieren.<sup>[26]</sup>

- **Anforderungen & Spezifikationen:** In dieser Phase erfolgen die Analyse und Spezifikation der vom Kunden gewünschten Anforderungen.
- **Planung:** Das Software-Projekt wird initial geplant und durch Methoden aus dem Projektmanagement gesteuert sowie überprüft. Es werden dabei projektspezifische Charakteristika, bspw. agiles oder sequenzielles Vorgehen, innerhalb der Planung berücksichtigt.
- **Entwurf & Design:** Während dieser Phase erfolgt die Auswahl geeigneter Komponenten, die den Aufbau und die Struktur der zu entwickelnden Software systematisieren.

- Implementierung & Integration: Es folgt die Umsetzung der definierten Anforderungen. Die programmierten Komponenten werden anschließend zusammengefügt bzw. integriert.
- Betrieb & Wartung: In dieser Phase werden Fehler in der Software korrigiert. Darüber hinaus kann die Software durch neue Funktionalitäten erweitert oder an neue Systemumgebungen angepasst werden.
- Stilllegung: Die Software wird planmäßig stillgelegt oder durch eine neue Software ersetzt.

Für die Software-Erstellung entfällt die Produktion, da Software beliebig vervielfältigt werden kann und oftmals per Download vom Kunden beziehbar ist. Auch ein Datenträger, der die Software speichert, gilt als verschwindend geringer Aufwand und wird nicht der Produktion zugewiesen.

Der Software-Lebenszyklus beschreibt den Lebenszyklus eines Software-Systems, das aus den Entwicklungsphasen, dem Betrieb und der Außerbetriebnahme besteht.<sup>[27]</sup>

In großen Unternehmen mit umfangreichen Entwicklungsbereichen werden die verschiedenen Projekte auch durch eine Vielzahl unterschiedlicher Vorgehensmodelle abgedeckt und behandelt. Je nach Anforderung, Dauer, Komplexität, Größe und Bereich der Software-Projekte eignet sich ein spezifisches Vorgehensmodell. Somit finden sich unterschiedliche Vorgehensmodelle in einer Organisation wieder. Die Vorgehensmodelle lassen sich in drei Kategorien einteilen, die sich in ihrem Ablauf stark differenzieren.<sup>[23]</sup>

### **1) Klassische, sequenzielle Vorgehensmodelle**

Einzelne Phasen werden behandelt und abgeschlossen, bevor das Ergebnis an die nächste Phase weitergegeben wird. Die Phasen werden strikt sequenziell durchlaufen und beinhalten zum Ende jeder Phase einen Meilenstein, der die Arbeitsergebnisse definiert und prüft. Rückkopplungen zwischen einzelnen Phasen sind bedingt möglich. Diese Modelle vernachlässigen Wechselwirkungen und Überschneidungen zwischen einzelnen Phasen und stellen den Entwicklungsprozess in stark idealisierter Form dar. Außerdem sind die Modelle nicht kundenfreundlich, da das Einfließen von Kundenfeedback nur zu Beginn des Entwicklungsprozesses vorgesehen ist. Die beiden bekanntesten Beispiele der sequenziellen Modelle sind das Wasserfall- und das V-Modell.

### **2) Klassische, iterative und inkrementelle Vorgehensmodelle**

Das Ziel der iterativen Ansätze ist es, den Kunden möglichst schnell einen Produkt-Prototypen zu liefern. Hierdurch wird der Kunde nah am Entwicklungsprozess gehalten. Durch die Berücksichtigung von Änderungswünschen können Missverständnisse früh erkannt oder vermieden werden. Nachteile der Modelle sind ihre komplexe, unflexible Struktur und der hohe Dokumentationsaufwand, die sie schwerfällig machen. Ein Beispiel für klassisch-inkrementelle Vorgehensmodelle ist das Spiralmodell.

### 3) Agile Vorgehensmodelle

In der Software-Entwicklung bedarf es der Möglichkeit, sich verändernden Anforderungen anzupassen, Kundenanforderungen kontinuierlich einfließen zu lassen und dabei Qualität und Sicherheit der Produkte sicherzustellen. Für diese schnellen Anpassungen sind dokumentationslastige Vorgehensmodelle nicht geeignet. Aus diesem Umstand heraus entwickelten sich, angestoßen durch die agilen Methoden. (s. Kapitel 4.8)

#### 3.2.2 Automotive SPICE

SPICE (Software Process Improvement and Capability Determination) beschreibt einen internationalen Standard zur Bewertung von Software-Entwicklungsprozessen und basiert auf der ISO-Norm 15504. SPICE setzt sich aus zwei Dimensionen zusammen: der Prozess- und der Reifegraddimension. Die Prozessdimension unterscheidet drei Prozesskategorien, in denen wiederum bestimmte Arbeitsabläufe zu Prozessgruppen zusammengefasst werden.

Die Prozesskategorie ‚Primary Life Cycle Processes‘ beinhaltet alle Prozesse, die den Kunden und den Lieferanten direkt einbeziehen (Beratung, Akquisition, Betreuung, Belieferung und Support) sowie die Prozesse, die bei der eigentlichen Produkterstellung ablaufen (Softwareentwurf, Implementierung, Test und Wartung). In der Prozesskategorie ‚Organizational Life Cycle Processes‘ finden sich alle Prozesse wieder, die dem Bereich des Managements zuzuordnen sind (bspw. Projektplanung, Fortschrittsüberwachung, Qualitäts- und Risikomanagement sowie Koordination und Überwachung der Zulieferer). Zudem enthält sie alle Arbeitsabläufe, die die Umsetzung der Unternehmensziele unterstützen (bspw. Aufrechterhaltung der Infrastruktur, systematische Bereitstellung von Ressourcen sowie die Definition und Verbesserung von Prozessen). Die letzte Prozesskategorie der obersten Ebene ist der ‚Support‘: Hierunter werden Prozesse zusammengefasst, die andere Prozesse unterstützen (bspw. die Dokumentation, das Konfigurationsmanagement, die Verifikation und Validation sowie die Qualitätssicherung).<sup>[28]</sup>

Die zweite Dimension von SPICE, die Reifegraddimension, dient der Beurteilung von Prozessen in einem Unternehmen. Darin unterscheidet sich das SPICE-Modell von den anderen Reifegradmodellen, denn diese bewerten den Reifegrad der Organisation.<sup>[28]</sup> Die Reifegrade ähneln den Maturity Levels der CMM/CMMI-Modelle, beinhalten jedoch einen zusätzlichen Reifegrad. (CMM: Capability Maturity Model. CMMI: Capability Maturity Model Integration)

Das SPICE-Modell wird nicht nur im Bereich der reinen Software-Entwicklung angewendet, sondern wurde auch in ähnlicher Form für andere Industrien abgewandelt. Für die Automobilindustrie wurde dementsprechend das Automotive-SPICE-Modell entwickelt, das speziell auf die Anforderungen im Bereich der Steuergeräteentwicklung abzielt, im Wesentlichen allerdings dem hier beschriebenen Standardmodell gleicht. Automotive SPICE dient mittlerweile nicht mehr nur als Bewertungsgrundlage für

Prozesse, sondern kann auch als Leitfaden für eine qualitativ hochwertige Software-Entwicklung im Automotive-Bereich angesehen werden.<sup>[29]</sup>

### 3.2.3 Softwaremerkmale

In der Automobilbranche und der IT-Branche handelt es sich um unterschiedliche Industriezweige. Wie in Tabelle 3.1 dargestellt, ist die Automobilbranche die deutlich ältere Branche. Die beiden Branchen besitzen sowohl deutlich verschiedene Längen der Entwicklungszyklen als auch unterschiedliche Bedingungen bzgl. Kundenkontakt und Sicherheit.

Automobilbranche	Mobilfunk- / IT-Branche
Lange Historie (>100 Jahre)	neuere Branche (ca. 15–20 Jahre)
Entwicklungszyklen länger (ca. 5–7 Jahre)	kürzere Entwicklungszyklen (1–2 Jahre)
Historisch kein direkter Kundenkontakt (häufig nur über Werkstätten oder Händler)	enger Kundenkontakt
hohes Sicherheitsbedürfnis	Sicherheitsthemen wichtig, aber nicht so ausgeprägt wie bei Automobilbauern
Methode: Schadteilanalyseprozess nach VDA	Methode: Statische und dynamische Tests
Fokus liegt auf Hardware	Fokus liegt auf Software

Tabelle 3.1: Vergleich Fehleranalyse IT-Branche mit Automobilbranche <sup>[30]</sup>

Software-Produkte sind immaterieller Natur und liefern ihren Nutzen nicht eigenständig, sondern im Verbund innerhalb eines Computersystems. Das verhindert im Gegensatz zu Maschinen und anderen materiellen Produkten eine einfache differenzierte Abschätzung des Nutzens der Software. Dieser Zustand beeinträchtigt die Bereitschaft des Kunden, die Entwicklung von Software entsprechend monetär zu würdigen bzw. den Zeitaufwand plausibel abzuschätzen. Nach dem Entwicklungsaufwand ist die Vervielfältigung von Software aufgrund des immateriellen Charakters verhältnismäßig kostengünstig.

Software ist aus einem komplexen Konstrukt aus Quellcodes aufgebaut. Zwischen den einzelnen Komponenten besteht eine Vielzahl von Verbindungen, Abhängigkeiten und Interdependenzen, die Änderungen im Quellcode hinsichtlich ihrer Folgeauswirkung schwierig kalkulierbar machen. Die Größe von Software-Programmen wird in ‚Lines of Code‘ (LoC) gemessen. Große Programme können mehrere Tausend Codezeilen umfassen, was ihre Komplexität weiter verdeutlicht.

Vorgehensmodelle strukturieren den Entwicklungsprozess und haben darüber hinaus das Ziel, die Komplexität des Entwicklungsprozesses zu reduzieren. Dafür werden Phasen, deren Abläufe und Meilensteine definiert, die sich am Software-Lebenszyklus orientieren. Abhängig von der Abfolge der Phasen wird zwischen sequenziellen und iterativen Vorgehensmodellen unterschieden. Eine Entscheidung für ein bestimmtes Vorgehensmodell sollte projektspezifisch erfolgen.<sup>[31]</sup> So können individuelle Projektgegebenheiten im Entwicklungsprozess berücksichtigt werden. Sind bspw. die Anforderungen an ein Produkt vor Beginn der Entwicklung stabil und eindeutig

definierbar, so ist die Anwendung eines sequenziellen Vorgehensmodells zu empfehlen. Im Bereich Connectivity ist ein solches Szenario nicht zu erwarten; vielmehr ist eine dynamische Anpassung der Software an sich verändernde Kundenerwartungen während der Entwicklung wahrscheinlich. Entsprechend ist die Anwendung agiler Methoden zu empfehlen. An dieser Stelle ist zu erwähnen, dass Vorgehensmodelle generisch ausgestaltet sind und deren Anpassung daher im jeweiligen Unternehmen stattfindet. Die Anpassung ist vom Umfang des betreffenden Projekts abhängig und wird als Tailoring bezeichnet.<sup>[32]</sup>

### 3.2.4 Qualitätsmerkmale von Software

Der Begriff Qualität wurde bereits umfassend definiert und dies lässt sich ebenso auf die Qualität von Software übertragen. So definiert die ISO/IEC 25000 die Software-Qualität wie folgt: „Softwarequalität ist die Gesamtheit der Merkmale und Merkmalswerte eines Software-Produkts, die sich auf dessen Eignung beziehen, festgelegte oder vorausgesetzte Erfordernisse zu erfüllen.“<sup>[24]</sup>

Im Produktqualitätsmodell wird die Software-Qualität eines Produkts anhand folgender acht Merkmale, die in Tabelle 3.2 zusammengestellt sind, beschrieben:

Hauptmerkmal	Unterkmerkmal
Funktionalität – Aspekte, die die angeforderten Funktionen eines Systems beschreiben.	<ul style="list-style-type: none"> <li>- Funktionelle Vollständigkeit</li> <li>- Angemessenheit</li> <li>- Richtigkeit</li> </ul>
Zuverlässigkeit – Erbringen eines vereinbarten Leistungsniveaus unter festgelegten Bedingungen über einen definierten Zeitraum hinweg.	<ul style="list-style-type: none"> <li>- Reife</li> <li>- Fehlertoleranz</li> <li>- Wiederherstellbarkeit</li> </ul>
Effizienz – Test messbarer Ergebnisse zur Erfüllung der Aufgabe.	<ul style="list-style-type: none"> <li>- Zeitverhalten</li> <li>- Ressourcenverbrauch</li> </ul>
Benutzerfreundlichkeit – Berücksichtigung des Aufwands für die Benutzung der SW durch die verschiedenen Benutzergruppen.	<ul style="list-style-type: none"> <li>- Verständlichkeit</li> <li>- Erlernbarkeit</li> <li>- Bedienbarkeit</li> <li>- Ästhetik der Benutzeroberfläche</li> </ul>
Sicherheit – Sicherheit im Umgang mit der SW, keine vertraulichen Daten zu streuen oder andere Systeme zu beeinträchtigen.	<ul style="list-style-type: none"> <li>- Vertraulichkeit</li> <li>- Integrität</li> <li>- Zurechenbarkeit</li> </ul>
Kompatibilität – Zusammenspiel des Systems mit anderen vorgegebenen Systemen.	<ul style="list-style-type: none"> <li>- Interoperabilität</li> <li>- Koexistenz</li> </ul>
Wartungsfreundlichkeit – Wartungsfreundlichkeit, die einen effizienten Einsatz der SW über einen längeren Zeitraum voraussetzt.	<ul style="list-style-type: none"> <li>- Modularität</li> <li>- Analysierbarkeit</li> <li>- Änderbarkeit</li> <li>- Prüfbarkeit</li> </ul>
Übertragbarkeit – Betrieb auf verschiedenen Betriebssystemen und HW-Plattformen.	<ul style="list-style-type: none"> <li>- Installierbarkeit</li> <li>- Anpassbarkeit</li> <li>- Konformität</li> </ul>

Tabelle 3.2: Qualitätsaspekte einer Software nach ISO/IEC 25010:2011<sup>[33]</sup>

(Quelle: Eigene Darstellung in Anlehnung an Hohler 2014; Vogenschow 2010)

Diese Qualitätsmerkmale lassen sich nicht alle gleichzeitig optimieren, sondern sind teils gegensätzlich. Aus diesem Grund ist es relevant, zu priorisieren.<sup>[33] [34]</sup> Somit ist nicht alles technisch Mögliche auch sinnvoll, sondern es zählt vielmehr die Orientierung an Erwartungen, was eine Kommunikation zwischen den verschiedenen Stakeholdern im

Software-Projekt nötig macht. Vogenschow greift die Qualitätsdefinition auf und vereinfacht sie praxisgerecht: „*Qualität ist Leistung im Verhältnis zur Erwartung.*“<sup>[34]</sup>

Die folgende Tabelle liefert einen Vergleich der Eigenschaften zwischen Hardware und Software:

Hardware	Software
Fehler werden durch Mängel in der Entwicklung, Produktion eingeführt. Sie können weiterhin durch Abnutzung verursacht werden	Fehler werden durch Menschen im Verlauf der Softwareerstellung, z.B. in der Spezifikation oder im Entwurf, verursacht
Fehler werden durch zunehmendes Altern verursacht. Oftmals macht sich ein kurz bevorstehender Ausfall vorher bemerkbar.	Fehler treten plötzlich und ohne Vorwarnung auf. Fehler in bestehender Software können durch neue Releases eingeführt werden.
Präventive Wartung zur Vermeidung eines Ausfalls ist möglich.	Präventive Wartung von Software existiert nicht.
Die Zuverlässigkeit ist i.A. eine Funktion der Betriebszeit	Die Zuverlässigkeit von Software hängt nicht von der Betriebszeit ab. Ältere Software kann zuverlässiger sein, als Software mit neuen Releases.
Die Zuverlässigkeit kann mit Hilfe physikalischer/ empirischer Gesetze berechnet werden.	Die Zuverlässigkeit kann durch Wissen über das Design, den Einsatz der Software und der Stressfaktoren nicht exakt berechnet werden.
Die Zuverlässigkeit eines Systems kann durch Redundanz i.d.R gesteigert werden.	Redundanz allein genügt nicht, um die Zuverlässigkeit zu erhöhen. Zwei Kopien des gleichen Programms enthalten die gleichen Fehler.
Schnittstellen sind visuell.	Schnittstellen sind konzeptionell.

Tabelle 3.3: Vergleich der Eigenschaften zwischen Hardware und Software<sup>[40]</sup>

Für Hardware gilt nach Moore's Law die Regel, dass sich die Leistungsfähigkeit von Computer-Hardware-Komponenten bei gleichbleibenden Kosten ungefähr alle ein bis zwei Jahre verdoppelt. Die Kosten sind dagegen bei Software-Produkten schwer einzuschätzen, da es keine allgemeingültigen Aussagen und Untersuchungen gibt, wie sich die Kosten für die Entwicklung hinsichtlich der Leistungsfähigkeit und Größe der Software über die Jahre entwickelt haben. Unstrittig ist jedoch, dass der relative Anteil der Software-Kosten an den Gesamtkosten von eingebetteten Systemen stetig zunimmt.<sup>[27]</sup>

Außerdem umfasst der Großteil der Kosten innerhalb der Software-Herstellung die Personalkosten, die sich aus reiner Arbeitszeit oder Aus- und Weiterbildungs- sowie Reisekosten zusammensetzen. Die Abschätzung der Kosten erscheint am Anfang eines Software-Projekts schwierig, da vielerlei Faktoren gegenseitige Abhängigkeiten aufweisen und nicht mit ausreichend Gewissheit bestimmt werden können.<sup>[27]</sup>

Die Kosten eines Software-Projekts korrelieren mit der Entwicklungsdauer, die wiederum abhängig von der Produktivität der Programmierer ist. Die Produktivität eines Entwicklers kann in LoC gemessen werden, wobei diese Kennzahl allerdings Schwächen aufweist: Zum einen schwankt der durchschnittliche Durchsatz an LoC innerhalb des Entwicklerteams. Somit kann sich nur auf Durchschnittswerte bezogen werden, die bei einem LoC von zehn bis zwanzig Zeilen pro Tag liegen. Außerdem muss neben der verwendeten Programmiersprache auch der Umfang des Software-Projekts

berücksichtigt werden. Des Weiteren lassen Produktivitätsüberlegungen die Qualität außer Acht, was zu zusätzlichen Kosten und Aufwand führen kann.<sup>[23]</sup>

Nach Hohler steigt die Größe eines Programms innerhalb von fünf Jahren um das Zehnfache an. Das bedeutet, dass die Entwicklung von Software-Produkten immer komplexer wird und immer mehr Entwickler bzw. Entwicklerteams daran beteiligt sind. Mit steigender Größe und Entwicklungsaufwand ändern sich somit auch ständig die Anforderungen an die Organisation der Entwicklung und es müssen neue Zusammenarbeits- und Verfahrensformen angewendet werden. Es ergibt sich also eine hohe Dynamik beim Austausch von etablierten und neuen Prozessen.<sup>[23]</sup>

Die Fehleranfälligkeit hängt mit der für Software charakteristischen Struktur zusammen: Software bildet aufgrund des binären Aufbaus und der Informationsverarbeitung ein diskretes, un stetiges System. Die Anzahl der Zustandsmöglichkeiten geht demnach gegen unendlich und lässt keinen Beweis vollständiger Korrektheit zu. So zeigen Untersuchungen, dass Software-Produkte bei der Auslieferung eine Fehlerquote in Höhe von 0,1-0,3 % aufweisen – bei bestehendem Qualitätsmanagement und durchgeführter Testphase.

Eine Anzahl an Zustandsmöglichkeiten gegen unendlich macht eine Anzahl an Tests in der gleichen Summe notwendig. Damit ist ein vollständiger Test nicht möglich. Das Toleranzprinzip veranschaulicht, wie schwierig Software im Vergleich zu Hardware zu überprüfen ist: Wohingegen für Hardware Toleranzbereiche festgelegt werden, um Qualität zu prüfen und sicherzustellen, gibt es für Software keinen Toleranzbereich, der ausreichend abgesichert ist. Jeder Fehler kann sich an einer beliebigen Stelle als Fehlerreaktion auswirken.<sup>[23]</sup>

Ähnlich wie Hardware muss Software ebenfalls nach der Entwicklung und Implementierung regelmäßig gewartet werden. In Form von Updates werden Änderungen und Erweiterungen an der ursprünglich ausgelieferten Software vorgenommen. Oftmals kommt es bei der Anzahl an nachträglichen Änderungen auch zum Einbau von Fehlern, die vergleichbar mit einer Art Materialalterung sind.

### **3.2.5 Nutzungsqualität von Software**

Neben den Qualitätsmerkmalen von Software ist in der ISO/IEC-25000-Reihe auch ein Modell enthalten, das den Nutzen von Software unter verschiedenen Einsatzbedingungen bewertet: Das Quality-in-Use-Modell (Nutzungsqualitäts-Modell) definiert die fünf Merkmale Effektivität, Effizienz, Zufriedenheit, Risikofreiheit und Kontextabdeckung, die die Interaktion von Nutzer und Software-Produkt im speziellen Nutzungskontext beschreiben. Das Modell bewertet die Mensch-Computer-Interaktion und außerdem, welche Software, Computersysteme und Umfeld-Systeme enthalten sind.<sup>[41]</sup>

Um die dargelegte Nutzer-Interaktion mit der Software zu veranschaulichen, soll im Folgenden die Nutzungsqualität anhand dreier aufeinander aufbauender Qualitätsaspekte charakterisiert werden (siehe Abb. 3.4).<sup>[34] [42]</sup>

1. Innere Qualität: Diese betrifft zum einen die Software selbst und zum anderen die Hardware, auf der sie eingesetzt wird. Die Software wird auf Grundlage der statischen Eigenschaften des Quellcodes bewertet, der mittels Codereviews oder statischen Analysen aufgenommen wird.
2. Äußere Qualität: Sie ist das Resultat des Zusammenspiels zwischen Software und Hardware und wird von außen über das Reaktionsverhalten oder das Durchlaufen von fachlichen Testszenarien gemessen.
3. Nutzungsqualität: Hier stehen die Erwartungen und Anforderungen der Kunden im Vordergrund. Untersucht werden die Einsatzszenarien im Betriebsumfeld und daraus ergeben sich Erkenntnisse zu Eignung und Tauglichkeit. Der Nutzen für den Kunden und damit die Quality in Use gehen also daraus hervor.

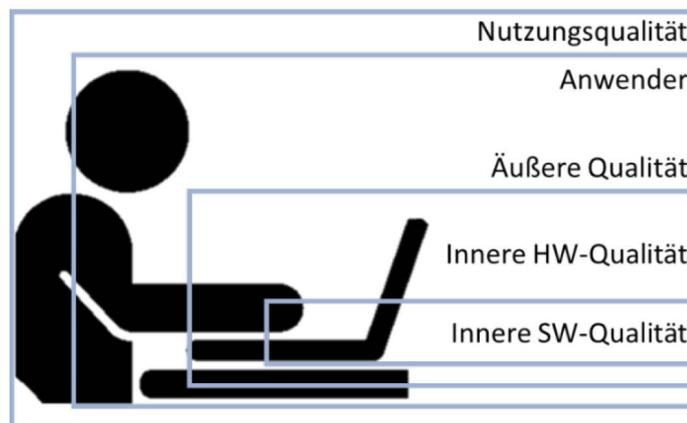


Abbildung 3.4: Zusammenhang zwischen innerer sowie äußerer Software-Qualität und Nutzungsqualität (Quelle: Eigene Darstellung in Anlehnung an Bevan 1999; Vigerschow 2010) <sup>[34]</sup> <sup>[42]</sup>

Die äußere Qualität entsteht aus dem Zusammenspiel von Software und Hardware sowie dem Verhalten der Hardware selbst. Die Nutzungsqualität ist die Kombination aus Effektivität, Produktivität und dem Maß der Zufriedenheit des Anwenders während des Betriebs unter realistischen Bedingungen. Weiterhin umfasst die Nutzungsqualität die äußere Qualität, die Benutzbarkeit sowie die Wechselwirkung mit anderen Umfeld-Systemen. Die Wechselwirkung mit anderen Systemen, in die die Software integriert ist, spielt eine essenzielle Rolle für die Benutzungsqualität und den Eindruck beim Kunden. Neben der Benutzbarkeit steht die Unterstützung der Arbeitsprozesse des Anwenders im Fokus der Nutzungsqualität.<sup>[34]</sup>

Bei der Verwirklichung der Nutzungsqualität besteht eine Wechselwirkung zwischen den drei Qualitätsaspekten. Grundsätzlich gilt: Die innere Qualität ist die Voraussetzung für die äußere Qualität, die wiederum die Voraussetzung für die Benutzungsqualität ist.<sup>[42]</sup>

### 3.3 Grundlagen und Definition der digitalen Dienste

Der Begriff Dienst (auch Service) beschreibt in der Informatik eine technische, autarke Einheit, die zusammenhängende Funktionalitäten bündelt und über definierte Schnittstellen zur Verfügung steht, z. B. E-Mail, Suche, Auktionen, Nachrichten,



der vom Fahrzeugnutzer nicht wahrnehmbar ist und insofern im Hintergrund Steuerungen ausführt.

Der Fahrzeugnutzer hat die Möglichkeit, per Handy (oder App) mit seinem Smartphone eine Kommunikation zu seinem Fahrzeug aufzubauen und so nicht nur passiv Informationen über den Fahrzeugzustand (Tankinhalt bzw. Reichweite sowie Diagnoseergebnisse von verschiedenen Steuerelementen) zu erhalten, sondern auch aktiv Fahrzeugkomponenten zu steuern und zu aktivieren. So kann zum Beispiel die Standheizung schon vorab auf eine spezifische Temperatur vorgeheizt werden, das Sonnendach kann bei Regen per App geschlossen werden und eine Suchfunktion des Fahrzeugs kann aktiviert werden bei unbekanntem Standort; ein Kartendienst führt dann per GPS zum Fahrzeug.<sup>[43]</sup>

### **Verschiedene Arten von Applikationen**

Der Terminus App steht als Abkürzung für Applikation Software und beschreibt eine Anwendungs-Software, die zahlreiche Funktionen erfüllen kann.<sup>[46]</sup>

Apps werden in der Regel bei Smartphones, Spielekonsolen oder auf unterschiedlichen Desktop-Computern genutzt. Mittlerweile sind sie aber auch auf TV-Geräten (mit Smart-TV-Funktion) zu finden. Nachfolgend werden die verschiedenen Arten von Apps kurz umschrieben.<sup>[46]</sup>

**Native App:** Diese Applikations-Software ist an ein spezielles Betriebssystem wie iOS oder Android gebunden. Das bedeutet, dass etwa eine auf iOS programmierte App nur auf einem iPhone ausführbar ist. Die Anwendungsbandbreite von Apps von Rollenspielen über Online-Shops bis hin zu Nachrichten-Apps.<sup>[46] [47]</sup>

**Web App:** Die Web-Applikations-Software ist eine für mobile Endgeräte optimierte Version einer Website. In diesem Zusammenhang wird auch von Responsive Design gesprochen. Web Apps benötigen zur Ausführung einen Browser. Das bedeutet, dass die App beim Öffnen im Browser geladen und dort auch ausgeführt wird. Diese Art von App hat zwei Vorteile: Zum einen müssen die Inhalte nur einmal statt zweimal (auf der Website sowie in der nativen Applikations-Software) aktualisiert werden. Zum anderen können Web Apps von Geräten mit jedem Betriebssystem aufgerufen werden, solange diese einen funktionierende Browser besitzen.<sup>[46] [47]</sup>

**Hybride App:** Diese Applikations-Software besteht aus verschiedenen Webtechnologien wie HTML oder Java Script.<sup>[47]</sup>

Der Kunde ist direkt mit dem Fahrzeug und dem mobilen Endgerät verbunden, weil er diese als Schnittstellen zum digitalen Dienst nutzt. Das Fahrzeug und das mobile Endgerät sind über die Infrastruktur mit dem Backend verbunden. Dies ist jedoch nur möglich, wenn eine funktionierende Infrastruktur gegeben ist. Das Fahrzeug und das mobile Endgerät sind zur Nutzung des digitalen Dienstes miteinander gekoppelt.<sup>[43]</sup>

## Heutige Anwendungsfelder digitaler Dienste in der Automobilindustrie:<sup>[43]</sup>

- Infotainment und Fahrzeug-Apps (z. B. *Nachrichten, Börsenkurse, Wetter, Musik, Facebook-, Twitter- und E-Mail-Nachrichten, aber auch ortsbasierte Werbung, Tankstellen, Restaurants, Hotels, Ladesäulen*)
- Sicherheit und Fahrassistenzsysteme (z. B. *Assistenzsysteme, Car2X-Kommunikation, Informieren nachfolgender Fahrzeuge*)
- Effizienz/Wirtschaftlichkeit (z. B. *Verkehrsfluss optimieren und damit Staus verhindern*)

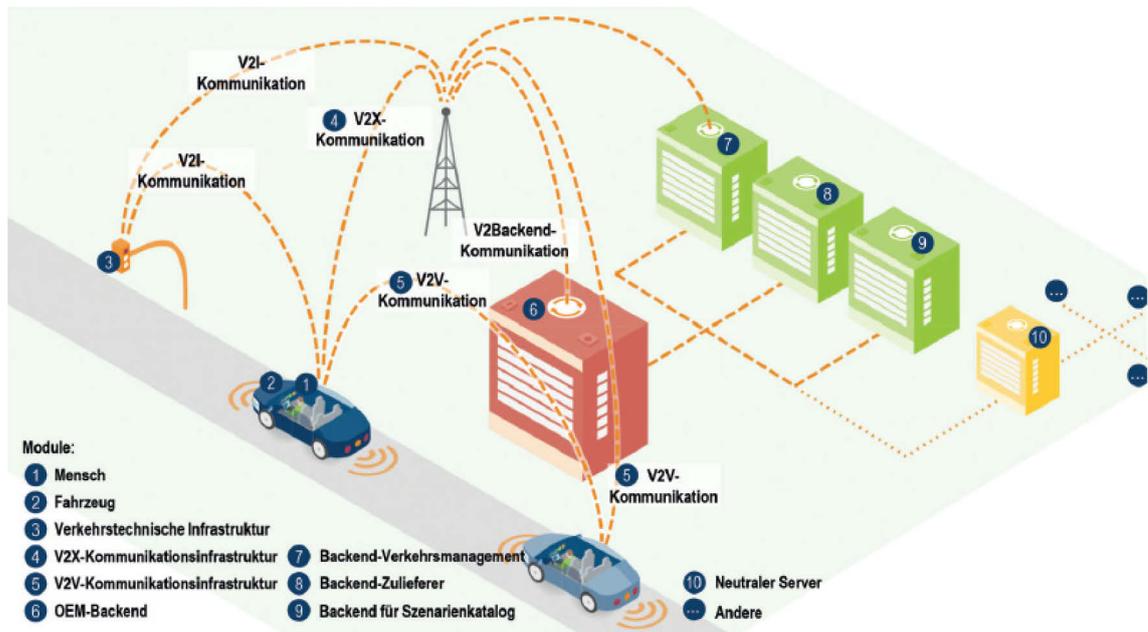


Abbildung 3.6: Neue Auto Mobilität. <sup>[50]</sup>

### 3.3.2 Beispiele von Ökosystemen digitaler Produkte

Die Produkte im Bereich Connectivity sind digital und können bspw. als Apps über Smartphones bezogen und genutzt werden. Die Funktionen und das Angebot der Automobilhersteller werden dadurch nutzbar für Kunden, ob sie ein Automobil besitzen oder nicht, oder ob sie sich innerhalb oder außerhalb des Fahrzeugs befinden. (vgl. Daimler AG 2017<sup>[7]</sup>) Der Kundennutzen von Produkten oder Services im Bereich Connectivity basiert primär auf der Funktionalität der Software-Komponente und bricht somit mit dem traditionellen Produktportfolio der Automobilindustrie.

Über Plattformen und Apps ermöglichen es die Automobilhersteller dem Kunden, Informationen rund um das Automobil abzurufen. (vgl. Daimler AG 2016)

So können das Infotainmentsystem, Türen oder die Standheizung per Smartphone von außen reguliert werden. Des Weiteren kann der Fahrzeugstatus, wie z. B. der Füllstand des Tanks, Standortinformationen oder das Unfall-, Wartungs- und Pannenmanagement, vom Nutzer abgerufen werden. (vgl. Daimler AG 2017)

Der Nutzer kann zudem per App von jedem Standort die kürzeste Route zu seinem Fahrzeug und im Anschluss die optimale Route zum Zielort mit dem Fahrzeug

berechnen lassen. Der Kunde muss also seinen Mobilitätsstatus nicht manuell angeben, sondern die App erkennt automatisch, wann der Kunde im Fahrzeug sitzt und wann er zu Fuß unterwegs ist. Viele der OEMs haben Car Sharing Services, die mit diesem Service kombiniert werden können.

Weitere Apps ermöglichen die Echtzeitverfolgung des Automobils im Herstellungsprozess nach einem Neukauf (vgl. Daimler AG 2017) oder die Analyse des Fahrverhaltens mit anschließender Auswertung, ob der Umstieg auf ein Elektroauto oder Hybridmodell empfehlenswert ist. Die Sprachsteuerung ist durch die Verbindung zum Internet weitaus umfangreicher und dank künstlicher Intelligenz lernfähig. (vgl. Daimler AG, 2018)

Integriert ist zudem die Navigation, die neben den üblichen Funktionen nun zusätzliche Informationen, etwa zu Kraftstoffpreisen oder Parkmöglichkeiten, online abrufen kann. Die Navigation besitzt eine Augmented-Reality-Funktion und verbindet das Echtzeitbild der Frontkamera mit Navigationsvorgaben auf der Fahreranzeige. Außerdem sind vernetzte Fahrzeuge in der Lage, Kartenmaterial weiter auszubauen und zu verbessern.

Die Kommunikation von vernetzten Fahrzeugen liefert Informationen über den Straßenzustand wie Glätte, Schnee, herannahende Rettungsfahrzeuge oder freie Parkplätze, die ebenfalls direkt auf der Navigationskarte dargestellt werden. Die Technologie in den Fahrzeugen kommuniziert aber nicht nur mit anderen Fahrzeugen, sondern ermöglicht auch einen Datenaustausch mit der umliegenden Infrastruktur, wie Ampeln, Baustellen oder Kreuzungen. Der Informationsaustausch geschieht eigenständig über das WLAN oder den Mobilfunk an Bord. Die neuen Technologien werden mit Kamerasystemen und Sensorik kombiniert, die im Zusammenspiel innovative Fahrerassistenzsysteme bilden. (vgl. Daimler AG 2016)

Kollisionsfrühwarnsysteme, autonomes Bremsen bei Gefahr oder teilautomatisiertes Staufolgefahren liefern dem Kunden neue Funktionen, die zur Sicherheit im Straßenverkehr beitragen. Eine Integration des Smartphones mit dem Fahrzeug ist darüber hinaus eine Möglichkeit, um Funktionen des Android Auto oder Apple Car Play zu nutzen. Apps des Smartphones, wie Musik-Streaming-Angebote, können damit über den Touchscreen im Fahrzeug gesteuert werden. Mit dem Einzug von WLAN- und WI-FI-Hotspots erhalten Fahrrauminsassen zusätzlich umfassenden Zugang zum Internet und auch das Fahrzeug selbst verfügt über einen eigenen Internet-Browser.

### **3.4 Standards und Definitionen der Fehlerklassifizierung und des Fehlermanagements**

Da ein digitaler Dienst, wie in Kapitel 3.3 beschrieben, ein überaus komplexes Netzwerk aus verschiedenen Beteiligten darstellt, ist es nicht einfach, Fehler dieses Netzwerkes aus Hardware und Software zu analysieren: Sowohl die Methoden der Automobilbranche als auch die der IT-Branche reichen nicht aus, um Fehler von digitalen

Diensten analysieren zu können. In der vorliegenden Arbeit wird deshalb ein Konzept zur Fehlerklassifizierung und zum Fehlermanagement von digitalen Diensten erarbeitet.

In diesem Kapitel wird zunächst der Fehlerbegriff für digitale Dienstleistungen in der Automobilindustrie definiert. Anschließend werden die Ziele eines Klassifizierungsmodells festgelegt. Bei bestehenden Klassifizierungsmodellen wird eine Übertragbarkeit auf Software-Anwendungen der Automobilindustrie geprüft. Ein zentrales Ziel bildet die Entwicklung eines zweckmäßigen Klassifizierungsmodells mit den dazugehörigen Dimensionen zur Fehleranalyse. Das Modell zur Fehlerklassifizierung soll im gesamten Produktlebenszyklus einheitlich und standardisiert anwendbar sein.

### **3.4.1 Normen und Definitionen von Fehlern**

Ein Fehler ist in der DIN EN ISO 9000 als eine Nichterfüllung einer festgelegten Anforderung definiert. Die Anforderung kann ein vorausgesetztes Erfordernis oder eine Erwartung an ein Produkt sein. Hierzu zählen festgelegte Abweichungen einer messbaren Anforderungsspezifikation, ein Nutzungsausfall, die Nichteinhaltung von Vorschriften, Verpflichtungen oder branchenspezifischen Verfahren. Allgemein sind aus einem Fehler resultierende Fehlerkosten abzuleiten. Diese Grundlage bietet eine Motivation zur näheren Auseinandersetzung mit Fehlern. Es werden die Ziele einer Fehlerbeseitigung und einer künftigen Fehlervermeidung verfolgt. Die Verpflichtung zu ständigen Verbesserungen ist in der DIN EN ISO 9001 formuliert.<sup>[51]</sup>

Das Fundament für die Klassifikation von Fehlern besteht aus unterschiedlichen Definitionen für Fehler und Klassifizierungen. Bei der Definition von Fehlern und Fehlerklassifizierungen wird in der digitalen Umwelt zwischen mindestens sechs Normen, Standardisierungen oder Richtlinien unterschieden. Diese unterschiedlichen Formulierungen für Regeln, Leitlinien bzw. Merkmale und Definitionen werden im nachfolgenden Abschnitt vorgestellt und dienen weiterhin als Leitfaden.

- 1) DIN 66271 Informationstechnik: Software-Fehler und ihre Beurteilung durch Lieferanten und Kunden<sup>[52]</sup>

Ein Fehler ist die Nichterfüllung einer festgelegten Forderung bzw. die Abweichung von einem erwarteten Merkmalswert und führt zu Differenzen zwischen Vertragsparteien. Die Fehlerklassifizierung ist an der Bewertung der Fehlerfolgen ausgerichtet. Es wird zwischen drei Stufen für die Beeinträchtigung des Einsatzes und das Schadensrisiko unterschieden: a) hoch, b) mittel, c) niedrig. Die Priorität und der Behebungsaufwand teilen sich ebenfalls in drei Stufen.

- 2) EN ISO 9000:2005 Qualitätsmanagementsystem<sup>[52]</sup>

Ein Fehler ist die Nichterfüllung einer angegebenen Forderung, eines Erfordernisses oder einer Erwartung. Diese können üblicherweise vorausgesetzt oder vorgeschrieben werden. Ein Mangel ist die Nichterfüllung einer Forderung hinsichtlich des beabsichtigten oder festgelegten Gebrauchs. Mängel haben rechtliche Folgen und unterliegen einer Produkthaftung. Die Fehlerklassifizierung ist nicht definiert.

### 3) DIN 55350 Teil 31, Begriffe zu Qualitätsmanagement und Statistik<sup>[52]</sup>

Ein Fehler ist die Nichterfüllung vorgegebener Forderungen durch einen Merkmalswert. Demnach sind Fehler Merkmalswerte, die außerhalb eines vorgegebenen Toleranzbereiches liegen. Die Verwendbarkeit eines Produktes muss nicht durch die Nichterfüllung der vorgegebenen Forderung beeinträchtigt sein. Die Fehlerklassifizierung und deren Bewertung sind ausgerichtet an den Fehlerfolgen. Es wird zwischen den drei folgenden Stufen unterschieden:

- Kritischer Fehler: Es werden gefährliche und kritische Situationen mit einem Nutzungsausfall erzeugt.
- Hauptfehler: Die Ausfallwahrscheinlichkeit ist wesentlich erhöht oder die Wahrscheinlichkeit der Nutzbarkeit wird herabgesetzt.
- Nebenfehler: Die Brauchbarkeit oder Betriebsbeeinträchtigung ist nur geringfügig beeinflusst.

### 4) IEEE 1044-1993 – Standard Classification for Software Anomalies

Ein Fehler ist jegliche Abweichung von Bedingungen, unter anderem von Anforderungsspezifikationen, Designdokumenten, Benutzerdokumenten und Standards, bezüglich der Erwartungen oder Erfahrungen. Die Fehlerklassifizierung wird indirekt über den aus einem Fehler resultierenden Produktstatus definiert. Es wird zwischen vier Statusmeldungen unterschieden: unbrauchbar, degradiert, beeinträchtigt, nicht beeinträchtigt.

Bei der Statusmeldung ‚beeinträchtigt‘ existiert eine provisorische Fehlerumgehung. Die provisorische Fehlerumgehung wird als Work-Around bezeichnet.

### 5) Telecom Standard TL9000, Quality Management System for the Telecommunications Industry<sup>[52]</sup>

Die branchenspezifische Norm ist eine Erweiterung der ISO 9000 und wird von den meisten bedeutenden Telekommunikations-Herstellern und -Netzbetreibern unterstützt sowie angewendet. Ein Fehler ist nicht explizit definiert. Es wird ausschließlich zwischen einem technischen Fehlverhalten, verursacht unter anderem von Hardware- oder Softwarekomponenten bei Dokumentation, Auslieferung, Service oder Rechnungsstellung, und Handhabungsproblemen bzw. Bedienerfehlern unterschieden. Handhabungsprobleme entstehen bspw. aus inkorrekten Systemeingaben, fehlerhaften Installationsschritten oder bei Nichtbefolgen der Arbeitsweise entsprechend der Benutzerdokumentation.

Hinsichtlich der Fehlerklassifizierung wird zwischen drei Fehlerklassen unterschieden. Diese Fehlerklassen werden stichpunktartig definiert:

- critical (kritisch): Die Beeinträchtigung der Hauptfunktionalität eines Produktes und die geschäftlichen Auswirkungen für den Kunden werden als kritisch eingestuft. Es werden sofortige Korrekturmaßnahmen unabhängig von der Tageszeit oder dem Wochentag erfordert, wie bei:

- a) der Funktionsunfähigkeit des Produktes, vollständigem oder teilweise Ausfall
  - b) der Verringerung der Kapazitäten, d. h. der Fähigkeit zur Verarbeitung von Daten/Datenverkehr sowie der Unfähigkeit, erwartete Lasten zu bewältigen
  - c) dem Verlust der Notfallfähigkeit, beispielsweise von Notrufen
  - d) dem Sicherheitsrisiko oder Risiko zur Nichterfüllung von Sicherheitsmaßnahmen
- major (bedeutend): Die Hauptfunktionen eines Produkts erfahren keine Beeinträchtigung. Die Leistungsfähigkeit und die Produktivität im Betrieb werden allerdings bezüglich der vordefinierten Bearbeitungszeiten bedeutend belastet. Der Kunde hat dadurch ökonomische Beeinträchtigung zu beklagen.
    - a) Reduzierung der Produktkapazität; erwartete Belastung können bewältigt werden
    - b) Verlust der Verwaltungs- oder Wartungssichtbarkeit des Produkts und/oder der Diagnosefähigkeit
    - c) Wiederholter Abbau von essenziellen Komponenten oder Funktionen
    - d) Verschlechterung der Produktfähigkeit, um erforderliche Benachrichtigungen über Fehlfunktionen zu erhalten
  - minor (gering): Es werden keine Beeinträchtigungen der Hauptfunktionalität eines Produkts und der Leistungsfähigkeitsindikatoren im Betrieb erwartet. Die Funktion des Systems wird geringfügig bis gar nicht beeinträchtigt.

6) Nach Six Sigma – Qualitätsziel und Managementmethode<sup>[52]</sup>

Ein Fehler ist ein unerwartetes Verhalten; er wird nicht in der Entstehungsphase identifiziert, sondern während der anschließenden Entwicklungsphase entdeckt. Die Abweichungen werden gemäß der Six-Sigma-Definition nach Freigabe zu Defekten erklärt. Ein Defekt ist ein unerwartetes Verhalten, das über das Produkt bis ins Feld, während der Nutzungsphase beim Kunden oder Endverbraucher, übertragen wird. Die Fehlerklassifizierung unterscheidet nach Six Sigma zwischen zwei Klassen:

- A-Fehler sind fehlerhafte oder fehlende Anforderung und nicht entdeckte Kundenerwartungen oder -bedürfnisse;
- B-Fehler sind fehlerhafte Implementierungen von definierten Anforderungen. Diese Fehler sind beispielsweise eine nicht vollständige Umsetzung von Spezifikationen oder Software-Fehler, wie inkorrekte Implementierungen oder Software-Abstürze.

In den wissenschaftlichen Literaturen werden Begriffe wie Abweichung, Defizit oder Störung verwendet; diese beziehen sich immer auf auftretende Fehler.

Definition Fehler dieser Arbeit: Ein Fehler ist die Nichterfüllung von verfügbaren funktionalen Produktmerkmalen sowie die Abweichung von Kundenerwartungen.

Nach den vorgestellten Normen, Richtlinien und Standardisierungen kann keine einheitliche Fehlerklassifizierung formuliert werden, denn jede Norm, Richtlinie oder Standardisierung betrachtet die Fehlerklassifizierung auf einer anderen Ebene. Die

unterschiedlichen Definitionen für Fehler und Fehlerklassifizierungen dienen als Grundlage für die konzeptionelle Erstellung einer Fehlerklassifizierung für die digitalen Dienste in der Automobilindustrie. Es wird nicht zwischen richtig und falsch selektiert; vielmehr wird die Übertragbarkeit von einzelnen Elementen zur Erstellung der branchenspezifischen Fehlerklassifizierung bewertet. Das Ziel ist die Erstellung einer Fehlerklassifizierung basierend auf den vorgestellten Normen, Richtlinien und Standards.

Die Fehlererfassung und -dokumentation spielt für das reaktive Fehlermanagement eine entscheidende Rolle. Ohne eine standardisierte Erfassung und Dokumentation der Fehler, die während der Entwicklung bspw. durch Tests und während des Betriebes bspw. durch Kundenreklamationen auftreten, sind eine anschließende Fehleranalyse, Klassifizierung und Fehlerbehebung kaum möglich. Somit muss die Relevanz dieses Teils des Fehlermanagements deutlich betont werden. Da die Fehlererfassung und Dokumentation stark von dem angebotenen Produkt abhängig sind, wird hier nicht näher darauf eingegangen. Im Anhang 1 befindet sich eine Liste von Fragen, die im Zuge der Dokumentation von Software-Fehlern bedeutsam sind.

Ein grundlegendes Werkzeug für ein effektives und effizientes Fehlermanagement sind Fehlerklassifizierungen. Sie bilden die Grundlage für standardisierte Verfahren zur Fehlerbehandlung und unterstützen zudem eine kontinuierliche Qualitätsverbesserung im Rahmen des Qualitätsmanagements. Eine Fehlerklassifikation muss immer unternehmensspezifisch an das entsprechende Portfolio angepasst werden. Zentrales Ziel von Fehlerklassifikationen ist es, eine möglichst unternehmensweite einheitliche Entscheidungshilfe für den Umgang mit Fehlern zu geben. Diese sollte sowohl während der Entwicklung und des Testens als auch während des Betriebes einheitlich und standardisiert sein. In Kapitel 7.4 wird detailliert darauf eingegangen.

### **3.4.2 Anforderungsmanagement**

Das Anforderungsmanagement ist ein wesentlicher Bestandteil des Qualitätsmanagements. Laut Rupp ist eine Anforderung eine Aussage über eine zu erfüllende Eigenschaft oder eine zu erbringende Leistung eines Produktes, eines Prozesses oder der am Prozess beteiligten Personen.<sup>[53]</sup> Daraus leitet sich die Definition des Anforderungsmanagements ab: Requirements management is „a systematic approach to eliciting, organizing and documenting the requirements of the system, and process that establishes and maintains agreements between the customer and the project team on the changing requirements of the system.“<sup>[54]</sup>

Das Anforderungsmanagement legt die Anforderungen an ein Produkt fest, die durch entsprechende Tests gemessen und überprüft werden. Fehler, die dabei erkannt werden, werden durch ein effektives Fehlermanagement verhindert bzw. beseitigt. Als Kern dieser Arbeit ist dem Fehlermanagement in Kapitel 7.4 gewidmet.

In Abbildung 3.7 sind die relativen Kosten für die Beseitigung eines Software-Fehlers in Abhängigkeit von der Phase der Software-Entwicklung dargestellt. Es ist deutlich zu erkennen, dass Fehler, die während des Betriebes bzw. der Wartungsphase auftreten,

einen Großteil der relativen Kosten verursachen. Wird zusätzlich in Betracht gezogen, dass 55 % der Fehler, die ein Software-Produkt aufweist, durch eine unzureichende Anforderungsanalyse und -definition zustande kommen, wird die Relevanz klar definierter Anforderungen evident (S. Abb. 3.7).

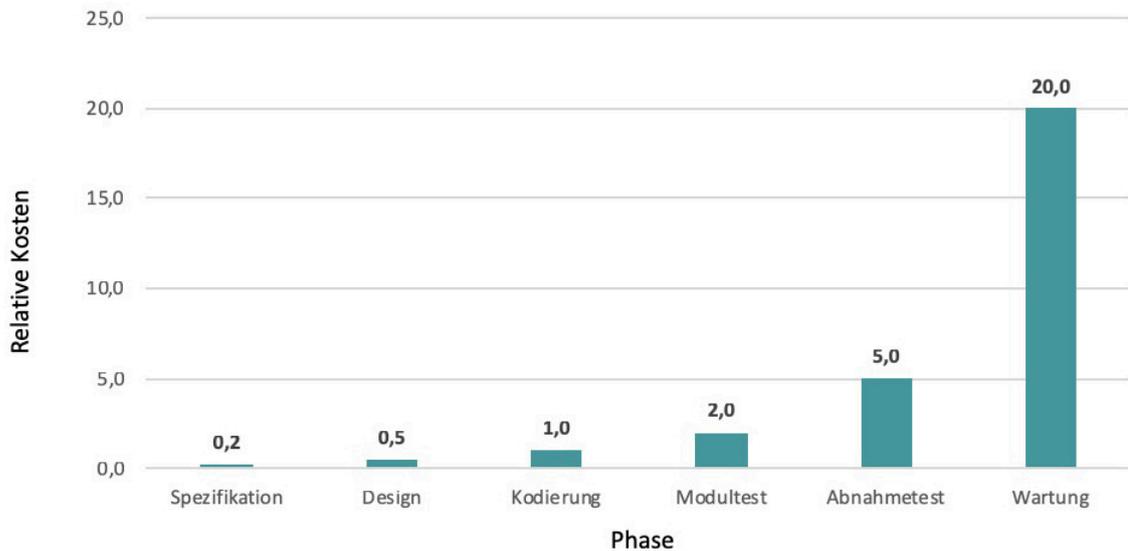


Abbildung 3.7: Relative Kosten für die Beseitigung eines Software-Fehlers (Quelle: Interne Forschungsstudie Fachgebiet Qualitätsstrategie und- kompetenz TU Berlin<sup>[20]</sup>, in Anlehnung an Leffingwell, D.; Widrig, D. (2000)<sup>[54]</sup>)

Als Beispiel für eine Fehlerbehebung in einer späten Phase sind Rückrufaktionen zu nennen. Diese sind mit enormen Kosten verbunden, denn die Rückrufaktionen fallen häufig unter die Garantie bzw. Kulanzregelungen der Hersteller, betreffen schnell Tausende oder sogar Hunderttausende von Fahrzeughaltern und werden für diese meist kostenfrei angeboten. Der dabei einhergehende Imageverlust verursacht ebenfalls beträchtliche finanzielle Schäden.

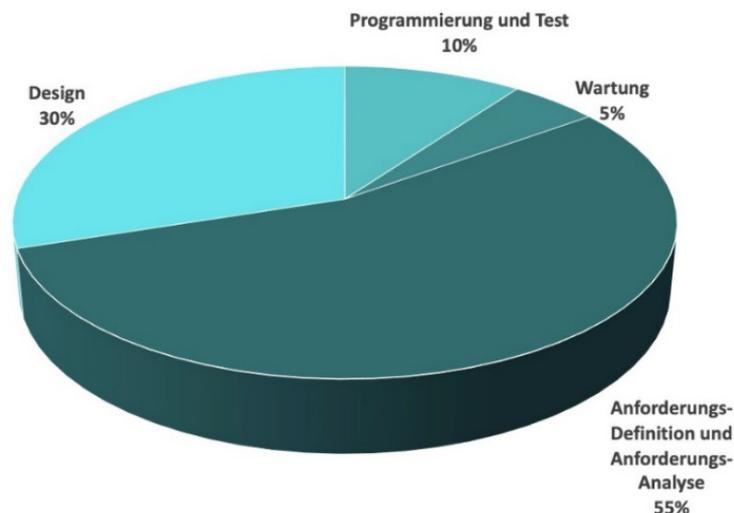


Abbildung 3.8: Eingeführte Software-Fehler in der Anforderungsentwicklung (Quelle: Interne Forschungsstudie Fachgebiet Qualitätsstrategie und- kompetenz TU Berlin<sup>[20]</sup>, in Anlehnung an Granden, M. (2011)<sup>[55]</sup>)

Die Dokumentation der Anforderungen erfolgt in einem Lasten- und Pflichtenheft und wird im Laufe des PEP durch das Änderungsmanagement angepasst. Das Lastenheft beschreibt alle Forderungen des Auftraggebers bezüglich der Leistungen und Lieferungen an den Auftragnehmer. Synonyme für Lastenheft sind Spezifikation, Kundenspezifikation oder Anforderungsspezifikation.<sup>[55]</sup> Im Pflichtenheft beschreibt der Auftragnehmer, wie er die Forderungen aus dem Lastenheft umzusetzen gedenkt. Die hier spezifizierten Anforderungen bilden oftmals auch die Grundlagen für die Werk- bzw. Dienstverträge der Lieferanten. Erst mit dem Dokumentieren der Anforderungen und den damit verbundenen Abnahmekriterien können die Anforderungen durch spätere Tests überprüft werden. Je detaillierter die Anforderungen formuliert und kommuniziert werden, desto wahrscheinlicher wird es, ein fehlerarmes Produkt zu erhalten, das den Anforderungen des Auftraggebers entspricht.<sup>[55]</sup>

In agilen Projekten stellt das frühe detaillierte Festlegen von Produktanforderungen jedoch eine Herausforderung dar, denn hier werden, wie oben erwähnt, die Anforderungen erst im Laufe des PEP durch iterative Schritte in enger Absprache mit dem Auftraggeber konkretisiert. Darauf wird im Kapitel 4.8 über agile Entwicklungsmethoden umfassender eingegangen.

## 4 Stand der Forschung

Wie im letzten Kapitel bereits erläutert wurde, werden digitale Dienste in heutigen Fahrzeugen vermehrt genutzt, ohne zu wissen, worum es sich bei digitalen Diensten genau handelt. Deshalb wird zunächst der Begriff der digitalen Dienste definiert und die Umgebung eines digitalen Dienstes mit allen relevanten Schnittstellen beschrieben.

Im vorherigen Kapitel wurde die aus dem Kontext der Plattformentwicklung resultierende Notwendigkeit für das Eingehen neuer Kooperationen von OEMs mit Start-ups aufgezeigt. Dabei wurde festgestellt, dass Kooperationen im Bereich Connectivity insbesondere mit Software-Start-ups erfolgen werden.

In der wissenschaftlichen Literatur sind zahlreiche Definitionen zu Start-ups zu finden. Es lässt sich feststellen, dass Start-ups ein schnelles Wachstum bei extremer Unsicherheit über die weitere Entwicklung anstreben.<sup>[56]</sup> Insbesondere Software-Start-ups sind mit der Herausforderung konfrontiert, unter dynamischen und unsicheren Marktgegebenheiten rasch zu wachsen. Daher kommt es dazu, dass viele Start-ups bereits in der Frühphase scheitern: Etwa 60 % halten sich nicht länger als fünf Jahre am Markt.<sup>[57]</sup> Trotz dieser Quote gelten Start-ups als bedeutender Treiber für das wirtschaftliche Wachstum eines Landes.<sup>[58]</sup> Eine nähere Betrachtung des Kontextes, in dem Start-ups agieren, wird im Folgenden anhand von typischen Merkmalen nach einer Veröffentlichung von Sutton erläutert.<sup>[59]</sup>

- Kurze oder noch gar keine Dauer der Geschäftstätigkeit – Start-ups haben wenig Erfahrungen in Entwicklungsprozessen und im Organisationsmanagement.
- Limitierte Ressourcen – Start-ups fokussieren sich darauf, ein Produkt auf den Markt zu bringen und strategische Partnerschaften einzugehen.
- Vielfältige Einflüsse – Start-ups agieren unter dem Einfluss verschiedener Interessensgruppen (Investoren, Wettbewerber, Kundenwünsche, Partner), die ihre Entscheidungen in hohem Maße beeinflussen.
- Dynamische Technologien und Märkte: Start-ups aus dem Softwarebereich können neue Märkte oftmals nur durch die Entwicklung von disruptiven Innovationen erschließen.

Es ist zu erwähnen, dass die Heterogenität zwischen verschiedenen Start-ups zu einer variierenden Ausprägung der beschriebenen Merkmale führt. Laut der jährlichen Studie des Bundesverbands Deutsche Start-ups e.V. (BVDS) gelten Unternehmen, die jünger als zehn Jahre sind, als Start-ups.<sup>[58]</sup> Somit wird deutlich, dass zwischen den beiden Extremwerten der Definitionsgrenze eine hohe Diskrepanz bezüglich der Reifegrade von Start-ups vorherrscht. Im Hinblick auf die Literaturrecherche kann vermutet werden, dass es für die konkrete Umsetzung von Kooperationen zwischen OEMs und Start-ups kein allgemeingültiges Vorgehen gibt. In Anbetracht der Tatsache, dass Geschäftsfelder im Bereich Connectivity überaus zukunftsgerichtet sind, liegt die Vermutung nahe, dass Start-ups aus diesem Bereich tendenziell einen niedrigen Reifegrad aufweisen.

## 4.1 Benchmarking

“Benchmarking is the search for industry best practices that lead to superior performance.” (Bob Camp, Xerox Inc., 1989) <sup>[60]</sup>

### 4.1.1 Grundlagen und Zielsetzung Bechmarking

Benchmarking ist die Durchführung eines Leistungsvergleichs von Unternehmen gegenüber dem Klassenbesten oder branchenfremden Unternehmen. Leistungsindikatoren bewerten Prozesse, Produkte, Strategien und Methoden. Das Ziel der Gegenüberstellung bildet eine Aufdeckung von Qualitäts- und Leistungspotenzialen. Mittels des systematischen Vergleichs können die identifizierten Leistungslücken mit den Methoden und Erfahrungen des Klassenbesten (Best Practice) implementiert werden. Das Instrument des Benchmarkings ermöglicht einen kontinuierlichen Verbesserungsprozess und gewährt einen ökonomischen Unternehmenserfolg. <sup>[61]</sup>

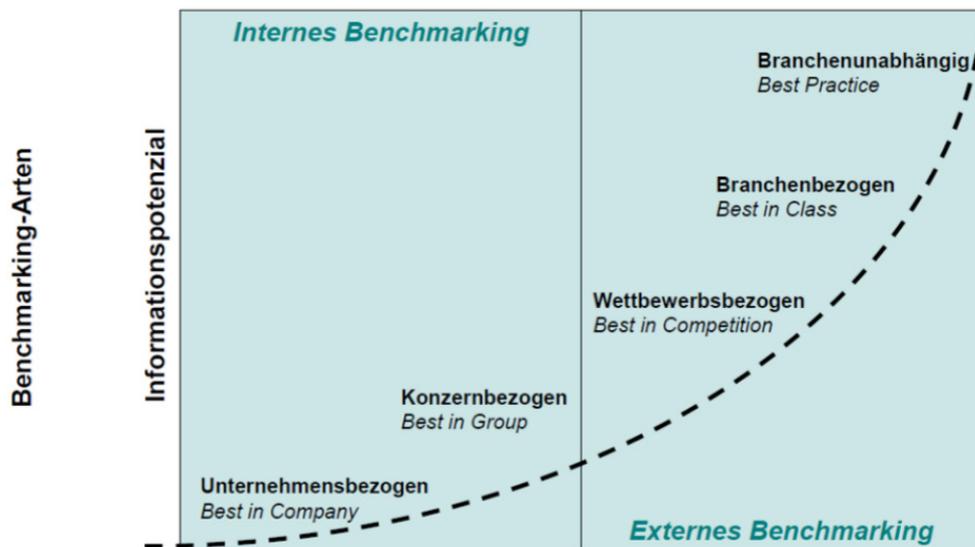


Abbildung 4.1: Benchmarking-Arten im Vergleich.

Quelle: Kohl, H. Prof. Dr.-Ing. (2016) <sup>[60]</sup>

Es wird zwischen internen und externen Arten des Benchmarkings unterschieden. Beim internen Benchmarking richtet sich der Fokus auf einen Leistungsvergleich von ähnlichen Funktionen und Prozessen innerhalb eines Unternehmens oder Konzerns. Das Ziel des Aufspürens des Verbesserungspotenzials wird mit einem geringen Zeit- und Kostenaufwand erreicht. Das externe, unternehmensübergreifende Benchmarking wird in ein wettbewerbsorientiertes und ein funktionales Benchmarking unterteilt. Das Wettbewerbs-Benchmarking vergleicht direkte unternehmerische Mitbewerber im Hinblick auf eine eindeutige Positionierung des Klassenbesten. Lösungsansätze für Defizite werden aus den Methoden und Prozessen des Klassenbesten (im Idealfall: Best Practice) initiiert. Die Zielsetzung verfolgt eine Verbesserung der Marktposition sowie eine Stärkung der Wettbewerbsfähigkeit. Die Wahl der Vergleichspartner ist ausschlaggebend für das Informationspotenzial. <sup>[61]</sup>

#### 4.1.2 Durchführung des Benchmarkings

Für die Durchführung eines zielorientierten Benchmarkings sowie eine schrittweise Annäherung an den Klassenbesten sind die nachstehenden Prozessabfolgen zu berücksichtigen:<sup>[61]</sup>

- Ziel- und Objektfestlegung für das Benchmarking
- Geeignete Unternehmen und Informationsquellen für den Vergleich eruieren
- Leistungskennzahlen (KPI) erfassen, auswerten und vergleichen
- Identifikation von Schwachstellen und deren Ursachen
- Ergebnisse der Analyse transparent darstellen
- Verbesserungsmaßnahmenplan erarbeiten
- Systematische Umsetzung der Verbesserungsmaßnahmen

Kernprodukte eines Automobilherstellers lassen sich in Sparten (PKW, VAN, NFZ) unterteilen und dienen als Grundlage für die digitalen Dienste. Digitale Dienstleistungen in der Automobilbranche erfordern für die Anwendung eine Informations- und Kommunikationsinfrastruktur, die in Industrieländern vorhanden ist. Die Fahrzeugsparten und die Beschränkung der Marktbetrachtung auf traditionelle Industrieländer dienen bei der Wahl des Automobilherstellers hinsichtlich des Benchmarkings als Selektionsfilter. Infolge von zahlreichen Fusionen und Akquisitionen besteht die Automobilindustrie heute aus einer Konzernstruktur, die sich in Konzernmarken gliedern lässt. Im Leistungsvergleich des Benchmarkings werden die einzelnen Automobilhersteller oder -marken miteinander verglichen und nicht die übergeordneten Automobilkonzerne. Für die Wahl geeigneter Konzernmarken dienen der europäische und der US-amerikanische Fahrzeugmarkt als Referenz.

Die Wahl von innovativen Herstellern erfolgt mittels einer Onlinerecherche. Innovative Automobilhersteller sind nach der hier verwendeten Definition die neuen Wettbewerber der traditionellen Automobilhersteller. Demnach sind innovative Hersteller neue Automobilhersteller wie TESLA mit einem vollelektrischen Antriebsstrang; Automobilhersteller aus Asien die in absehbarer Zukunft in die Märkte der Industrieländer eindringen werden; oder branchenfremde Unternehmen wie Google und Apple. Der Begriff innovativ bezieht sich auf neue digitale Technologien oder neuartige Antriebskonzepte. Im Fokus der Recherche nach innovativen Herstellern stehen hier die weltweit größten Automessen: die IAA in Frankfurt, der Pariser Salon, der Genfer Autosalon, die North American International Auto Show oder die Tokyo Motor Show. Eine letzte Auswahl von Partnerunternehmen für das Benchmarking erfolgt über den J.D. Power Report 2017, eine Kundenzufriedenheitsstudie des US-amerikanischen Marktes.<sup>[62]</sup> Aus den insgesamt fünf Kategorien zur Benchmarking-Partnerwahl werden jeweils zehn Automobilhersteller untersucht. Die Kategorie ‚abhängige Leistungsmerkmale‘ filtert die zehn besten, häufigsten oder beliebtesten Hersteller. In der Kategorie ‚Kundenzufriedenheit Report‘ erfolgt die Auswahl der Benchmarking-Unternehmen in Abhängigkeit des Rankings der J.D.-Power-Zufriedenheitsstudie. Die

ausgewählten Benchmarking-Partner aus der Automobilbranche sind kategorisiert zusammengefasst (s. Anhang 2).

Das zentrale Ziel des Benchmarkings verfolgt die Ermittlung des Klassenbesten in Bezug auf die Kundenzufriedenheit mit digitalen Diensten in der Automobilindustrie. Beim vorliegenden Benchmarking sind die digitalen Dienstleistungen beschränkt auf mobile Anwendungs-Software (Apps) für iOS und Android-Geräte. Der Kunde benötigt für die Verwendung einer Software-Applikation ein mobiles Endgerät. Die mobilen Endgeräte unterscheiden sich neben Modellen und Herstellern bezüglich des mobilen Betriebssystems, hier Android und iOS. In Abhängigkeit des mobilen Betriebssystems ist eine Vertriebsplattform zu wählen. Die für das Benchmarking relevanten Vertriebskanäle sind folglich der Google Play Store (Android) und der App Store von Apple (iOS). Beide Vertriebskanäle verfügen über ein identisches Bewertungssystem zur Erfassung der Kundenzufriedenheit: Die Bewertung einer App erfolgt über ein Vergabesystem von Sternen. Die Anzahl der Sterne ist eine Leistungskennzahl (KPI) sowie ein Indikator für die Kundenzufriedenheit. Eine mobile Software-Anwendung kann mindestens einen Stern und höchstens fünf Sterne erreichen. Je mehr Sterne ein Kunde einer Applikation verleiht, desto höher ist seine persönliche Kundenzufriedenheit.

Die Leistungsperformance eines Herstellers bezüglich der Kundenzufriedenheit lässt sich zudem aus Kundenrezensionen, also von Kunden verfassten Textdokumenten, ableiten. Für das Benchmarking entfallen die Kundenrezensionen als Leistungsindikator (KPI) der Kundenzufriedenheit, da die Textdokumente keine messbaren numerischen Kenngrößen für einen Leistungsvergleich aufweisen. Zusammenfassend dient das fünfstufige Bewertungssystem der beiden Vertriebskanäle mit den numerischen Leistungskennzahlen (KPI) als Grundlage für den Leistungsvergleich im Benchmarking. Bei der Auswertung von Leistungskennzahlen muss die ungleiche Anzahl von Bewertungen in Bezug auf die Vertriebsplattform berücksichtigt werden. Für einen qualitativen Leistungsvergleich wird ein gewichteter Durchschnittswert berechnet. Dieser ermittelt den Klassenbesten in Bezug auf die Kundenzufriedenheit mit digitalen Diensten in der Automobilindustrie.

Die Formel des gewichteten Durchschnittswerts lautet wie folgt:

*Durchschnitt mit Gewichtung*

$$= \frac{\left( (Bew. \emptyset(iOS) * Anz. Bew. (iOS)) + (Bew. \emptyset(Android) * Anz. Bew. (Android)) \right)}{(Anz. Bew. (iOS) + Anz. Bew. (Android))}$$

*mit Bew.  $\emptyset$  (Vertriebskanal)*

*= durchschnittliche Bewertung in Abhängigkeit des Vertriebskanals*

*mit Anz. Bew. (Vertriebskanal)*

*= Anzahl der Bewertung in Abhängigkeit vom Vertriebskanal*

### 4.1.3 Ergebnisse des Benchmarkings der digitalen Dienste

Beim Benchmarking (Best in Class) wird der klassenbeste Automobilhersteller in Bezug auf die Kundenzufriedenheit mit digitalen Diensten in der Automobilindustrie ermittelt. Mit einer (halb-)automatischen Aktualisierungsfunktion, auch Update genannt, werden Fehlerkorrekturen, Störungsbeseitigungen und Verbesserungen ausgeführt. Außerdem kann sich im Zuge einer Software-Aktualisierung der Funktionsumfang einer Applikation ändern. Updates können demnach einen Einfluss auf die Kundenzufriedenheit aufweisen. Das Ergebnis des Benchmarkings ist aufgrund von kontinuierlichen Veränderungen der Applikationen zeitabhängig. Die nachstehenden Ergebnisse des Benchmarkings mit den dazugehörigen KPI beziehen sich auf den 3. März 2018.

Hersteller	Anzahl Remote-Zugriff	Anzahl Statusabfragen	Anzahl Unterstützung	Σ
Mercedes me	5	6	2	13
Car-Net	4	6	3	13
BMW Connected	3	4	2	9
Audi MMI connect	4	5	2	11
FordPass	n/a	1	1	2
myOpel	2	5	3	10
Skoda Connect	2	5	3	10
Toyota Service	n/a	n/a	1	1
MY Renault	n/a	n/a	1	1
Volvo On Call	5	6	3	14
MY DAF	n/a	n/a	1	1
IVECO Non Stop	n/a	n/a	1	1
DAILY Business UP	2	5	2	9
MAN mobile24	n/a	n/a	n/a	0
MAN Telematics	n/a	2	1	3
Remote Truck (Mercedes-Benz)	3	4	n/a	7
Scania Fleet	2	2	1	5
My Truck (Volvo)	2	4	n/a	6
Fiat Ducato Camper Mobile	n/a	2	n/a	2
Tesla	6	4	1	11
Lexus Enform Remote	2	4	2	8
Porsche Car Connect	3	5	3	11
myBuick	4	5	3	12
Hyundai Service Guide	n/a	n/a	n/a	0
myChevrolet	4	5	2	11
My Honda	n/a	4	3	7
Jaguar InControl Remote	4	4	2	10

Hersteller	Anzahl Remote-Zugriff	Anzahl Statusabfragen	Anzahl Unterstützung	Σ
1. Tesla	6	4	1	11
2. Mercedes me	5	6	2	13
2. Volvo On Call	5	6	3	14
3. Car-Net	4	6	3	13
3. Audi MMI connect	4	5	2	11
3. myBuick	4	5	3	12
3. myChevrolet	4	5	2	11
3. Jaguar InControl Remote	4	4	2	10

Hersteller	Anzahl Remote-Zugriff	Anzahl Statusabfragen	Anzahl Unterstützung	Σ
1. Volvo On Call	5	6	3	14
2. Mercedes me	5	6	2	13
2. Car-Net	4	6	3	13
3. myBuick	4	5	3	12

Tabelle 4.1: Funktionsanalyse Automobilhersteller, Anzahl Funktionen

(Stand: März 2018, Quelle: Eigene Darstellung, Forschungsstudie QSK, TU Berlin)

Nach der methodischen Vorgehensweise werden die Hersteller der Automobilbranche unter Verwendung einer standardisierten Vorlage zunächst bezüglich der Funktionalität untersucht. In einer Funktionsanalyse werden die verfügbaren Funktionen einer

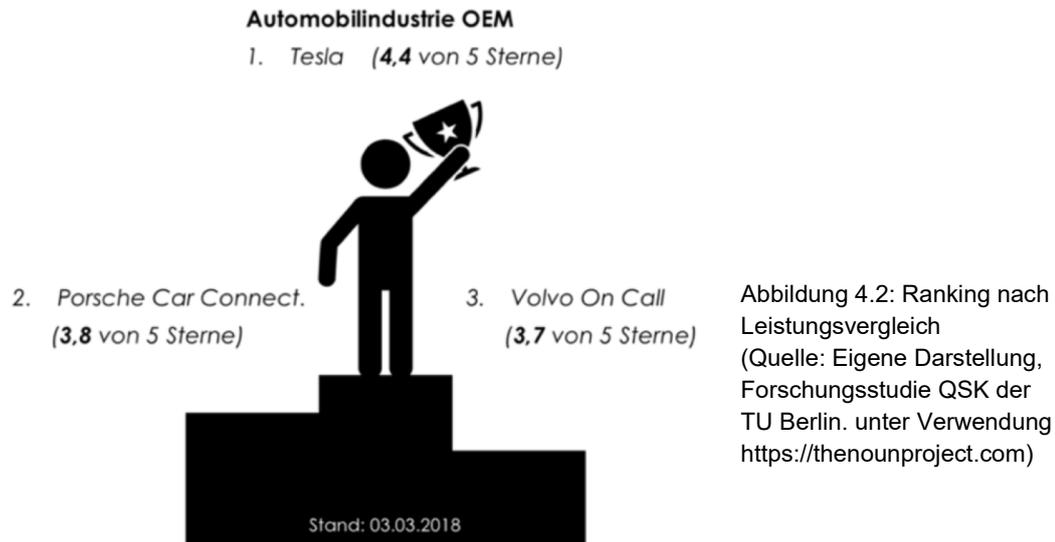
Anwendungssoftware aufgeschlüsselt und einheitlich dokumentiert. Für den anschließenden Leistungsvergleich des Benchmarkings muss nach meiner Definition die App über eine Remotefunktion verfügen. Die eigens definierte Mindestanforderung dient zur qualitativ funktionalen Vergleichbarkeit von herstellerübergreifenden Apps. In einer allgemeinen Funktionsübersicht sind alle verfügbaren Funktionen kategorisiert zusammengefasst (s. Anhang 16). Die Funktionsübersicht dient als Grundlage für die Erstellung einer herstellerbezogenen Funktionsübersicht. Die Kategorien, Remote-Zugriff (grün), Statusabfragen (orange) und Unterstützung (gelb) erhalten in der Tabelle eine farbliche Zuordnung. Dieses Farbschema wird konsequent in den nachfolgenden Untersuchungen und Tabellen übernommen. Herstellerspezifische Sonderfunktionen der Kategorie „weitere Funktionen“ werden beim tabellarischen Funktionsvergleich vernachlässigt, da diese herstellereinspezifischen Sonderfunktionen in einer nachstehenden Auswertung von Kundenrezensionen keinen nachweisbar negativen Einfluss auf die Kundenzufriedenheit aufweisen. Für den nachstehenden Vergleich von Leistungskennzahlen entfallen insgesamt zehn Fahrzeughersteller wie erkennbar durch „n/a“ (s. Tab. 4.1). Die Abkürzung „n/a“ steht für nicht verfügbar („not available“). Der tabellarische Funktionsvergleich gewährt weitere Vergleichsmöglichkeiten. Der innovative Hersteller Tesla besitzt quantitativ die meisten Remotefunktionen. Insgesamt bietet „Volvo On Call“-App seinen Kunden die meisten Funktionen unter Berücksichtigung der tabellarischen Funktionskategorien. Bei der Auswertung der folgenden Leistungskennzahlen wird ein Bezug zu der quantitativen Funktionszahl genommen.

Die Leistungspunkte im Sterneformat dienen als Leistungskennzahlen zur Ermittlung des Klassenbesten. Da die Leistungszahlen aus zwei unterschiedlichen Vertriebskanälen stammen, muss in einem weiteren Prozessschritt ein Durchschnittswert berechnet werden. Die Anzahl der Kundenbewertungen ist nicht gleichmäßig auf die beiden Vertriebsplattformen verteilt. Diese ungleiche Verteilung der Kundenbewertungen muss bei der Durchschnittsberechnung berücksichtigt werden (s. Formel des gewichteten Durchschnittswerts). Der gewichtete Durchschnittswert ist eine aufbereitete Leistungskennzahl zur Ermittlung des Klassenbesten im Leistungsvergleich des Benchmarkings. Für eine Leistungsbeurteilung von Applikationen und eine numerische Bewertungsanalyse wird ein Tool zur Überwachung entwickelt – die Watchlist (s. Anhang 3). Zyklische Aktualisierungen der Watchlist sollen Fehler frühzeitig identifizieren und beheben. Mit der Entwicklung von Algorithmen für Prognosen und Trends können Maßnahmen frühzeitig und präventiv eingeleitet werden. Die Bewertungsverteilung kann grafisch visualisiert werden. Zahlreiche bedeutende Unternehmen verwenden bereits heute ein vergleichbares Tool zur App-Überwachung.<sup>[63] [64]</sup>

Die Watchlist mit den gebündelten Leistungskennzahlen bildet die Grundlage für die Auswertungen und somit die Ermittlung des Klassenbesten Automobilherstellers. Hinsichtlich des gewichteten Durchschnittswertes als alleiniges Leistungsmerkmal weist Tesla mit Abstand den höchsten Zahlenwert auf. Folglich ist der innovative

Automobilhersteller Tesla als klassenbester Hersteller in Bezug auf die Kundenzufriedenheit mit digitalen Diensten in der Automobilindustrie ermittelt (s. Abb. 4.2).

Ranking der aktuellen Top-Performer (Top 3)



Der erreichte Durchschnittswert von 4,4 von 5 (= 88 %) deutet auf eine hohe Kundenzufriedenheit. Der zweite Rang wird mit 3,8 von 5 (= 76 %) von Porsche Car Connect besetzt. Die Differenz von 0,6 zwischen den Platzierungen ist hinsichtlich des Ergebnisses als eindeutig einzustufen. Den dritten Platz nimmt Volvo On Call mit 3,7 von 5 (= 74 %) ein. Die letzte Platzierung im Leistungsvergleich nimmt Car-Net von Volkswagen mit 1,7 (= 34 %) ein. Der Branchendurchschnitt liegt bei 3,0 von 5 (= 60 %) und folgende Unternehmen befinden sich neben Volkswagen unter dem Durchschnittswert: Audi (2,9/5), Skoda (2,7/5), BMW (2,5/5), Jaguar (2,5/5), Lexus (2,4/5) und Mercedes-Benz (2,1/5). Damit liegen 7 von 13 (= 54 %) unter dem Branchendurchschnitt.

Die Erwartungshaltungen von Käufergruppen an einen Automobilhersteller können voneinander abweichen bzw. variieren. Folglich können zwei Hersteller eine identische Qualität bieten und unterschiedliche Ergebnisse bezüglich der Kundenzufriedenheit erzielen.

Im anschließenden Vergleich von Leistungskennzahlen werden Mobilitätsdienstleister und weitere Unternehmen (Best Practice) nach dem numerischen Auswertungsverfahren der Automobilindustrie untersucht. Demnach dient hierbei ebenfalls der gewichtete Durchschnittswert als Leistungskennzahl zur Identifizierung des Klassenbesten. Diese beiden Kategorien werden im Leistungsvergleich separat betrachtet. Die Ergebnisse können aufgrund der identischen Leistungskennzahl mit denen der Automobilhersteller verglichen werden. Im Fokus des zusätzlichen Leistungsvergleichs stehen nicht die Top-Performer des Rankings, sondern die numerischen Vergleichswerte für die Automobilbranche. Bezugnehmend auf die

Bewertungszahlen sind sowohl innerhalb der Automobilbranche als auch branchenübergreifend große Differenzen festzustellen (s. Tab. 4.2).

Hersteller App	Anzahl Bewertungen
car2go	32.882
DriveNow Carsharing	7.218
Drivy	4.845
Moovel	2.313
eCooltra	1.777
Free2Move	1.648
Flinkster	1.439
Coup eScooter Sharing	1.322
Cambio	528
myScotty	74
Ø	5.405

Hersteller App	Anzahl Bewertungen
Instagram	60.517.793
Spotify	10.759.449
ebay	2.985.987
Amazon	851.210
Sonos	102.090
Huawei HiLink	88.495
Samsung Smart View	74.381
McDonalds	10.927
Philips Hue	8.691
Apple Store	634
Bosch Smart Home	341
Ø	6.854.545

Hersteller App	Anzahl Bewertungen
myChevrolet	6.761
Mercedes me	4.655
Volvo On Call	3.359
Tesla	3.144
Car-Net	2.886
myOpel	2.754
Audi MMI connect	2.753
BMW Connected	2.544
Lexus Enform Remote	1.052
myBuick	731
Jaguar InControl Remote	679
Skoda Connect	577
Porsche Car Connect	195
Ø	2.468

Tabelle 4.2: Anzahl der Bewertung (Eigene Darstellung)

Die absolute Differenz innerhalb der Automobilbranche zwischen dem Meistbewerteten und dem Hersteller mit den wenigsten Bewertungen liegt bei 6.566 Bewertungen bei einem Durchschnittswert von 2.468 Bewertungen. Die durchschnittliche Bewertungsanzahl der Mobilitätsdienstleister beträgt 5.405 Einheiten und die der weiteren Unternehmen 6.854.535 Bewertungen. Die Anzahl der Bewertungen ist ein Indikator für die Anzahl der Nutzer und kein Qualitätsmerkmal der Kundenzufriedenheit. Bei der Überwachung von Apps kann eine sinkende Nutzerzahl auf eine Abweichung der Kundenzufriedenheit hindeuten. Der durchschnittliche gewichtete Leistungswert von branchenfremden Unternehmen ist mit 3,9 von 5 (= 78 %) größer als der Wert der Automobilbranche. Die Kategorie der weiteren Unternehmen kann in Unternehmen der IT-Branche sowie klassische Unternehmen, die eine mit der Autobranche vergleichbare Digitalisierung erleben, selektiert werden. Die Durchschnittsbewertung der IT-Unternehmen beträgt 4,4 von 5 (= 88 %) und die der klassischen Unternehmen 3,0 von 5 (= 60 %) (s. Anhang 5). Die klassischen Unternehmen weisen, wie die Automobilhersteller, große Abweichungen bezüglich der Kundenzufrieden auf. Die Automobilhersteller sollten demnach in Zukunft Maßnahmen zur Erhöhung der Kundenzufriedenheit einleiten.

Die Differenz bezüglich der Leistungskennzahl zwischen dem besten und dem schlechtesten Hersteller unterscheidet sich in Abhängigkeit der jeweiligen Kategorie. Der

herstellerspezifische Performanceunterschied ist in der Automobilindustrie mit 2,7 am größten. Im brancheninternen Herstellervergleich der Automobilindustrie sind die gewichteten Durchschnittswerte der Hersteller inhomogen. Die Unternehmen der Mobilitätsdienstleister sowie der IT-Branche weisen demgegenüber eine konstante Leistungsperformance auf einem insgesamt höheren Leistungsniveau auf. Die Differenz beträgt 1,1 bei den Mobilitätsdienstleistern und 0,4 bei den IT-Unternehmen. Die Streuung um den Durchschnittswert in der internen Bewertungsverteilung eines Unternehmens (1-5) wird anhand der Standardabweichung erfasst (s. Anhang 4).

Eine niedrige numerische Standardabweichung deutet auf eine geringe Streuung um den Durchschnittswert hin. In diesem Fall weist die Gesamtheit der Nutzer eine vergleichbare Kundenzufriedenheit auf. Der ermittelte Durchschnittswert ist als eindeutig zu werten, denn die Anzahl abweichender Kundenbewertungen gegenüber dem Durchschnittswert ist schwach ausgeprägt. Die Nutzer neigen insgesamt bei der Kundenbewertung zu extremen Randbewertungen, womit die Vergabe von einem oder fünf Sternen gemeint ist (Beispiel s. Anhang 6). Demnach sind Kunden branchenübergreifend von mobilen Software-Anwendungen begeistert oder enttäuscht. Die Kundenzufriedenheit hängt nach dem Kano-Modell vom Erfüllungsgrad der Anforderungen ab. Beim Kano-Modell wird zwischen Basis-, Leistungs- und Begeisterungsmerkmalen unterschieden. Einen wesentlichen negativen Einfluss auf die Kundenzufriedenheit hat die Nichterfüllung von Basisanforderungen, wie Funktionseinschränkungen aufgrund von Fehlern oder Störungen während der Nutzungsphase.<sup>[65]</sup>

## 4.2 Kundenanforderungen

Die Beschaffenheit eines Produkts orientiert sich möglichst nah an den Anforderungen des Kunden hinsichtlich Funktion, Lieferzeit, Preis und Service. Damit Kundenanforderungen in die Entwicklung und Fertigung eines Produktes einfließen können, muss diesen eine Untersuchung der erforderlichen Qualitätsmerkmale von Produkten oder Dienstleistungen vorausgehen.<sup>[66]</sup>

### 4.2.1 Grundlagen der Kundenanforderungen

Das Spannungsgefüge zwischen Kunde und Lieferant (s. Abb. 4.3) wird von Masing veranschaulicht.<sup>[66]</sup> Angefangen auf der Kundenseite erwartet der Kunde ein Produkt, das seinen Ansprüchen entsprechen soll. Diese Ansprüche formuliert der Kunde in Qualitätsanforderungen, die wiederum mit einer Vorstellung über den Nutzungszeitpunkt und einem Preis, den der Kunde zu zahlen bereit ist, verbunden sind. Aus Herstellersicht wird ein materielles oder immaterielles Produkt gefertigt und dem Markt zur Verfügung gestellt. Dieses Produkt besitzt eine bestimmte Beschaffenheit, die zunächst die wertfreie Gesamtheit aller Merkmale und Eigenschaften des Produkts darstellt. Anhand der Gegenüberstellung der Beschaffenheit und den Forderungen sowie Erwartungen erfolgt eine Aussage über die Produktqualität. Die Nichterfüllung einer Forderung kommt einem Fehler gleich und kann nicht durch die übermäßige Erfüllung einer anderen

Produkteigenschaft ausgeglichen werden. Dem Produzenten fallen Kosten bei der Erstellung des Produkts oder der Dienstleistung an; außerdem ist die Verfügbarkeit der Leistung in zeitlicher Hinsicht zu berücksichtigen. Die Parameter Anforderungen (bzw. Beschaffenheit), Preis und Termin bilden das sogenannte magische Dreieck auf der Nutzerseite (bzw. Herstellerseite). Die Eckpunkte des Dreiecks sind abhängige Variablen und lassen sich nicht unabhängig voneinander wählen. Besteht nun Bedarf an einem Produkt, so bestimmt der Kunde die Beschaffenheit, den Termin und den Preis, die seine

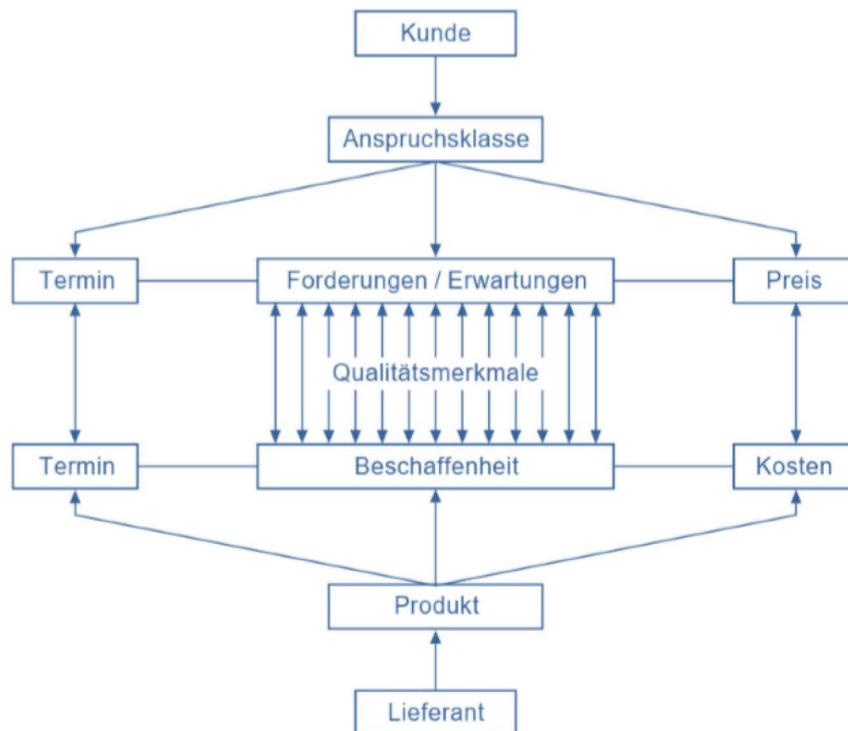


Abbildung 4.3: Spannungsdreieck nach Masing (Quelle: Masing 2014 <sup>[66]</sup>)

### Änderung der Kundenbedürfnisse

Sich ändernde Kundenbedürfnisse stellen die Automobilhersteller vor neue Herausforderungen. Kunden präferieren digitale Dienstleistungen für Kommunikation, Unterhaltung, Vernetzung und Produktivität bei vorausgesetzten klassischen Produktmerkmalen. Die Markenloyalität verliert dabei zunehmend an Bedeutung: Kunden neigen bei einem besseren Angebot von Technologie-Features zum Wechsel der Automarke.<sup>[67]</sup> Die Anforderungen an Infotainmentsysteme zur Erfüllung der Kundenerwartung im Fahrzeug sind auch ausschlaggebend für die Kundenzufriedenheit. Die Herausforderung für Hersteller besteht darin, den Kunden eine zufriedenstellende digitale Dienstleistung und Technologie bereitzustellen. Mittels digitaler Dienstleistungen kann somit eine zusätzliche Einnahmequelle erschlossen werden. Aufgrund von auftretenden Fehlern und Störungen während der Nutzungsphase mit funktionalen Beeinträchtigungen besteht ein nur beschränktes Vertrauen gegenüber den digitalen Dienstleistungen der Automobilhersteller. Im Gegensatz dazu weisen Google, Apple und

weitere Unternehmen der IT-Branche einen Vertrauensvorsprung beim Kunden auf. Dieser Vertrauensvorsprung resultiert aus den bekannten und bewährten Systemen der Rookies. Hinsichtlich der Nutzererfahrung oder des Nutzererlebnisses (User Experience) verlangen die Kunden eine einheitliche Darstellungsweise. Veraltete Systeme und Produkt-Features sind zu vermeiden. Die Standards im Bereich der Infotainmentsysteme und Kundenprodukte bzw. -gebrauchsgegenstände (Consumer Devices) werden von den Rookies festgelegt. Dieses Gebiet bildet seit mehreren Jahren unter anderem das Kerngeschäft von Apple und Google. Für die Automobilhersteller besteht ein dringender Nachholbedarf.<sup>[67]</sup>

Eine bedeutsame Herausforderung ist die Verknüpfung von traditionellen und digitalen Wertschöpfungsstufen. Die traditionellen Wertschöpfungsstufen der Automobilindustrie lehnen sich an die klassische Fahrzeugentwicklung, die Produktion und den Verkauf an. Die Wertschöpfung orientiert sich entlang der organisierten KIEFA-Struktur. Die digitalen Wertschöpfungsstufen sind den digitalen Geschäftsmodellen und Dienstleistungen zugeordnet.

#### 4.2.2 Ergebnisse Benchmarking-Analyse und Kundenzufriedenheit bezüglich digitaler Dienste

Fast alle traditionellen Automobilhersteller sowie neuen Anbieter bieten heutzutage den Kunden neue Mobility- sowie Connectivity-Services an. (s. Abb. 4.4) Basierend auf der Recherche gibt es eine Vielzahl neuer Begriffe auf dem Connectivity-Gebiet in der Automobilindustrie.<sup>[68] [69] [74]</sup> Die bekannten Automobilhersteller, neue Mobility-Service-Dienstleister sowie die weltweit erfolgreichen IT-Firmen wurden im Benchmark analysiert. Abbildung 4.4 zeigt einen Auszug der Automobilhersteller mit den vorrangigen digitalen Diensten: Remote-Zugriff, Statusabfragen und Unterstützungsfunktionen wurden bereits von mehreren OEMs umgesetzt.

Hersteller/Apps	Remote-Zugriff Funktionen						Statusabfragen Funktionen						Unterstützungsfunktionen		
	Remote-Parking	Remote Control	Remote Search	Remote Temperature	Remote Settings	Remote Monitoring	Digital Key	Information	Check	Geographic Monitoring	Maintenance	Liquid Level	Trip	Emergency	Evaluation
Mercedes me	x	x		x	x			x	x	x	x	x	x	x	
Car-Net			x	x		x		x	x	x	x	x	x	x	x
BMW Connected		x	x	x				x	x	x			x	x	
Audi MMI connect		x		x			x	x	x	x	x	x	x	x	
FordPass											x	x			
myOpel		x	x					x	x		x	x	x	x	
Skoda Connect			x			x		x	x	x	x	x	x	x	x
Toyota Service														x	
MY Renault														x	
Volvo On Call		x	x	x			x	x	x	x	x	x	x	x	x
.....															

Abbildung 4.4: Auszug der Funktionsübersicht nach Herstellern<sup>[68] [69]- [74]</sup> (Stand 03.03.2018)

Wie in Abb. 4.4 dargestellt, steht der amerikanische Hersteller Tesla mit über 4 Punkten an der Spitze, während die deutschen Premiummarken Audi, BMW und Mercedes relativ weit unten liegen. Die Säulen zeigen die direkte Online-Kundenbewertung und die Kundenzufriedenheit mit den Apps an. Abb. 4.5 veranschaulicht einen Vergleich der Kundenbewertung der Mercedes-me-App der Daimler AG mit der Tesla-App. Die blaue Kurve zeigt eine deutlich unterschiedliche Punkteverteilung.

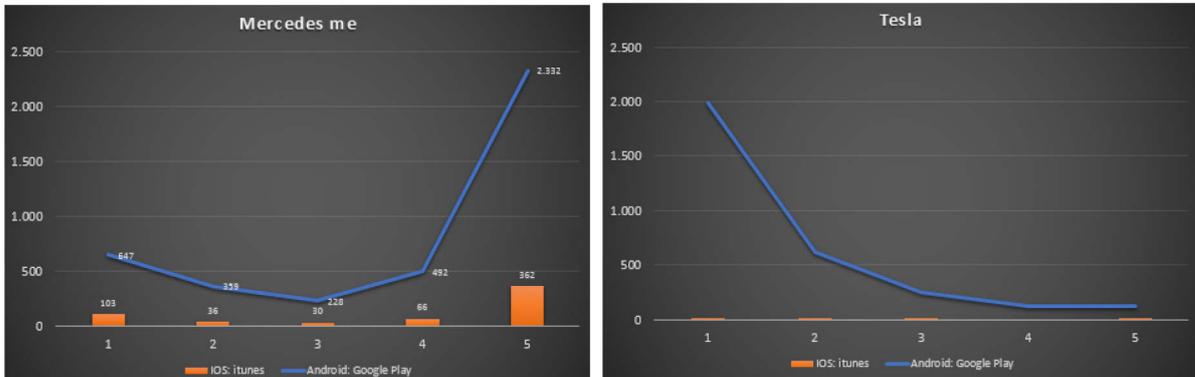


Abbildung 4.5: Vergleich Kundenbewertungen Mercedes-me- und Tesla-App (Eigene Darstellung, vgl. iTunes Store und Google Play)

Die nachfolgende Abb. 4.6 zeigt das Ranking basierend auf folgenden Kriterien an. (beliebte Automobilhersteller, innovative Automobilhersteller sowie Car Sharing aus verschiedenen Quellen im Jahr 2017)

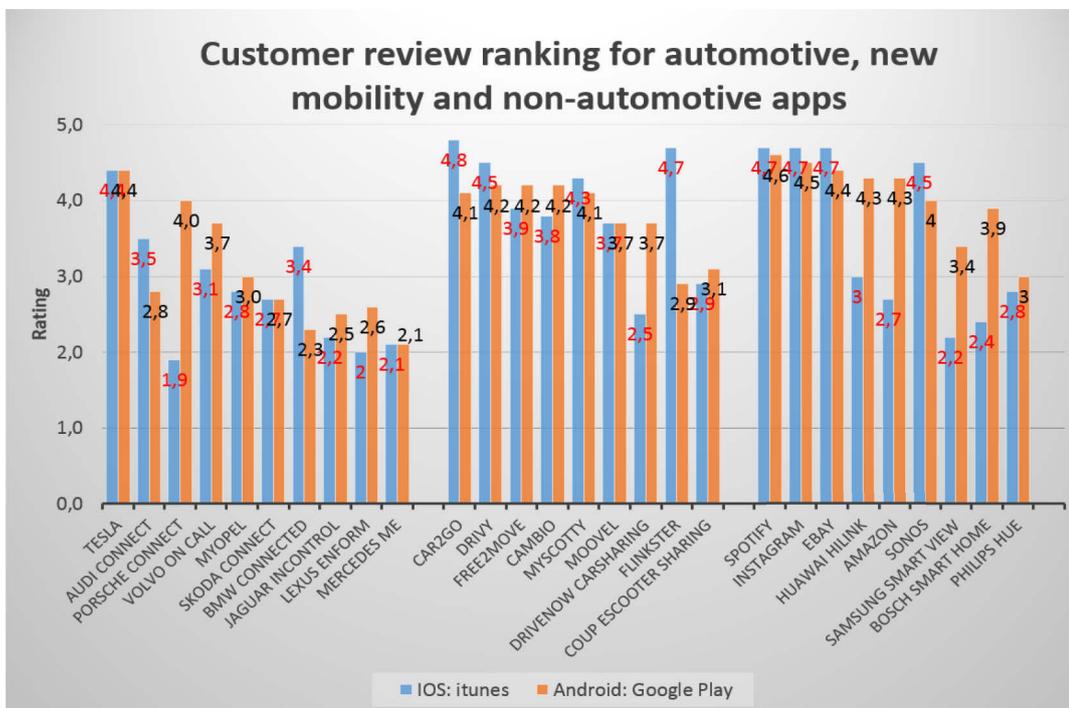


Abbildung 4.6: Ranking nach Automobilhersteller (Eigene Darstellung vgl. iTunes Store und Google Play)

*The driver of a software project is the customer. If the software doesn't do what they want it to do, you have failed.*<sup>[75]</sup>

Durch den Vergleich in Abb. 4.6 wird der Unterschied der Kundenbewertungen für verschiedene Apps der Automobilindustrie, Mobilitätsdienstleister und IT-Firmen deutlich. Im Vergleich mit der Nicht-Automotive-Branche stehen Automobilhersteller vor der Herausforderung, die Kundenbedürfnisse möglichst optimal zu erfüllen, um zukünftig weiterhin wettbewerbsfähige Fahrzeuge am Markt anbieten zu können und an den Marktpotenzialen für vernetzte Fahrzeuge und Funktionen zu partizipieren. Dabei ist die Erwartungshaltung der Kunden stark von dem mobilen Bereich geprägt, die ebenfalls diese Potenziale ausschöpfen möchten.

Bei einer gleichzeitig erhöhten Nachfrage nach individueller Mobilität ändern sich auch die Rahmenbedingungen drastisch, wodurch die neuen und intelligenten Angebote auf dem Markt an Bedeutung gewinnen werden. Auch die Automobilbranche muss sich dieser Änderungen annehmen, um in Zukunft als relevantes Mobilitätskonzept von Kunden wahrgenommen zu werden. In diesem Zusammenhang kam die Idee zur Vernetzung von Fahrzeugen auf. Diese Vernetzung von Fahrzeugen tragen einen entscheidenden Beitrag dazu bei, sowohl den Kundenanforderungen an ein modernes Fortbewegungsmittel gerecht zu werden als auch übergreifende Anforderungen von regulierenden Instanzen an zukunftsfähige Mobilitätslösungen zu erfüllen. Verschiedene Studien zeigen Ergebnisse, dass die Kundengruppen der Nutzer digitaler Dienste anders als die der traditionellen Fahrzeugbesitzer sind: Die neuen Kunden, die digitale Dienste heute nutzen, können wahrscheinlich kein eigenes Auto besitzen.<sup>[16]</sup>

#### Methode der User Journey Map

Die User Journey Map ist eine beliebte Methode aus dem Feld der User Experience, die überwiegend in Verbindung mit der Konzeptionierung neuer Produkte Anwendung findet. Die Methode veranschaulicht das Erlebnis eines Kunden entlang der gesamten Nutzung einer Dienstleistung. Die Map, die das Ergebnisse der Anwendung der Methode bildet, entsteht in der Regel durch die Zusammenarbeit vieler Beteiligter (Konzepte, Product Owner, UX-Designer, Marketing-Verantwortliche etc.). Es gibt kein standardisiertes Format zur Gestaltung der Journey. Der entscheidende Wert der User Journey liegt in der Kundenperspektive sowie dem Einbeziehen aller Teammitglieder aus den verschiedenen Disziplinen, sodass eine Vielzahl an kreativen Ideen zur kundenfreundlichen Gestaltung der einzelnen Nutzungsschritte gesammelt wird und differente Sichtweisen eingenommen werden.<sup>[76] [77]</sup>

### **4.3 Fehler und Fehlerarten der digitalen Dienste**

Im Zuge der vorliegenden Dissertation wurden die bekannten Fehlerarten der digitalen Dienste recherchiert, zusammengefasst und neu klassifiziert. Die klassische Qualitätssicherung in der Automobilindustrie ist oft hauptsächlich auf das Finden von bereits existenten Fehlern ausgerichtet. Im Idealfall findet sie die Fehler durch den Einsatz mannigfaltiger Techniken wie statische Codeanalysen, Stichproben, Reviews und Testen so früh wie möglich nach dem Auftreten.

Der aktuelle Schadteilanalyseprozess in der deutschen Automobilindustrie fokussiert sich auf die von Kundenfahrzeugen getauschten fehlerhaften Fahrzeugkomponenten. Die Software-Diagnose wurde nur oberflächlich beschrieben und es gibt dazu aktuell noch keine konkreten Maßnahmen oder Prozessbeschreibungen.<sup>[78]</sup> Die aktuellen Qualitätsanalyse-Strukturen und -Prozesse bilden die Anforderungen nicht mehr ausreichend ab. Um auch den zukünftigen Qualitätsansprüchen der Kunden und einer nachhaltigen Sicherstellung der Feldqualitätsanalyse gerecht zu werden, sind neue Fehleranalysestrategien erforderlich.

Wie in Kapitel 3.4.1 definiert, ist ein Fehler die Nichterfüllung von verfügbaren funktionalen Produktmerkmalen sowie die Abweichung von Kundenerwartungen.

Im folgenden Kapitel wird erst der Status quo des Fehlerabstellprozesses in der Automobilindustrie und wissenschaftliche Methoden vorgestellt. Die aktuellen Fehlermanagement Methoden in der heutigen Automobilindustrie werden analysiert und vorgestellt. Abschließend folgen kritische Fragen an aktuelle Methoden bzw. Datenquellen der klassischen Autokomponenten.

### 4.3.1 Status quo des Fehlerabstellprozesses

Das Fehlermanagement kann in das Aachener Qualitätsmanagement-Modell eingeordnet werden. Schmitt/Pfeifer<sup>[79]</sup> definieren zu diesem Zweck einen rollenbasierten Referenzprozess für das Fehlermanagement. Der im Folgenden als Fehlerabstellprozess bezeichnete Referenzprozess ist in Abbildung 4.7 dargestellt.<sup>[80]</sup>

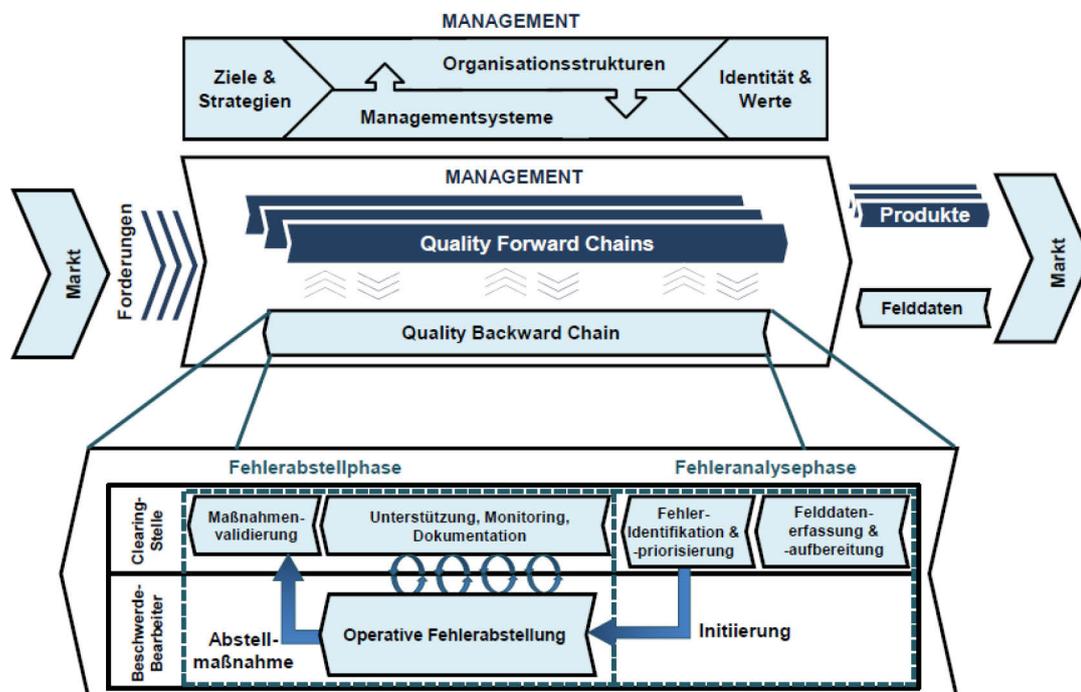


Abbildung 4.7: Fehlerabstellprozess im Aachener Qualitätsmanagementmodell<sup>[80]</sup>

Die Verantwortung für den Prozess wird auf zwei Rollen verteilt: die Clearing-Stelle und den Beschwerdebearbeiter. In der ersten Phase, der Fehleranalysephase, führt die Clearing-Stelle die Erfassung von Fehlerdaten aus den Feldqualitätssensoren, die Fehleraufbereitung und -identifizierung sowie die Fehler-Priorisierung durch. Die Clearing-Stelle initiiert die Fehlerabstellphase und den operativen Fehlerabstellprozess, der von den Beschwerdebearbeitern durchgeführt wird.

In der aktuellen deutschen Automobilbranche wird zur Fehleranalyse der Schadteilanalyseprozess verwendet, der im VDA-Band seit dem Jahr 2009 „Schadteilanalyse Feld“ beschrieben wird. In Abbildung 4.8 ist zu erkennen, dass dieser aus zwei Hauptbestandteilen besteht: zum einen aus der Befundung und zum anderen aus dem No-Trouble-Found-Prozess (NTF-Prozess).<sup>[78]</sup>

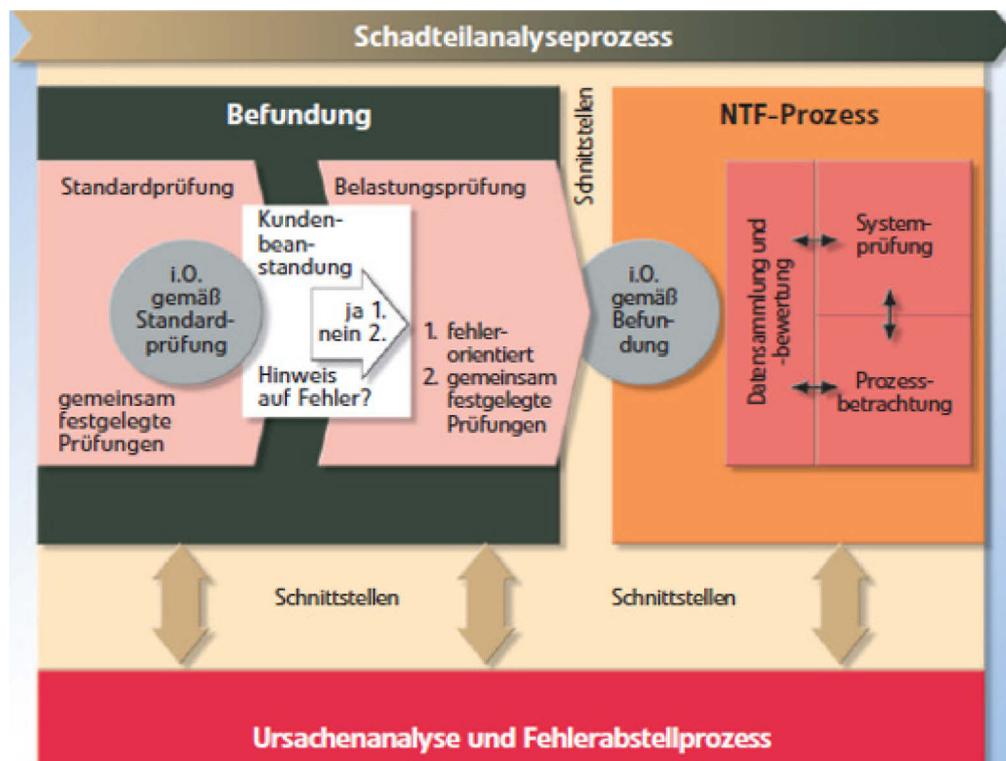


Abbildung 4.8: VDA Blauer Band, Schadteilanalyseprozess <sup>[78]</sup>

Die auftretenden Fehler werden von den Automobilherstellern analysiert, um die Fehlerursachen zu ermitteln und diese in den Fehlerabstellprozess einfließen zu lassen. Wie in Kapitel 3.3.1 definiert, bestehen digitale Dienste aus Hardware, Software und viel mehr. Bei digitalen Diensten geraten die Fehleranalysemethoden der Automobilhersteller jedoch an ihre Grenzen, da diese sich hauptsächlich auf Hardware-Probleme beziehen. Um auch Software-Fehler analysieren zu können, werden Methoden aus der IT-Branche benötigt. Die relevanten Unterschiede der ganzheitlichen Fehleranalyseprozesse für Hardware-Bauteile und digitale Dienste werden in Abbildung 4.9 dargestellt.

Die Anpassung der Rolle des Qualitätsmanagements im Feld der digitalen Dienste wird in Kapitel 5 weiter erforscht. Die Weiterentwicklung des Fehleranalyseprozesses wird im Laufe dieser Doktorarbeit bearbeitet.

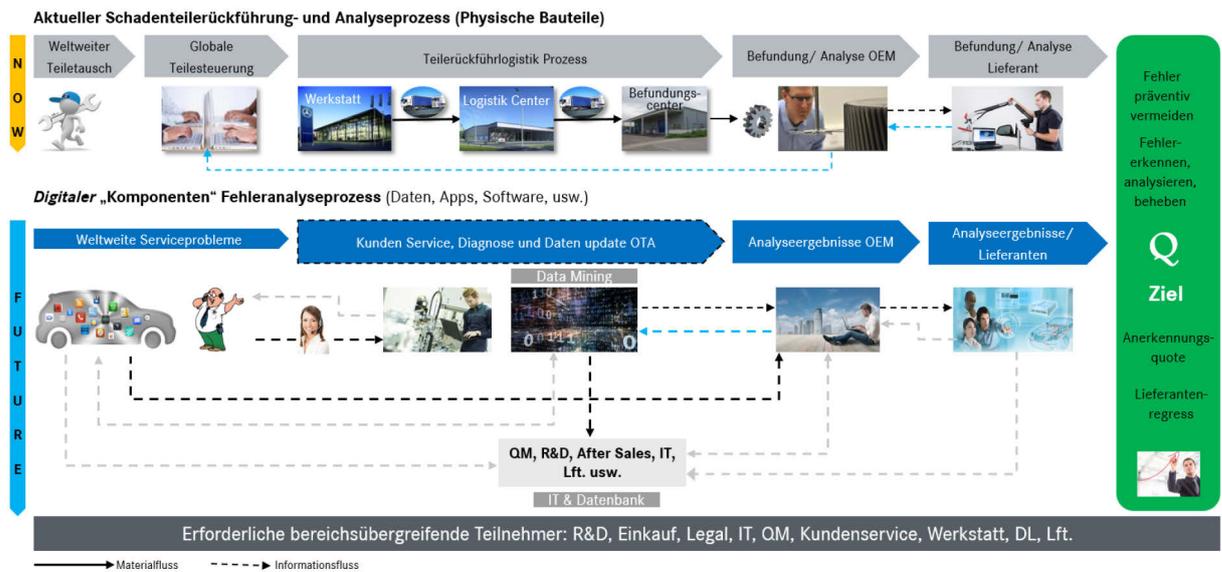


Abbildung 4.9: Vergleich des ganzheitlichen Schadenteile-Rückführungsprozesses und desjenigen digitaler ‚Komponenten‘ (eigene Darstellung)

### 4.3.2 Fehlermanagement-Methoden

Im folgenden Abschnitt werden einige der grundlegenden Methoden vorgestellt, die im Bereich des Fehlermanagements verwendet werden. **Quality Function Deployment (QFD)**

Das QFD ist eine weitverbreitete Methode zur Qualitätssicherung in der Produktentwicklung. Die Methode dient der Ermittlung der Kundenanforderungen (extern & intern) an ein Produkt, um daraus Entwicklungsanforderungen im Hinblick auf die Qualität abzuleiten. Die verschiedenen Bereiche eines Unternehmens verwenden unterschiedliche Fachsprachen und legen ihren Fokus auf unterschiedliche Produkthanforderungen, die sich aufgrund ihrer Spezialisierung ergeben. Das QFD hilft, zwischen den verschiedenen Unternehmensbereichen (sowie zwischen Kunden und Lieferanten) zu vermitteln und so die verschiedenen Aktivitäten im Entwicklungsprozess konsistent auf den Markt auszurichten.<sup>[79]</sup>

Die Methode wird in vier Phasen unterteilt, wobei mit jeder Phase die Kundenanforderungen konkretisiert werden. Zu Beginn werden die Kundenanforderungen mittels Workshops, Interviews und Marktanalysen identifiziert. Anschließend findet die Planung des Produktes statt, indem die Kundenanforderungen in technische Funktionsmerkmale mittels Lasten- und Pflichtenheft umgesetzt werden.<sup>[79]</sup> In der dritten Phase werden die einzelnen Bauteile und Gruppen geplant. Zuletzt werden die betrieblichen Abläufe und Prozesse geplant sowie Prüfpunkte und Prozessparameter festgelegt. Das zentrale Formblatt wird als House of Quality bezeichnet, das als Verständigungsmittel zwischen den einzelnen Abteilungen dient. Es stellt die Kundenanforderungen, die Ableitung der Qualitätsmerkmale, die Festlegung der Zielgrößen, die Wechselwirkungen zwischen den Qualitätsmerkmalen und einen Leistungsvergleich mit den Wettbewerbern dar. Die Vorteile dieser Methode sind die

Nachvollziehbarkeit der Produkteigenschaften und der damit verbundenen Kundenwünsche. Die Methode deckt allerdings Probleme häufig nur auf, ist aber nicht in der Lage, die Probleme zu lösen, sodass andere Methoden zur Ergänzung verwendet werden müssen.<sup>[79]</sup>

### **1. Die Fehlermöglichkeits- und Einflussanalyse**

Die Fehlermöglichkeits- und Einflussanalyse (FMEA) ist eine der am häufigsten verwendeten Methoden im Bereich des Fehlermanagements. Die FMEA ist eher dem Bereich der Fehlerprävention zuzuordnen. Das Ziel der FMEA ist es, ausgehend von einem Fehler die Fehlerfolgen, das potenzielle Risiko sowie geeignete Maßnahmen einzuschätzen bzw. abzuleiten. Die FMEA wird meist in drei Arten gegliedert: die System-, Konstruktions- und Prozess-FMEA, die zu unterschiedlichen Zeitpunkten während des PEP angewendet werden.

Der Ablauf sowie die Richtlinien der FMEA sind in der DIN EN 60812 geregelt. Der Ablauf einer FMEA beginnt mit dem Festlegen der Teamzusammensetzung, der Terminplanung und der inhaltlichen Vorbereitung in Bezug auf die Aufgabenstellung. In der Durchführungsphase werden zunächst potenzielle Fehler, Fehlerquellen, -folgen sowie Ursachen und entsprechende Maßnahmen ermittelt bzw. analysiert. Anschließend werden den potenziellen Fehlern Auftretens-, Bedeutungs- und Entdeckungszahlen zugeordnet. Die Multiplikation dieser Zahlen ergibt die Risikoprioritätszahl (RPZ). Diese gibt Auskunft über das Ausmaß des jeweiligen Fehlers, sodass die Relevanz der einzelnen Fehler beurteilt werden kann. Die Auftretens-, Bedeutungs- und Entdeckungszahlen werden auf Basis der Unternehmensrichtlinien subjektiv eingeschätzt. Aus der FMEA können Maßnahmen abgeleitet werden, die durch Terminverfolgung und Erfolgskontrolle überprüft werden.

### **2. Fehlerbaumanalyse**

Ähnlich wie bei der FMEA wird auch bei der Fehlerbaumanalyse (FTA) ein ganzheitliches Fehlerbild ermittelt, jedoch wird hier der Top-Down-Ansatz gewählt. Dem FTA kann sowohl ein präventiver als auch ein reagierender Ansatz zugrunde liegen. Auf Basis des systematischen Einsatzes der Fehlerbaumanalyse werden die Teilsysteme sowie die Einflüsse auf die Funktionsfähigkeit abgebildet und über Boolesche Verknüpfungen (UND, ODER, NICHT) in Verbindung gebracht (DIN 25424).

Zu Beginn der FTA wird das zu untersuchende System analysiert. Hierzu werden Informationen wie Abhängigkeiten, Eintrittswahrscheinlichkeiten und Parameter der einzelnen Bestandteile des Gesamtsystems (bspw. Untersysteme, Bauteile, Komponenten) gesammelt und miteinander verknüpft. Das Ziel ist es, mithilfe dieser Informationen die Abhängigkeiten in einem Fehlerbaum darzustellen. Anhand der FTA lassen sich Ausfallwahrscheinlichkeiten (quantitativ) eines Systems berechnen oder nur Abhängigkeiten (qualitativ) darstellen. Ähnlich wie bei der FMEA, können abschließend Maßnahmen zur frühzeitigen Fehlererkennung und zur Abstellung von Fehlerursachen abgeleitet werden.

Der strukturierte Aufbau und Ablauf dieser Methode macht eine Verknüpfung mit bzw. eine Umsetzung als Software möglich. Diese Methode stößt bei komplexen Systemen aufgrund mangelnder Übersichtlichkeit allerdings an ihre Grenzen.

### 3. Six Sigma

Die Verbesserung der Qualität von Produkten und Prozessen bei gleichzeitiger Kostenreduktion, Umsatzsteigerung und höherer Kundenzufriedenheit ist das grundlegende Ziel dieser Methode. Die Six-Sigma-Methode stellt eine klar strukturierte Vorgehensweise sowie Methoden und Tools für die Verbesserung von Prozessen im Unternehmen zur Verfügung und orientiert sich dabei an dem DMAIC-Zyklus (S. Abb. 4.10). Der Zyklus strukturiert das Vorgehen in fünf aufeinanderfolgende Phasen, in der jeweils bestimmte Methoden und Werkzeuge zur Verfügung gestellt werden. Die Methoden und Werkzeuge sind in einer Toolbox zusammengefasst, die hauptsächlich aus Tools anderer Statistik- und QM-Methoden besteht. Six Sigma zählt zu den statistischen Verfahren. Die bekannteste Bemessungseinheit in Six Sigma ist dpmo (defects per million opportunities). Zur Ermittlung des dpmo-Wertes ist es nötig, die qualitätskritischen Attribute bzw. Merkmale eines Produktes oder Prozesses zu bestimmen. Diese Kenngröße gibt an, mit welcher Wahrscheinlichkeit ein Merkmal innerhalb der zuvor für dieses Merkmal festgelegten Spezifikationsgrenzen um einen bestimmten Zielwert liegt.

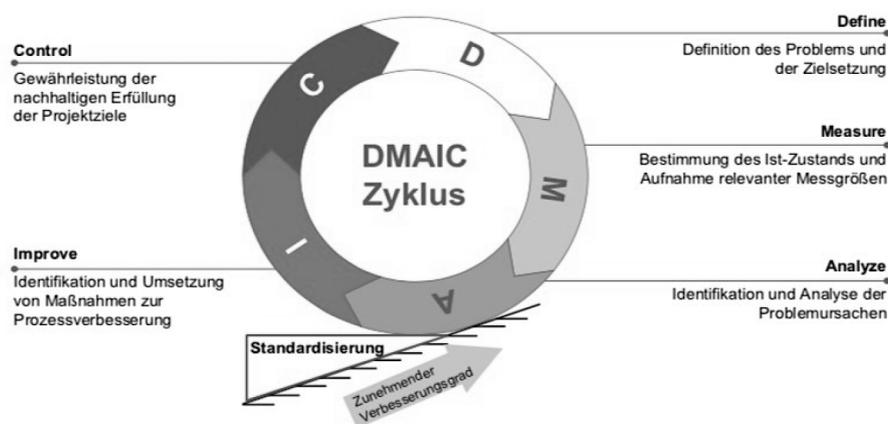


Abbildung 4.10: Six-Sigma-DMAIC-Zyklus

Six Sigma kann in drei unterschiedlich tiefgehenden Stufen im Unternehmen eingeführt werden. Die leichteste Stufe bezieht sich lediglich auf das Einführen der Toolbox. Das Verbesserungsprogramm dient der Integration von Six Sigma in ausgewählten Unternehmensbereichen. Die unternehmensweite Strategie stellt die höchste Stufe dar; dabei setzt das gesamte Unternehmen Six Sigma um. Das Einführen dieser Methode ist aufwendig und kann langwierig sein. Daher scheitern besonders kleine Unternehmen oft an ihrer Umsetzung.

### 4.3.3 Kritik an Fehlermanagement-Methoden

Die beschriebenen Methoden unterscheiden sich erheblich im Hinblick auf Komplexität, Aufwand und Eignung für das Fehlermanagement. Fast alle Methoden legen den Fokus auf präventive Fehlervermeidung. Das ergibt im Hinblick auf die Verteilung der Fehlerursachen und Fehlerkosten Sinn, jedoch ist eine schnelle Fehlerbehebung aus Kundensicht und damit ein aktives Fehlermanagement mindestens genauso relevant. Lediglich der 8D-Report ist ausschließlich für das Beheben von bereits geschehenen Fehlern geeignet. Dieser folgt jedoch einem starren, sequenziellen und wenig agilen Ablauf, der für ein schnelles Agieren besonders im Bereich von schnelllebigen digitalen Produkten und Apps in der Regel nicht optimal ist.

Der 8D-Prozess ist ursprünglich für die Fehlerbehebung physikalischer Produkte konzipiert. Dabei wird davon ausgegangen, dass der Entwicklungsprozess ab einem bestimmten Punkt abgeschlossen ist und anschließendes Testen bzw. der Betrieb getrennt voneinander zu betrachten sind. Dies ist bei Software jedoch nicht der Fall. Es bedarf daher eines Ansatzes, wie präventive Maßnahmen in der agilen Software-Entwicklung so genutzt werden können, dass reaktive Maßnahmen effektiv und effizient auf die agile Entwicklung abgestimmt werden.

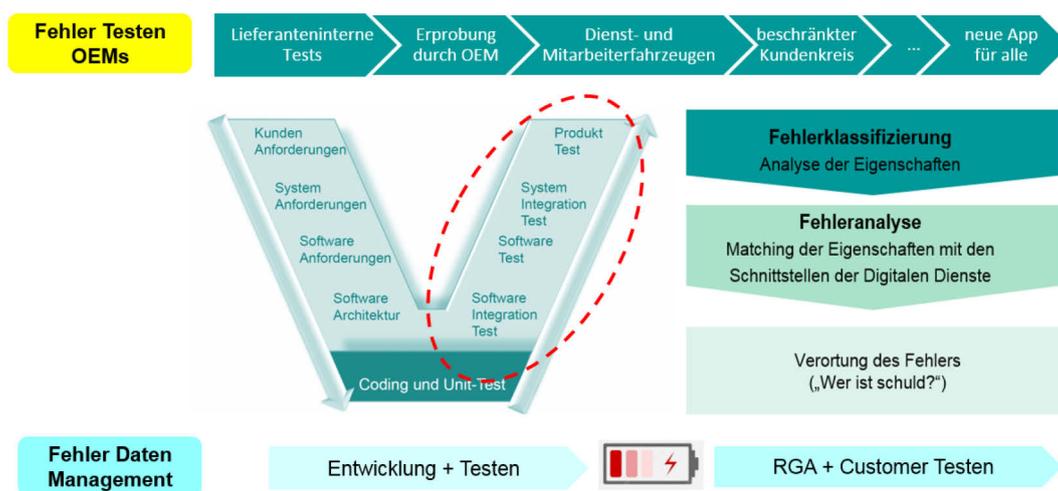


Abbildung 4.11: Fehler Klassifizierungsprozess für Software in der Automobilindustrie (Eigene Darstellung)

Wie Abbildung 4.11 zeigt, sind die 4 Kernschritte des aktuellen Fehlerabstellprozesses in der Automobilindustrie die folgenden: Fehler definieren, erkennen, klassifizieren und Ursachen analysieren. Dabei kann die vom VDA eingefügte 8D-Methode sowohl zu Informations- als auch zu Dokumentationszwecken eingesetzt werden.<sup>[37]</sup>

Mit der Tendenz der Digitalisierung in der heutigen Automobilindustrie spielt Agilität eine bedeutsame Rolle im Connectivity-Feld. Um auf Feedback der Kunden schnell zu reagieren sowie Probleme rasch zu beheben im Sinne von ‚Customer (Feedback) First‘, ist der aktuelle 8D-Prozess für Software sowie digitale Dienste nicht geeignet. Automotive-SPICE<sup>[81]</sup> Software-Entwicklungsprozesse hängen stark von dem Konzept

der Veröffentlichung der App bzw. Software ab. Eine schrittweise Einführung der Software vor der Veröffentlichung der App ist essenziell. Die Zielsetzung ist es dabei, schwerwiegende und häufige Fehler nicht erst beim Kunden zu entdecken, sondern frühzeitig vor dem ‚Go-Live‘ (bzw. SOP in der Automobil Industrie) zu testen und zu beheben.

In dieser Doktorarbeit wird ein Teil des Gesamtprozesses analysiert (siehe Abb. 4.12). Hier gibt es eine enge Zusammenarbeit mit verschiedenen Ressourcen. Details dazu werden in den Kapiteln 5 und 7 vorgestellt.

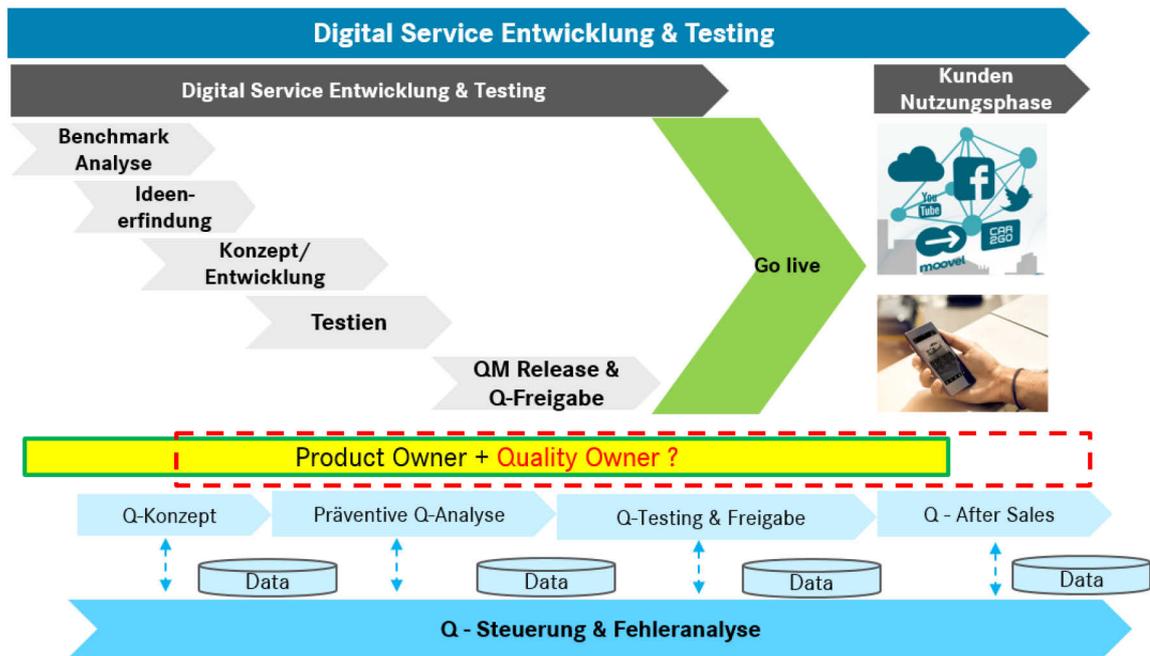


Abbildung 4.12: Integration des Fehler-Analyseprozesses für digitale Dienste in der Automobilindustrie (Eigene Darstellung)

Wie schon verdeutlicht wurde, ist ein digitaler Dienst eine komplexe Plattform und Kombination aus mehreren Usern. Demnach ist es auch nicht einfach, die Fehler einer solchen Plattform bezüglich Hardware und Software zu analysieren. Sowohl die Methoden der Automobilbranche als auch die der IT-Branche sind nicht ausreichend, um Fehler von digitalen Diensten untersuchen zu können. Diesbezüglich werden in dieser Arbeit eine Fehlerklassifizierung und ein Konzept zur Fehleranalyse und Bewertung von digitalen Diensten erarbeitet.

Vollständige Daten sind eine Voraussetzung zur Fehleranalyse. In der folgenden Darstellung sind die bekannten Datenquellen in der heutigen Automobilbranche eingezeichnet. Für digitale Dienste sind die Datenquellen erweitert; die Kundendaten sind z. B. keine physischen Bauteile und ein neues Update wird per Over-the-Air (OTA) ohne Werkstattbesuch aktualisiert.<sup>[80]</sup>

Eine transparente bereichsübergreifende Datenbank ist eine Basis für das zukünftige Qualitätsmanagement digitaler Dienste.

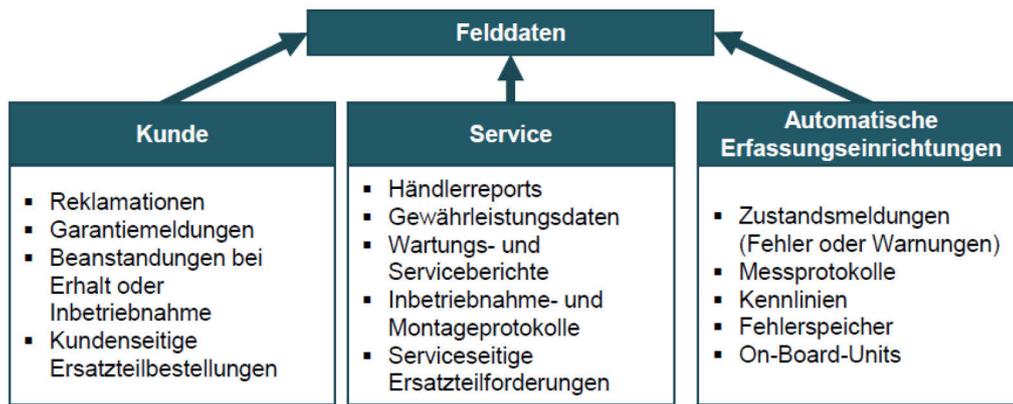


Abbildung 4.13: Quellen der Felddaten klassische Komponenten [80]

## 4.4 Umgang mit Reklamationen

Obwohl der standardisierte Reklamationsprozess in der Automobilindustrie zwischen Lieferanten und OEM, und damit auf einer B2B-Ebene stattfindet, wird im nächsten Abschnitt die allgemeine Ablaufbeschreibung des B2C-Beschwerdemanagementprozesses wiedergegeben. Ursächlich dafür ist der bereits thematisierte Mangel an entsprechender Fachliteratur. Einige Prozessschritte besitzen zudem eine beziehungsübergreifende Relevanz.

An dieser Stelle der Arbeit waren ursprünglich Ausführungen zu dem entsprechenden Reklamationsprozess in der Software-Branche vorgesehen. Allerdings ließen sich nach eingehender Recherche keine diesbezüglichen Quellen finden. Der wahrscheinlichste Grund für diesen Umstand besteht darin, dass es keine übergeordnete Instanz wie den VDA gibt, die entsprechenden Richtlinien für diesen Industriezweig entwickelt und veröffentlicht. Folglich ist von einem Zustand wie in der Automobilindustrie im Jahr 2009 auszugehen. Dieser ist gekennzeichnet durch individuelle Lösungen jeder Firma.

### 4.4.1 Beschwerdemanagement (BM)

Durch die zuvor thematisierte Verschiebung der Wertschöpfungsanteile ist gleichsam eine Zunahme der durch Lieferanten verursachten Fehler zu beobachten. Folglich tritt das Beschwerdemanagement in den Fokus der Betrachtung. Es bildet den übergeordneten Rahmen der vorliegenden Ausarbeitung. In diesem Zusammenhang werden zunächst die zum Verständnis notwendigen Begriffe definiert und in den Betrachtungskontext eingeordnet. Darauf folgend werden, wie bereits zuvor, Ausarbeitungen hinsichtlich der Bedeutung sowie des Prozessablaufs vorgenommen.

Mit der Beschwerde und der Reklamation existieren zwei häufig synonym verwendete Begriffe. Die Literatur empfiehlt jedoch eine Unterscheidung beider Ausdrücke. Demnach sind unter einer Beschwerde „alle Unzufriedenheitsartikulationen von Konsumenten zu verstehen, die darauf gerichtet sind, in Verbindung mit Konsumgütern und Konsumgüteranbietern wahrgenommene Probleme (Konsumentenprobleme) zu lösen.“<sup>[82]</sup> Kurz gesagt: „Beschwerden sind Äußerungen von Unzufriedenheit.“<sup>[83]</sup>

Reklamationen stellen dabei einen Sonderfall bzw. eine Teilmenge der Beschwerden dar. Mit ihnen beabsichtigt der Kunde eine rechtliche Entschädigung für das erhaltene Produkt oder die erhaltene Dienstleistung (rechtliche Gewährleistung). Somit können Reklamationen allein in der Phase nach dem Abschluss des Geschäfts auftreten, wohingegen Beschwerden bereits vor oder während der Transaktion an ein Unternehmen gerichtet werden können. Kapitel 7.5 beschäftigt sich folglich mit Reklamationen.

Unter dem bereits mehrfach angesprochenen Kunden ist nicht zwangsläufig eine Einzelperson im Sinne eines Verbrauchers zu verstehen. Vielmehr existiert eine in Abb. 4.14 dargestellte Struktur. In diesem Zusammenhang wird zwischen B2B und B2C unterschieden. In einer B2B-Beziehung ist der OEM folglich ein Kunde des Lieferanten, wohingegen der Verbraucher in einer B2C-Beziehung Kunde des OEM ist.



Abb. 4.14: Partner im Produktentstehungsprozess (Eigene Darstellung nach <sup>[84]</sup>)

Dieser Darstellung kommt eine hohe Bedeutung zu, da sich die Beschwerdeforschung fast ausschließlich auf B2C-Beziehungen konzentriert. Dies ist kritisch zu bewerten, da der Kunde im B2B-Geschäft bedeutend abhängiger von seinem Lieferanten ist, als es in B2C-Beziehungen der Fall ist. Hinzukommt, dass der eigentliche Kauf bzw. Verkauf eines Produktes oder einer Dienstleistung nur einen Bruchteil der gemeinsamen Beziehung darstellt. Zusätzlich übersteigen die monetären Werte der B2B-Geschäfte, trotz eines kleineren Kundenmarktes mit weniger Transaktionen, die der B2C-Geschäfte um ein Vielfaches. Dies führt letztlich zu dem Schluss, dass „die professionelle Abwicklung der Kundenbeschwerde in B2B-Geschäftsbeziehungen deutlich wichtiger als im B2C Bereich“ ist.

Das BM als übergeordneter Prozess besaß in der deutschen Literatur vormals den Namen Beschwerdepolitik. Unterdessen wurde der erstgenannte Begriff weitestgehend aus dem Amerikanischen übernommen. Er beinhaltet die „Planung, Durchführung und Kontrolle aller Maßnahmen [...], die ein Unternehmen im Zusammenhang mit Beschwerden ergreift.“ Aufgrund der Zweiteilung des BM ist dieses sowohl dem Customer-Relationship-Management (CRM) als auch dem Qualitätsmanagement zurechenbar.

#### 4.4.2 Ablauf 8D-Methode

Die Ziele der 8D-Methode sind eine sofortige Abstellung des Fehlers sowie eine Beseitigung der Fehlerursache, um ein weiteres Auftreten des Fehlers auszuschließen. Diese Methode wird häufig in der Automobilindustrie angewendet, um Fehler zwischen Kunden und Lieferanten (sowohl intern als auch extern) zu beseitigen. Die 8D-Methode

umfasst acht Disziplinen, die im Folgenden kurz erläutert werden. Als Vorlage dient dabei der VDA-Band 4: 8D-Methode, Langversion.

In diesem Abschnitt wird die 8D-Methode als das Kernelement des standardisierten Reklamationsprozesses sowie des Fehleranalyseprozesses zwischen OEM und Lieferanten in der Automobilindustrie veranschaulicht. Zu diesem Zweck werden wiederum zunächst der Hintergrund und die Bedeutung der Vorgehensweise verdeutlicht, ehe eine schrittweise Prozessdarstellung vorgenommen wird.

Der Ausdruck 8D bezeichnet dreierlei Dinge: zum ersten die Problemlösung gemäß einer Standardmethode; an zweiter Stelle ist der eng damit verbundene Problemlösungsprozess zu nennen; den dritten Aspekt bildet der 8D-Report als Berichtsform der Methode bzw. des Prozesses.<sup>[85]</sup> Unter 8D sind dabei die acht verschiedenen Disziplinen im Vorgehensmodell zu verstehen, die den Anwender in strukturierter Weise durch den Problemlösungsprozess leiten.<sup>[86]</sup>

Innerhalb des beschriebenen Reklamationsprozesses nimmt das 8D-Verfahren die Rolle des Problemlösungswerkzeugs ein.<sup>[85]</sup> Damit stellt es einen festen Bestandteil des Gesamtablaufs dar.<sup>[87]</sup> In der Automobilindustrie hat es sich als Standard etabliert.<sup>1[86]</sup> Die Grundvoraussetzung für den Einsatz des Verfahrens besteht im Vorhandensein einer Problemstellung mit unbekannter Ursache.<sup>[85]</sup> Neben der nachhaltigen Problemlösung ist die Anwendung von Sofortmaßnahmen erforderlich.<sup>[86]</sup> Zusätzlich sollten eine umfassende und verständliche Faktensammlung über die festgestellte Abweichung vorliegen und die Lösung des Problems die Fähigkeiten eines interdisziplinären Teams benötigen. Aufgrund dieser Einschränkungen ist der Einsatz der 8D-Methodik erst ab komplexeren Problemen zielführend.<sup>[85]</sup>

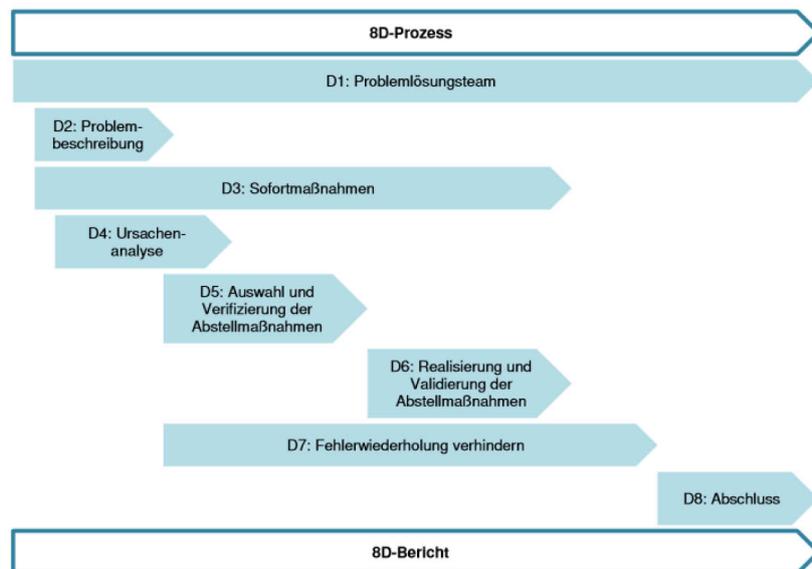


Abbildung 4.15: Prozessdarstellung 8D <sup>[85]</sup>

Die folgenden Ausführungen zu den einzelnen Disziplinen sollen mit den Abbildungen im Anhang 7 lediglich ein Grundverständnis über die Methodik vermitteln und erheben keinen Anspruch auf Vollständigkeit.<sup>[85]</sup>

In der ersten Disziplin (D1: Problemlösungsteam) wird das interdisziplinäre Team zusammengestellt. Neben den Teammitgliedern (Anzahl und Zusammensetzung problemspezifisch) wird ein Sponsor (Verantwortung für benötigte Ressourcen) sowie ein Teamleiter (Methodenkompetenz) bestimmt. Neue Erkenntnisse während des nachfolgenden Prozesses können in diesem Zusammenhang einen direkten Einfluss auf die Zusammensetzung der Gruppe nehmen. Von entscheidender Bedeutung für diesen Schritt ist, dass das Team zum Zugriff auf sämtliche benötigte Informationen berechtigt wird und die Kommunikationswege klar definiert sind.

Im anschließenden Schritt (D2: Problembeschreibung) erfolgt das Verständnis über den Gesamtzusammenhang des vorliegenden Problems. Es wird eine simple, aber gleichzeitig möglichst genaue und nachvollziehbare Aussage über die Abweichung angestrebt, ohne bereits über die Ursachen zu mutmaßen. Dabei hat sich die Anfertigung von Soll-Ist-Vergleichen auf Basis von vereinbarten Spezifikationen (Zeichnungen, Stücklisten etc.) bewährt. Zu diesem Zweck ist es notwendig, dass alle benötigten Informationen, z. B. über die Art und Anzahl betroffener Bauteile, vorliegen und aufbereitet sind. Zusätzlich ist eine Untersuchung potenzieller Auswirkungen auf vergleichbare Produkte oder Prozesse zu prüfen. Da die Ergebnisse dieses Schrittes den Input für die nachfolgenden darstellen, ist eine gewissenhafte Durchführung unabdingbar. So würde eine unvollständige Beschreibung unter Umständen zu einer fehlerhaften Ursachenanalyse und/oder unwirksamen Maßnahmen führen.

Während der dritten Disziplin (D3: Sofortmaßnahmen) wird auf die Beseitigung des aufgetretenen Problems beim Kunden abgezielt. Bei Software-Problemen kann beispielsweise auf ein Backup oder die Vorgängerversion zurückgegriffen werden, wohingegen bei Hardware-Problemen eine Liefersperre und ein sofortiges Aussortieren der fehlerhaften Bauteile angestrebt werden kann. Die getroffenen Maßnahmen müssen in ihrer Wirksamkeit bestätigt werden und bleiben danach bis zum Abschluss der sechsten Disziplin bestehen. Während dieses Schrittes wird noch keine nachhaltige Behebung des Fehlers, sondern lediglich eine kurzfristige Symptombeseitigung angestrebt. Es ist dabei von Bedeutung, dass die gesamte Lieferkette betrachtet und bei Bedarf informiert wird. Neben der Beseitigung beschädigter Elemente ist die Sicherstellung der Lieferfähigkeit während dieses Schrittes von hoher Relevanz. Häufig resultiert dies für den Hersteller (von Hardware-Komponenten) in einer 100%-Prüfung, Zusatzschichten und damit deutlich erhöhten Herstellungskosten. In der Automobilindustrie wird die dritte Disziplin in den meisten Fällen in enger Abstimmung mit dem Kunden durchlaufen. Unter Umständen muss dieser die im Zuge der Sofortmaßnahmen geänderten Prozesse freigeben. Letztlich wird das Ziel verfolgt, dass der Kunde am Ende von D3 nicht länger von den Auswirkungen des Problems betroffen ist.

Im vierten Schritt (D4: Ursachenanalyse) werden die Hintergründe des beschriebenen Problems ermittelt. Unterschieden wird dabei zwischen der Ursache des Auftretens und der Nichtentdeckung des Problems. Eine bewährte Methode der Ursachenforschung besteht im Ishikawa-Diagramm. An dessen Ende steht eine Sammlung von möglichen

Begründungen, von denen die wahrscheinlichsten näher, z. B. mittels der 5-Why-Methode, untersucht werden sollten. Die ermittelten Ursachen müssen in der Folge durch Tests bzw. Versuche als die tatsächlichen Grundursachen bestätigt werden. In diesem Schritt ist die richtige und vor allem vollständige Ermittlung der Problemhintergründe von höchster Bedeutung.

Im fünften Schritt (D5: Auswahl und Verifizierung der Abstellmaßnahmen) erfolgt die Erarbeitung von Lösungen für die in D4 ermittelten und verifizierten Grundursachen (Auf-treten und Nichtentdeckung). Am Ende dieser Disziplin steht ein mit Terminen, Ressourcen und Zuständigkeiten hinterlegter Aktionsplan. Die Umsetzung bzw. die Einführung der Maßnahmen findet in diesem Schritt noch nicht statt. Unter Umständen ist an dieser Stelle die Berücksichtigung von zurückliegenden und ähnlichen Fehlerursachen hilfreich. Die Voraussetzung dafür ist allerdings, dass für diese bereits Maßnahmen implementiert und in ihrer Wirksamkeit bestätigt wurden. Weiterhin sind im Zuge einer Risikobewertung mögliche Wechselwirkungen mit bestehenden Prozessen zu berücksichtigen. Die erdachten Abstellmaßnahmen müssen in ihrer Effektivität durch Versuche bestätigt werden und sollten mit vertretbarem wirtschaftlichem Aufwand umsetzbar sein (Berücksichtigung der Effizienz). Am Ende der fünften Disziplin muss das Team von der nachhaltigen Problembeseitigung überzeugt sein, sobald die erdachten Maßnahmen umgesetzt wurden.

Die sechste Disziplin (D6: Realisierung und Validierung der Abstellmaßnahmen) beinhaltet die Umsetzung des in D5 entwickelten Aktionsplans. Auch in diesem Schritt findet eine Wirksamkeitsuntersuchung in Form einer Langzeitbeobachtung statt. Während dieser können beispielsweise Fehlersammelkarten angefertigt oder Prozessfähigkeitsuntersuchungen durchgeführt werden. Nach der Bestätigung der Wirksamkeit sind die in D3 implementierten Sofortmaßnahmen aufzuheben. Weiterhin müssen die eingeführten Veränderungen in der Organisation verankert werden. Dazu ist es z. B. nötig, Dokumente wie Arbeitspläne, Zeichnungen oder Verfahrensbeschreibungen anzupassen und die Belegschaft entsprechend zu schulen. In der Automobilindustrie ist der Kunde spätestens an dieser Stelle in den 8D-Prozess einzubeziehen, denn dessen Zustimmung ist häufig für Prozessänderungen nötig. Weiterhin ist es möglich, dass der OEM auf der vorläufigen Beibehaltung ausgewählter Sofortmaßnahmen (z. B. 100%-Prüfung) besteht.

Der anschließende siebente Schritt (D7: Fehlerwiederholung verhindern) transferiert die gewonnenen Erkenntnisse auf vergleichbare Produkte oder Prozesse, um den aufgetretenen Fehler nachhaltig aus der Organisation zu entfernen. Es wird ein Lessons-Learned-Prozess angestoßen, von dem auch Unternehmensbestandteile profitieren, die mit der ursprünglichen Problemstellung nicht konfrontiert waren. Beispielsweise können die gewonnenen Erfahrungen in die Fehlermöglichkeits- und -einflussanalyse (FMEA) eines künftigen und ähnlichen Produktes einfließen.

Die letzte Disziplin (D8: Abschluss und Würdigung des Teamerfolgs) sieht die Beendigung des Prozesses vor. Der Abschluss der 8D-Methodik kann erst angestoßen

werden, sobald alle vorangegangenen Schritte erfolgreich durchlaufen wurden. Das Fazit des Problemlösungsprozesses wird durch ein Gespräch mit möglichst allen beteiligten Personen gezogen. In diesem werden Raum für Verbesserungsvorschläge sowie die Würdigung der Teamleistung geboten. Der Sponsor sowie der Teamleiter beenden die 8D-Methodik mittels einer dokumentierten Freigabe.

Während des kompletten Problemlösungsprozesses wird über den parallel zu erstellenden 8D-Bericht die Kommunikation zwischen Kunden und Lieferanten in einer transparenten Form sichergestellt. Er enthält die umfassende Dokumentation der gewonnenen Erkenntnisse. Der VDA bietet entsprechende Vorlagen an.

### **Fazit zur 8D-Methode**

Bei der Betrachtung der vielfältigen Prozessschritte der 8D-Methode mit ihren häufigen Wirksamkeitsnachweisen stellt sich zwangsläufig die Frage nach der Dauer des Verfahrens. Der VDA bescheinigt einzelnen Elementen einen mitunter erhöhten Zeitaufwand, ohne dabei jedoch konkrete Zahlen zu nennen. Es wird lediglich darauf verwiesen, dass „eine angemessene, nachhaltige Problemlösung nicht in Konflikt mit Zeitvorgaben“ kommen darf.<sup>[85]</sup> Diese Aussage ist in erster Linie richtig; dennoch sollte die Zeit als zentraler Kostentreiber während einer Reklamation berücksichtigt werden. Zusätzlich gilt diese Aussage zusammen mit der regelmäßigen und verständlichen Informationsweitergabe an den Kunden als Basisanforderung.<sup>[88]</sup>

In der Literatur lassen sich einige konkrete Beispiele für die erwarteten Bearbeitungszeiten der einzelnen Disziplinen finden. Diese werden jedoch individuell durch die Unternehmen festgelegt. Im Folgenden werden drei Beispiele dafür angeführt. Im ersten Beispiel eines Zulieferers der Automobilindustrie für Multimedia-Systeme wird der Abschluss von D1 nach einem und die Beendigung von D2 und D3 nach einem weiteren Tag angestrebt. Nach 14 Tagen sollen die Disziplinen vier und fünf durchlaufen sein. Das Ende des kompletten Prozesses ist nach spätestens 60 Tagen vorgesehen.<sup>[88]</sup> Im Zuge einer Dissertation wurden verschiedene Unternehmen u. a. zu der vorgesehenen Bearbeitungszeit der Disziplinen der 8D-Methode befragt. Auch in dieser Untersuchung sprach sich die Mehrheit (78,9 %) für eine Bearbeitungszeit von zwei Tagen für die Schritte D1 bis D3 aus. Knapp 70 % der Unternehmen erwarteten den Abschluss des vollständigen Prozesses jedoch bereits innerhalb eines Zeitraums von 14 Tagen.<sup>[89]</sup> In der Automobilindustrie besteht zudem die häufige Forderung, dass die Schritte eins bis drei bereits nach 24 Stunden durchlaufen sein müssen.<sup>[86]</sup> Dies bestätigt der VDA im Band des standardisierten Reklamationsprozesses. Demnach fordern Kunden Ergebnisse zu Sofortmaßnahmen in der Regel nach 24 bis 48 Stunden ein.<sup>[85]</sup> Es wird deutlich, dass hinsichtlich der Disziplinen eins bis drei ein relativ großer Konsens bezogen auf die Bearbeitungszeit besteht (24 bis 48 Stunden). Lediglich der zeitliche Horizont für das Durchlaufen der verbleibenden fünf Schritte scheint unterschiedlich gewählt zu werden.

Neben dem zeitlichen Aspekt lassen sich einige weitere Schwachstellen der 8D-Systematik ausmachen: Zum einen setzt sie methodische Kenntnisse vieler weiterer

Verfahren (z. B. des Ishikawa-Diagramms) voraus.<sup>[87]</sup> Weiterhin scheitert die Ursachenanalyse häufig bei komplexeren Wertschöpfungsketten, in denen die Fehler erst firmenübergreifend entstehen.<sup>[89]</sup> Außerdem erscheint es durch die steigende Anzahl und die zunehmenden Umfänge der 8D-Reporte häufig notwendig, eine automatische Bewertung der Inhalte vorzunehmen.<sup>[90]</sup> Für die beiden letztgenannten Punkte wurden im Rahmen der zitierten Veröffentlichungen bereits erste Lösungsansätze erarbeitet.

#### **4.4.3 Umgang mit Reklamationen in der Software-Branche**

Um dennoch einen Einblick in den Umgang mit Reklamationen innerhalb der Software-Entwicklung zu erhalten, wurde ein Interview mit einem Qualitätsingenieur aus der Automobilindustrie durchgeführt. Dieser entwickelt seit sechs Jahren Server-Software nach dem agilen Vorgehensmodell Scrum (s. Kapitel 4.8.3). Das vollständige Interview ist dem Anhang 8 im Wortlaut beigefügt. Im Folgenden soll lediglich auf die Kerninhalte des Gesprächs eingegangen werden.

Zunächst ist festzuhalten, dass dem interviewten Qualitätsingenieur keine entsprechenden Normen oder Richtlinien innerhalb der Branche bekannt sind. Er geht daher von individuellen Vorgehensweisen innerhalb der verschiedenen Firmen aus, die sich dennoch in großen Teilen überschneiden. Eine erste zentrale Aufgabe des Reklamationsprozesses in der Software-Branche besteht in der Eingrenzung und Spezifizierung der eingehenden Beschwerde. Es folgt die Formulierung als sogenannter Task. Dieser zeichnet sich durch eindeutig definierte Akzeptanzkriterien aus, die bei der Bearbeitung der Reklamation berücksichtigt werden müssen. Im Anschluss wird die Aufgabe der Problemlösung vom Produktverantwortlichen (Product Owner) priorisiert und direkt in den gewöhnlichen Arbeitsablauf des Entwicklungsteams eingetaktet. Die Priorisierung nimmt einen entscheidenden Einfluss auf die Dauer des Prozesses: Schwerwiegende Probleme können in der Regel innerhalb weniger Stunden abgewickelt werden, wohingegen die Lösung unkritischer Fehler mitunter erst nach Wochen entwickelt wird. Folglich ist der Prozess relativ flexibel gestaltet und stark abhängig von der Ausprägung der aufgetretenen Fehler. Dem Product Owner kommt in diesem Zusammenhang eine entscheidende Rolle zu.

Die Frage nach der Anwendung eines speziellen Problemlösungswerkzeuges nach dem Vorbild der 8D-Methode wurde verneint. Der nachfolgend vom Fachexperten beschriebene Ablauf scheint jedoch ähnliche Ausprägungen aufzuweisen: Das Problemlösungsteam (D1) besteht bereits durch das Entwicklungsteam und auch die Problembeschreibung (D2) ist durch den zuvor beschriebenen Prozess bereits abgeschlossen. Sofortmaßnahmen (D3) in Form von Backups sind durchaus üblich, werden jedoch nur bei einer entsprechenden Dringlichkeit ergriffen. Die innerhalb der 8D-Methodik strikt getrennten Disziplinen D4 - D6 stellen im Arbeitsalltag des Ingenieurs einen verbundenen Schritt dar, da Probleme in den meisten Fällen leicht zu überschauen und damit zu beheben sind. Bei der Definition des Tasks wird zudem genauestens festgelegt, welche Schritte bzw. Maßnahmen umgesetzt sein müssen, damit das

Problem als gelöst gilt (Definition of Done). Dazu zählen u. a. die Erfüllung aller Akzeptanzkriterien sowie das (automatisierte) Testen sämtlicher Neuerungen. Letztere spielen in der Software-Entwicklung, aber auch in der Fehlerbehebung eine entscheidende Rolle. Die Disziplinen sieben und acht der 8D-Methode sieht der Fachexperte durch die sogenannten Retrospectives innerhalb der Scrum-Vorgehensweise repräsentiert. In diesen tauscht sich das Team über die Erfahrungen des letzten Sprints (einen Zeitraum von ca. zwei Wochen) aus. Das Lernen aus Fehlern sowie deren zukünftige Vermeidung stellen vorrangige Punkte auf der diesbezüglichen Agenda dar.

Eine Dokumentation der durchgeführten Maßnahmen nach dem Vorbild eines 8D-Reports wird in aller Regel nicht vorgenommen. Dies ist vor allem Datenschutzgründen geschuldet.

Eine Use Case Software erschwert die digitale „Parkplatzsuche“ und wird im Lauf der Doktorarbeit erarbeitet; die Ergebnisse werden in Kapitel 7 (Use Case Study Reklamationsmanagement) vorgestellt.

#### **4.4.4 Standardisierter Reklamationsprozess in der Automobilindustrie**

Der standardisierte Reklamationsprozess in der Automobilindustrie bildet den zentralen Betrachtungsgegenstand der vorliegenden Arbeit. Seine Weiterentwicklung hinsichtlich der durch Connectivity aufkommenden Herausforderungen stellt das diesbezügliche Ziel dar. Zu diesem Zweck erfolgt zunächst die Darstellung des Ist-Zustandes. Dabei werden der Hintergrund sowie die Bedeutung des Prozesses vorgestellt, ehe der eigentliche Ablauf in einer verkürzten Form wiedergegeben wird.

In der Automobilindustrie ist es bisweilen üblich, dass die Qualitätsstandards von den OEMs und den großen Zulieferern (Tier 1) vorgegeben werden. Der VDA-Arbeitskreis Sicherung der Qualität im Produktlebenszyklus veröffentlichte bezogen auf das Beschwerdemanagement im Jahr 2009 einen standardisierten Reklamationsprozess. Einbezogene Vertreter aufseiten der OEMs waren dabei u. a. AUDI, BMW, Daimler sowie Volkswagen und aufseiten der großen Systemlieferanten Robert Bosch, ZF Sachs und MAGNA STEYR. Der OEM (oder der Lieferant mit vorgelagerten Wertschöpfungsstrukturen) kann die Einhaltung des VDA-Bandes von seinem Zulieferer fordern. In diesem Fall wird die Empfehlung zu einer Verpflichtung. Die Wahl des Begriffs des Reklamationsprozesses verdeutlicht dabei die Anwendung auf die Periode im Nachgang eines Geschäfts. Neben dem VDA-Band existieren innerhalb der Normenlandschaft weitere Leitfäden, die jedoch allgemeingültig gehalten sind und nachfolgend daher keine weitere Betrachtung finden.

Durch die zunehmende und bereits beschriebene Verschiebung der Wertschöpfungsanteile zu den Lieferanten steigt der Anteil der zulieferbedingten Fehler in der Automobilindustrie an. Als direkte Folge kommt der effektiven Abwicklung der resultierenden Reklamationen eine immer höhere Bedeutung zu. Bis zur Standardisierung dieses Prozesses im Jahr 2009 bestanden vielfältige Unterschiede in

der Kommunikation und der Einbindung der Prozesspartner. Durch die Festlegung des Ablaufs werden eine verringerte Bearbeitungszeit und eine damit verbundene Kosteneinsparung erzielt. Das Hauptziel des standardisierten Reklamationsprozesses besteht in der Wiederherstellung der Kundenzufriedenheit sowie in der Minimierung negativer Auswirkungen, die durch diese Unzufriedenheit bedingt sind (Kundenverluste, Imageschäden etc.).

Der standardisierte Reklamationsprozess des VDA beruht auf einer umfangreichen Prozessdarstellung, die dem Anhang 9 beigelegt ist und parallel zu den folgenden Ausführungen eingesehen werden sollte. Dabei findet die ereignisgesteuerte Prozesskette als Modellierungssprache Verwendung. Es wird ferner zwischen den vier Teilprozessen der Initialisierung des Reklamationsprozesses, der 8D-Methode, der Verifikation/des Abschlusses sowie des Ablehnens der Beanstandung differenziert. Jeder Teilprozess beinhaltet eine Reihe von weiteren Prozessschritten, die im VDA-Band genauestens beschrieben werden. Die diesbezügliche Zuordnung der Prozessschritte zu den vier Teilprozessen ist ebenfalls der Gesamtdarstellung zu entnehmen. Die Beschreibung erfolgt im VDA-Band anhand eines wiederkehrenden Schemas, das die Inhalte in einer übersichtlichen Tabellenform darstellt. Dabei werden u. a. Angaben zum Namen, der Ausprägung oder zum Ergebnis des Prozessschrittes gemacht. Eine vollständige Übersicht der Kategorien ist dem Anhang 10 beigelegt. Eine genaue Wiedergabe jedes Elements würde über den Rahmen der Arbeit hinausgehen; aus diesem Grund wird im Folgenden lediglich eine Zusammenfassung der übergeordneten vier Teilprozesse vorgenommen. Die 8D-Methode wird aufgrund ihrer zentralen Bedeutung im gesonderten Abschnitt 7.5 beleuchtet. Die Betrachtung der drei verbleibenden Teilprozesse erfolgt in einer an den VDA-Band angelehnten Form.

Den Abschluss der Initialisierung bildet gewöhnlich die kundenseitig geforderte Anwendung der 8D-Methode. Wie bereits erwähnt, kommt dieser im Reklamationsmanagement der Automobilindustrie und auch darüber hinaus eine Schlüsselfunktion zu. Für das Verständnis des nächsten Teilprozesses Verifikation/Abschluss ist daher davon auszugehen, dass das 8D-Verfahren vom Lieferanten durchlaufen und das Ergebnis in Form eines 8D-Reports an den Kunden übergeben wurde.

In Abhängigkeit der geforderten Stellungnahme sind mitunter nicht alle Prozessschritte notwendig. Im Falle einer vorangegangenen Anwendung der 8D-Methode würde beispielsweise die Wirksamkeitsprüfung lediglich eine Wiederholung bereits durchgeführter Kontrollen bedeuten.

Neben dem bisher beschriebenen erfolgreichen Durchlauf des vollständigen Reklamationsprozesses ist weiterhin eine vorläufige Ablehnung der Beanstandung durch den Lieferanten möglich. Diese kann an mehreren Stellen im Prozess angestoßen werden. Das zugehörige Prozessschaubild ist dem Anhang 11 beigelegt.

## **Fazit zum standardisierten Reklamationsprozess**

Durch die oben geschilderten Schritte des standardisierten Reklamationsprozesses wird deutlich, dass sich dieser lediglich auf den direkten BM-Prozess beschränkt. Es werden keine Verweise auf ein Lernen aus den bearbeiteten Reklamationen gegeben. Laut VDA-Band „kann eine Kundenbeanstandung als Ausgangspunkt für Verbesserungsmaßnahmen innerhalb des Unternehmens genutzt werden (Kontinuierlicher Verbesserungsprozess).“<sup>[85]</sup> Diese Option wird jedoch erst durch eine häufig angewendete 8D-Methode mit ihren Disziplinen vier bis sieben ermöglicht. Ferner wurde die Komplexität des Reklamationsprozesses hervorgehoben; diese wird durch die 8D-Anwendung weiter gesteigert. Dies resultiert in einer zeitintensiven Bearbeitung.

Durch die Beschreibungen der Prozessschritte auf der einen und durch das Veröffentlichungsjahr 2009 auf der anderen Seite wird deutlich, dass sich der hier vorgestellten und aktuellen Anwendung findende Reklamationsprozess vornehmlich auf den klassischen Automobilbau sowie Hardware-Komponenten bezieht. Im Jahr 2009 war die Automobilindustrie eine andere als heute, zehn Jahre später. Das erste Smartphone war zu diesem Zeitpunkt gerade einmal zwei Jahre auf dem Markt. Für die aktuellen und zukünftig vermehrt an Bedeutung gewinnenden Software-Anwendungen im Rahmen der Connectivity scheint eine Anpassung des Prozesses unabdingbar: Es werden Vorgehensweisen mit deutlich beschleunigten Abläufen benötigt.<sup>[91]</sup>

## **4.5 Neue Begriffe rund um die Fehleranalysestrategien digitaler Dienste**

Im Vergleich zu klassischen Bauteilen im Fahrzeug ist der Entwicklungsprozess für Apps in der Automobilindustrie deutlich schneller und komplizierter. Eine große Herausforderung stellt die dynamische sowie agile Zusammenarbeit und Unfassbarkeit der Komponenten sowie Codes dar.

Der Product Owner (PO) ist eine zentrale Rolle innerhalb des Software-Entwicklungsteams. An dieser Stelle wird daher aufgezeigt, was zu den täglichen Aufgaben eines PO gehört und welche Funktionen er wahrnimmt.<sup>[96]</sup> Die Definition richtet sich nach dem Scrum-Guide von Jeff Sutherland und Ken Schwaber, den Entwicklern des Scrum-Frameworks. Der PO ist für die Wertmaximierung des Produkts und die Arbeit des Entwicklungsteams verantwortlich. Er ist die einzige Person, die für das Management des Product Backlog zuständig ist und dafür Rechenschaft trägt. Der PO ist stets eine einzelne Person und kein Komitee; die Rolle des PO kann auch nicht zwischen mehreren Personen aufgeteilt werden, die sich dann phasenweise abwechseln.<sup>[96]</sup>

Ähnlich der Funktion eines PO soll in Zukunft eine neue Rolle, der ‚Quality Owner‘, in der Automobilindustrie für digitale Felder eingeführt werden. Diese Rolle ist in der heutigen deutschen Automobilindustrie im klassischen Qualitätsbereich noch nicht existent und muss im Rahmen des Fehleranalyseprozesses neu definiert werden.

In jeder Phase sind verschiedene Fehlermöglichkeiten denkbar. Es ist somit notwendig, in Zukunft eine transparente Datenbank für alle relevanten bereichsübergreifenden Ressourcen zu bauen.

Der PO, dessen Rolle nur im Softwarebereich bekannt ist, sowie die von Google stammende Rolle des ‚Site Reliability Engineering (SRE)‘<sup>[97]</sup> müssen hinsichtlich ihrer Rolle, Kompetenz sowie Aufgaben für digitale Dienste in der Automobilindustrie analysiert und entsprechend neu definiert werden. SRE definiert einen neuen Ansatz für den Betrieb und die Überwachung seiner Dienste und Server. Im Zentrum steht dabei eine neue Art, Verfügbarkeit zu definieren und zu messen.

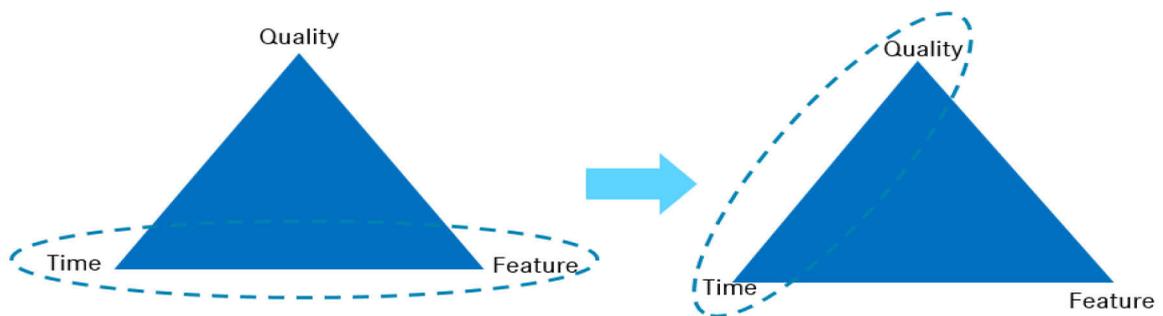


Abbildung 4.16: Fehleranalyse Pyramide & Quality-Time-Feature Dreieck (Eigene Darstellung)

Quality, Time und Feature sind drei Kernpunkte im Software-Entwicklungsprozess. In der heutigen Automobilindustrie sind digitale Produkte wie Applikationen relativ neu – im Jahr 2000 wusste kaum ein Manager, dass 2018 jeder deutsche OEM eine oder mehrere Apps für Kunden liefert. Im Vergleich zu Abbildung 4.16 fokussieren sich die Firmen in der ersten Generation intensiver auf die Anzahl der Features und der neuen Funktionen sowie das Markteinführungsdatum als auf das Kundenfeedback oder die Produktqualität. Von der Kapazitäts-, Know-how- sowie Firmenstrukturseite her stellt dies zudem eine beträchtliche Herausforderung für weitere Schritte dar. Mit Entwicklung sowie Reifegrade des Marktwachstums muss die Automobilindustrie die Relevanz der Qualität und Kundenzufriedenheit stärker priorisieren.

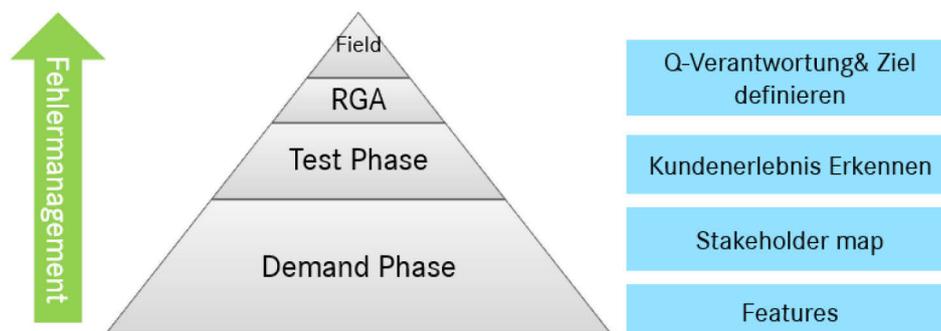


Abbildung 4.17: Rolle des Fehlermanagements im gesamten Produktlebenszyklus (Eigene Darstellung)

(RGA\*=Reifegradabsicherung)

Abbildung 4.17 zeigt den Hauptprozess einer neuen Applikation. Demand Entwicklung, Testen, Reifegradabsicherung (RGA), Felddatenanalyse sind vergleichbar mit klassischen Bauteilen. Wie die verschiedenen Phasen reibungslos zusammenarbeiten können, soll in dieser Arbeit mit Experten gemeinsam analysiert werden. Eine detaillierte Analyse des Geschäftsprozesses der zukünftigen digitalen Dienste sowie der Verantwortung für die Fehleranalyse soll im Lauf dieser Dissertation bearbeitet werden. Ein Entwurf dazu ist in Anhang 12 zu finden.

## 4.6 Agilität

### Agile Zusammenarbeit

Die Verbreitung des Internets um das Jahr 2000 veränderte durch den Auf- und Abstieg verschiedenster Technologien die gesamte Software-Industrie nachhaltig. Ideen konnten nun in kürzester Zeit mit einer Vielzahl von Menschen geteilt und weiterentwickelt werden, was die Verbreitung und Adaption von Technologien deutlich erleichterte. Investitionen in die IT-Branche bewirkten ein Aufkommen neuer Unternehmen und eine Zunahme an Merger & Acquisition-Aktivitäten, um neue Kompetenzen und Technologien frühzeitig einzubinden.<sup>[98]</sup>

Neben der veränderten Organisationsstruktur wurde eine Berücksichtigung von sich schnell verändernden Kundenanforderungen immer essenzieller für Software-Projekte. Um die Anforderungen von außen zu kompensieren, musste sich auch die Form der Zusammenarbeit bei der Software-Entwicklung drastischen Änderungen unterziehen. Unter dem Stichwort der agilen Software-Entwicklung etablierten sich Vorgehensmodelle, die ein zügiges Einbinden von Kundenfeedback in den Entwicklungsprozess zulassen. Das Etablieren von kurzen Entwicklungszyklen und einer vielfachen Erstellung von Prototypen für Abstimmungszwecke hält Sponsoren, Kunden, Nutzer und Entwickler nahe beisammen und ermöglicht eine konsequente Kollaboration.<sup>[99]</sup>

#### 4.6.1 Agiles Qualitätsmanagement nach DGQ

Die bisherigen Ausführungen verdeutlichen, dass OEMs vor der Herausforderung stehen, agile Prozesse und Methoden zu etablieren, um in zukünftigen Geschäftsfeldern dynamisch und damit erfolgreich agieren zu können. Vor diesem Hintergrund kann auch der Bereich des QMs agil gestaltet werden.

Es ist anzunehmen, dass eine strikte Aufrechterhaltung der Grundprinzipien des klassischen QMs eine erfolgreiche digitale Transformation nicht optimal unterstützen kann. Hierfür ist eine engere Verzahnung von Entwicklung und dem Qualitätsmanagement notwendig.<sup>[100]</sup> Agile Vorgehensmodelle sind durch die im agilen Manifest zusammengefassten Werte und Prioritäten gekennzeichnet. Weitere interessante Überlegungen hierzu wurden von der Deutschen Gesellschaft für Qualität (DGQ) veröffentlicht. In dieser wurden die Grundsätze der ISO 9001 als Basis für die Formulierung agiler QM-Grundsätze verwendet. Im Folgenden werden diese

zusammengefasst (Tab. 4.3) und auf Grundlage der Ausführungen von Sommerhoff und dem DGQ-Fachkreis erläutert.<sup>[21]</sup>

Grundsätze der ISO 9001	Grundsätze des agilen QM
Kundenorientierung	Kundeninteraktion
Führung	Dienste Führung
Einbeziehung von Personen	Interdisziplinäre Vernetzung
Prozessorientierter Ansatz	Evolutionärer Ansatz
Verbesserung	Iteration
Faktengestützte Entscheidungsfindung	Knackpunktbasierte Lösungsfindung
Beziehungsmanagement	Menschenzentrierung

Tabelle 4.3: Grundsätze des agilen QM (Eigene Darstellung, Forschungsstudie QSK der TU Berlin, nach Sommerhoff, B./ DGQ-Fachkreis)<sup>[21]</sup>

Eine Betrachtung der agil umgedeuteten Grundsätze zeigt, dass klassische hierarchische Strukturen für agile Arbeitsweisen nicht zielführend sind. Interdisziplinäre Teams arbeiten mit einer hohen Eigenverantwortung und sollten mit angemessenen Entscheidungskompetenzen ausgestattet werden. Weiterhin steht der Kunde deutlich stärker im Mittelpunkt. Entwicklungen werden in enger Verzahnung mit Kunden vorangetrieben, um die Umsetzung von tatsächlich gewünschten Anforderungen zu gewährleisten. Die Integration von Kunden in Entwicklungsprozesse gilt in diesem Zusammenhang als Erfolgsfaktor für die Transformation vom Produktanbieter zum Dienstleistungsunternehmen.<sup>[101]</sup> Weiterhin wird die Verbesserung iterativ betrachtet und lässt somit Rückschritte während der gesamten Entwicklungslaufzeit zu. Zwar ist es üblich, dass die Qualitätssicherung in agilen Projekten weitestgehend durch Entwicklungsteams übernommen wird, jedoch stößt dieses Konzept spätestens bei komplexeren Großprojekten an seine Grenzen. In derartigen Projekten müssen Entwicklungen von mehreren autark arbeitenden Entwicklungsteams gesteuert werden. An dieser Stelle ist es nicht zielführend, wenn große Diskrepanzen innerhalb der Qualitätsdefinitionen von selbstorganisierten Entwicklungsteams vorliegen.<sup>[102]</sup> Vor diesem Hintergrund ist, insbesondere für das QM global verteilter Großprojekte, die Entwicklung eines agilen Qualitätsverständnisses auf höchster Ebene als strategisch wichtiger Schritt einzuordnen.<sup>[100]</sup> Die zuvor formulierten Grundsätze spiegeln die im agilen Manifest ausformulierten Grundsätze wider und könnten als Basis für einen agilen Kulturwandel im Unternehmen verankert werden.

## 4.6.2 Werte und Prinzipien agiler Methoden

Ins Leben gerufen wurde die Diskussion über agile Methoden von 17 Experten im Sektor der Softwareentwicklung. Anfang der Jahrtausendwende wurde bei einem Treffen jener das agile Manifest verfasst, welches die vier Prinzipien der agilen Softwareentwicklung aufzeigt. Die vier Wertvorstellungen bilden die Essenz der agilen Methoden. Die vier Werte des agilen Manifestes sind:<sup>[103]</sup>

*„Individuen und Interaktionen mehr als Prozesse und Werkzeuge“*

Eine Formalisierung der Prozesse besitzt eine untergeordnete Rolle gegenüber der kontinuierlichen, informalen Arbeitsweise, welche sich in enger Kooperation mit allen Beteiligten orientiert. Die Beziehung und das Gefühl der Zusammengehörigkeit der Entwickler sind weitaus wichtiger als institutionelle Prozesse und Entwicklertools. Der Mitarbeiter und Teamarbeit stehen im Zentrum, Beziehungsgefüge und das angebotene Arbeitsumfeld gelten als Erfolgsfaktoren.<sup>[104]</sup>

*„Funktionierende Software mehr als umfassende Dokumentation“*

In der agilen Arbeitsweise steht die kontinuierliche Ausgabe von testfähigen Software-Releases im Fokus. Innerhalb von Zeitfenstern, welche nur wenige Wochen umfassen, werden einsatzfähige Softwarepakete entwickelt. Es wird versucht, die Dokumentation so schlank wie möglich zu halten. Es wird dokumentiert, was nötig ist, aber nicht des dokumentieren Willens.<sup>[105]</sup>

*„Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung“*

Die enge und hochfrequentierte Abstimmung zwischen Kunden und Entwicklern ist eine weitere Kernessenz der agilen Arbeitsweise. Verträge sind weiterhin wichtig, aber es sollen keine detailliert ausgehandelten Vertragsvereinbarungen vor dem Projekt bestehen. Abrahamsson<sup>[105]</sup> spricht in diesem Zusammenhang von Vertragsverhandlungen, welche eine zukunftsfähige, kooperative Beziehung erreichen und bewahren sollen.

*„Reagieren auf Veränderung mehr als das Befolgen eines Plans“*

Verträge müssen die Möglichkeiten zur Veränderung der Anforderungen zulassen, um eine sinnvolle Entwicklung zu ermöglichen. Des Weiteren sind Entwickler und Repräsentanten beim Kunden mit entsprechenden Entscheidungsbefugnissen zu ermächtigen, welche ihnen Anpassungen während des Softwareentwicklungsprojekts erlauben.<sup>[105]</sup> Die vier beschriebenen Werte bilden das Fundament für die agile Arbeitsweise und werden von zwölf Prinzipien weiter ausformuliert:<sup>[106]</sup>

- 1) Höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufriedenzustellen.
- 2) Anforderungsänderungen selbst spät in der Entwicklung willkommen zu heißen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.

- 3) Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.
- 4) Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.
- 5) Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.
- 6) Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.
- 7) Funktionierende Software ist das wichtigste Fortschrittsmaß.
- 8) Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten ein gleichmäßiges Tempo auf unbegrenzte Zeit halten können.
- 9) Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.
- 10) Einfachheit; die Kunst, die Menge nicht getaner Arbeit zu maximieren, ist essenziell.
- 11) Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierende Teams.
- 12) In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann, und passt sein Verhalten dementsprechend an.

Die zwölf agilen Prinzipien sowie die vier Werte des agilen Manifests bilden den Grundgedanken von agilen Arbeitsweisen ab.

#### **4.6.3 Agile Methode Scrum**

Nachdem die Grundlagen und die damit verbundenen Werte und Prinzipien der agilen Arbeitsweise dargestellt wurden, wird nun ein tieferer Einblick in das wohl bekannteste agile Vorgehensmodell, auch bekannt als Scrum, geliefert.

Scrum ist unter den agilen Methoden zurzeit das relevanteste Vorgehensmodell für Unternehmen. Scrum bietet Unternehmen die Möglichkeit, während der Entwicklung dynamisch auf Veränderungen reagieren zu können. Somit bietet Scrum insbesondere für Unternehmen aus der Softwarebranche einen geeigneten Rahmen für die Entwicklung passgenauer Softwareprodukte.

Scrum basiert auf der Theorie der empirischen Prozesssteuerung und auf den drei Säulen *Transparenz*, *Überprüfung* und *Anpassung*, welche entscheidende Rollen für den Erfolg von Scrum einnehmen.<sup>[107]</sup>

- 1) *Transparenz*: Verantwortliche müssen Einsicht in alle Aspekte eines Prozesses besitzen. Das heißt, es muss ein Verständnis der Prozessschritte bei allen Beteiligten vorherrschen und die Aspekte müssen klaren Standards folgen.
- 2) *Überprüfung*: Die Anwender von Scrum müssen in regelmäßigen Abständen den Fortschritt in Bezug auf die Zielerreichung überprüfen, um Abweichungen rechtzeitig zu erkennen.
- 3) *Anpassung*: Wird festgestellt, dass das Ergebnis von den akzeptablen Grenzwerten abweicht, muss der Prozess oder das Material angepasst werden.

Diese Grundpfeiler werden durch fünf grundlegende Werte unterstützt: *Mut, Verantwortung, Fokus, Offenheit* und *Respekt*.<sup>[107]</sup>

Das Scrum Team verpflichtet sich, nach diesen Werten zu streben, und lebt diese bei der täglichen Arbeit mit den Scrum Ereignissen, Artefakten und Rollen. Im Scrum Team wird mutig agiert, das heißt, Herausforderungen, etwas Neues zu tun, werden angenommen und Fehler zugegeben. Jeder im Team übernimmt Verantwortung und trägt mit vollem Einsatz und Engagement einen Betrag zum Gesamtergebnis bei. Es wird sich auf eine begrenzte Anzahl von Anforderungen fokussiert, damit dem Kunden innerhalb einer kurzen Zeitperiode erste Ergebnisse geliefert werden können. Das Team organisiert sich selbst und es besteht eine Fehlerkultur, die es verlangt, dass Probleme offen angesprochen und diskutiert werden. Zuletzt bildet gegenseitiger Respekt den Kern der Zusammenarbeit im Team.<sup>[106]</sup>

### Scrum Prozess

Der Ablauf des Scrum Prozesses folgt einem inkrementellen Muster und hat fest definierte Ereignisse, Artefakte sowie Rollen mit ihren jeweiligen Verantwortlichkeiten. Im Folgenden wird der Ablauf der Entwicklung in einem Scrum Projekt nach Schwaber<sup>[107]</sup>, Schmidt<sup>[108]</sup> und Broy/Kuhrmann<sup>[27]</sup> dargestellt (s. Abb. 4.18).

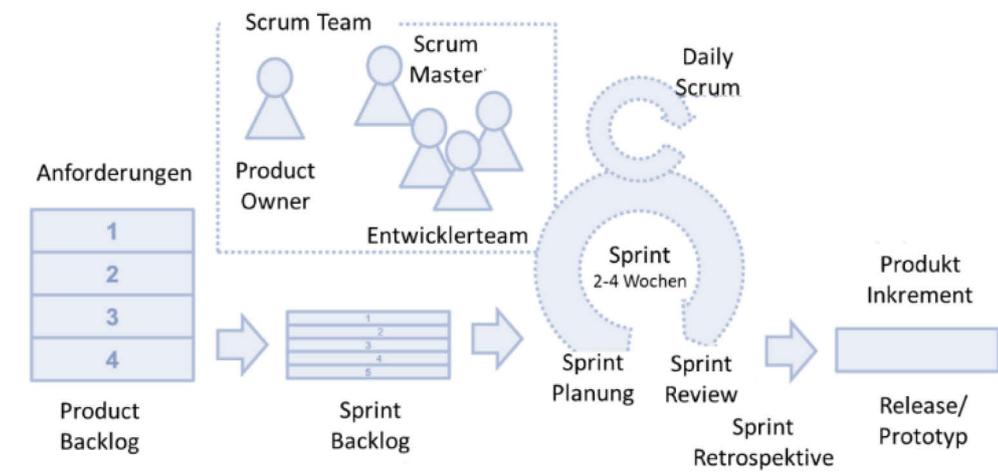


Abbildung 4.18: Ablauf des Scrum-Entwicklungsprozesses (Quelle: Schmidt 2016<sup>[108]</sup>)

### 4.6.4 DevOps

Die immer kürzer werdenden Innovationszyklen, die steigenden Qualitätsanforderungen und eine stetige Zunahme an Komplexität der Softwareanwendungen und -systeme haben die Entwicklung agiler Methoden und Vorgehensmodelle vorangetrieben. Diese ermöglichen es, Software in kurzen Zeiträumen zu entwickeln, zu verbessern und zu implementieren. Im klassischen Software-Produktlebenszyklus findet, ähnlich wie in der Produktion der Automobilindustrie, eine klare Unterscheidung in Entwicklungs-, Test- und Betriebsphase statt. Dementsprechend sind auch die Organisationseinheiten in klassisch funktional aufgebauten IT-Unternehmen in Entwicklung („Developer“), Test- bzw. Qualitätssicherungsabteilung und Betrieb („Operations“) strukturiert. Der Erfolg der

agilen Entwicklungsmethoden hat dazu geführt, dass ein Teil der Softwaretests schon während der Entwicklung durchgeführt werden kann. Test- und Entwicklungsabteilung sind daher kaum mehr voneinander zu trennen. Die schnelle nutzerzentrierte Realisierung von Kundenanforderungen in der Entwicklungsphase beeinflusst die Benutzerakzeptanz und Kundenzufriedenheit neuer IT-Produkte und -Services maßgeblich.<sup>[109]</sup> Besonders bezogen auf Produkte und Services im Kontext der Digitalisierung (z. B. Web oder Mobile Applications) hat sich das herkömmliche Verständnis von Benutzerfreundlichkeit (Usability) durch die Fokussierung auf das Kundenerlebnis (User Experience) erweitert.<sup>[110]</sup> Während agile Teams in kurzen Releasezyklen Software an den Kunden ausliefern wollen, hat der Betrieb traditionell das Bestreben, Änderungen an stabil laufender Software möglichst zu vermeiden.<sup>[26]</sup> Diesen Zielkonflikt versucht Developer Operations (Abkürzung: DevOps) zu lösen. Die Grundprinzipien von DevOps sind laut Jez Humble und Joanne Molesky:<sup>[111]</sup>

- Culture: Entwickler und Mitarbeiter der IT-Betriebsabteilung sollten in den jeweiligen anderen Bereichen integriert werden,
- Automation: Die Automatisierung der Prozesse von Entwicklung über Test und Bereitstellung bis hin zur Produktivnahme ist der Schlüssel für die Reduzierung von Durchlaufzeiten, die Vermeidung von Fehlern und für ein schnelleres Feedback zum Release an die Entwickler,
- Measurement: Messung, Überwachung und Visualisierung des Fortschritts mittels ausgewählter KPIs – bspw. Auswirkungen der entwickelten und ausgelieferten Software auf die Stabilität der Systeme,
- Sharing: Wissen, Erfolge, Techniken, Tools und Infrastruktur sollten zwischen Entwickler und Betriebsabteilung geteilt werden.

Ähnlich wie Extreme Programming setzt auch DevOps auf den Ansatz von „Continuous Integration“.<sup>[111]</sup> Auf dem Weg vom Entwicklungsteam zum Anwender durchläuft Software nach dem Erstellen („Build“) diverse Tests, wie etwa Funktions-, Last- oder Integrationstests, die ein Deployment in unterschiedliche Umgebungen erfordern, bevor die getestete Anwendung in die Produktivumgebung ausgeliefert bzw. implementiert wird.<sup>[111]</sup> Dieser Weg wird in der Regel manuell unter hohem Zeitaufwand und hoher Fehleranfälligkeit ausgeführt. DevOps versucht, den gesamten Prozess in der Entwicklungs- bis zur Produktivumgebung weitestgehend zu automatisieren, sodass eine kontinuierliche Auslieferung von Software (Continuous Delivery) ermöglicht wird. Der Ablauf und dabei verwendeten Software- und Managementtools werden als „Delivery Pipeline“ bezeichnet.<sup>[111]</sup> Die kontinuierliche Auslieferung wird durch Konzepte aus agilen Methoden wie bspw. XP erreicht. Continuous Integration, Continuous Testing oder auch Agile Testing sind Konzepte, die sich im Grunde ähneln und eine enge Verknüpfung von Entwicklung und Durchführung von Tests realisieren. Ziel dieser Methoden ist vor allem eine frühzeitige Fehlererkennung und -behebung. Dabei spielt die Automatisierung der Tests eine entscheidende Rolle, um die Geschwindigkeit und Qualität der Softwareentwicklung nochmals zu erhöhen. Facebook hat bspw. ein automatisiertes Testtool namens SAPIENZ entwickelt, das Code wesentlich schneller als andere Tools testet, auswertet und dokumentiert.<sup>[111]</sup> Zukünftig setzt Facebook im

Bereich der Fehlererkennung auf KI, so entwickelt Facebook gerade ein auf KI basierendes Tool namens SapFix, das Fehler mithilfe von KI finden und beheben kann.<sup>[113]</sup> SapFix soll zukünftig als Open-Source-Code zur Verfügung gestellt werden.

Das Konzept des Continuous Deployment erweitert den Ansatz des automatisierten Testens auf die automatisierte Installation und Integration in die eigentliche Produktivumgebung. Hier ist eine enge Zusammenarbeit zwischen Entwicklung und Betrieb nötig, um ein schnelles Feedback und ggf. Anpassungsmaßnahmen zu ermöglichen. Neben den „Continuous“ Konzepten ist das Konzept „Infrastructure as Code“ (IaC) eine Voraussetzung für die erfolgreiche Umsetzung von DevOps.<sup>[110]</sup> Das Ziel von IaC ist eine regelbasierte automatisierte Installation und Konfiguration von virtuellen und physischen Infrastrukturkomponenten (bspw. Backend, Server) durch Softwareskripte.<sup>[114]</sup> IaC erfordert die frühzeitige Einbeziehung von Systemadministratoren in den Softwareentwicklungsprozess sowie ein tieferes Verständnis der Entwickler für die Basis-Infrastruktur.<sup>[114]</sup>

Auch Google verfolgt den Ansatz von DevOps und versucht, diesen mit radikaler Automatisierung zu verbinden. Reine Systemadministratoren gibt es bei Google nicht mehr, sondern die Aufgaben werden durch sog. SRE (Site Reliability Engineering) Teams übernommen. Diese führen die Aufgaben der Systemadministratoren jedoch nicht selbst manuell aus, sondern versuchen, diese komplett in Code zu übersetzen und zu automatisieren. Dazu werden Fachkräfte gesucht, die zum einen Qualifikationen im Bereich der Softwareentwicklung und zum anderen im Bereich der Netzwerk- und Systembetreuung aufweisen.<sup>[116]</sup> Diese bilden den Kern der SRE Teams und orchestrieren die Entwicklung in Bezug auf das Gesamtsystem. Benjamin Treynor Sloss, Vizepräsident der Entwicklungsabteilung bei Google, beschreibt die SREs folgendermaßen: „SRE is what happens when you ask a software engineer to design an operations team“. Der Betrieb und die Architektur des Betriebes muss demnach aus Entwicklersicht entworfen werden. Der „State of DevOps Report 2016“ ermittelte verschiedene Produktivitätskennzahlen von Unternehmen, die Praktiken im Sinne von DevOps verwenden. Dabei wurde festgestellt, dass Unternehmen, die viele DevOps-Praktiken integriert haben („High Performers“), signifikante Vorteile im Bereich der Softwareentwicklung aufweisen.<sup>[117]</sup>

## **5 Anpassung der Rolle des Qualitätsmanagements im Hinblick auf agile Methoden im Bereich Connectivity**

Qualität bleibt in Software-Projekten wichtigster Treiber für Kostensenkungen und Zeitreduzierung. Nach Bennett/Wennberg<sup>[3]</sup> ist die Beseitigung von Fehlern in fortgeschrittenen Lebenszyklusphasen, ähnlich wie bei Hardware, mit mehr Zeitaufwand und Kosten verbunden als frühzeitige Fehleridentifikation oder -prävention.

Das Qualitätsmanagement der OEMs steht nun also vor der Herausforderung, die Qualität von Software sicherzustellen. Aber während sich die Produktstruktur und die Anforderungen der Zusammenarbeit in der Entwicklung geändert haben, ist das Qualitätsmanagement-System noch immer das traditionelle nach ISO 9000.

Vor diesem Hintergrund ist zu klären, ob das etablierte Qualitätsmanagement bei den OEMs in seiner jetzigen Form weiterhin optimal die Qualität in den Produktentstehungsprozess einbringen kann oder ob es neue Modelle von Qualitätssystemen bedarf.

### **5.1 Zielsetzung und Forschungsfragen**

In diesem Kapitel soll die neue Rolle des agilen Qualitätsmanagers hergeleitet werden. Dazu wird verdeutlicht, wie sich das Qualitätsmanagement anpassen muss, um die ermittelten Problembereiche zu kompensieren. Vorbild sind bei der Entwicklung der neuen Rolle des Qualitätsmanagers 2.0 die empirischen Untersuchungsergebnisse in der agilen Softwareherstellung, welche schon jahrelange Erfahrungen auf dem Gebiet der agilen Qualitätssicherung von Software besitzen. Der Fokus des Kapitels liegt auf der Anpassung der Rolle des Qualitätsmanagers.

Zur Vervollständigung wird der Fokus im letzten Unterkapitel auf das Qualitätssystem gelenkt. Hier wird eine Aussicht auf mögliche Anpassungen der Organisation gegeben, welche in den Einflussbereich des Managements und der Unternehmensführung fällt und somit nicht durch das Aufgabengebiet der Rolle des Qualitätsmanagers berücksichtigt wird.

Angesichts der sich verändernden Aufbaustruktur digitaler Produkte und aufkommenden agilen Entwicklungsprozessen in der Automobilindustrie (siehe Kapitel 2.4) liegen diesem Kapitel folgende zwei Forschungsfragen zugrunde:

- 1) An welchen Stellen befinden sich Abweichungen zwischen traditionellem QM und agilen Arbeitsweisen in der Automobilindustrie? Gibt es Potentiale und Gemeinsamkeiten?
- 2) Wie kann die traditionelle QM-Rolle weiterentwickelt und in agile Vorgehensmodelle integriert werden?

## 5.2 Zusammenfassung der Analyse

In diesem Kapitel werden anhand aufgestellter Untersuchungsobjekte die agilen Methoden und Qualitätsmanagement-Systeme in ihrem Vorgehen gegenübergestellt. Darauf aufbauend werden anhand der empirischen Untersuchungen durch Interviews Problembereiche zwischen der Rolle des Qualitätsmanagers und der agilen Softwareentwicklung im Automobilbereich ermittelt. Mit den Untersuchungen im IT-Sektor werden erste Lösungsansätze angedeutet .

Im ersten Teil des Kapitels werden Untersuchungsobjekte hergeleitet, anhand derer sich die folgende Analyse orientieren kann. Die Untersuchungsobjekte orientieren sich an Ringbauer<sup>[118]</sup> und werden durch die Gegenüberstellung der agilen Prinzipien, den Grundsätzen der ISO 9000 und den Leitsätzen des Lean Managements erweitert.

Im zweiten Teil des Kapitels, dem Analysekapitel, werden die hergeleiteten Untersuchungsobjekte differenziert betrachtet. Der praktische Teil beschreibt die aktuelle Situation führender OEMs hinsichtlich des Wandels durch verschiedene Technologietrends. Interessant sind hierbei die Vision und die Strategie, welche die Richtung der Unternehmensaktivitäten aufzeigen. Hieran soll vermittelt werden, wie sich die Ausrichtung auf neue Geschäftsmodelle in der Anpassung von Unternehmenskultur und -strategie bemerkbar macht. Im letzten Abschnitt dieses Kapitels werden Beispiele digitaler Produkte und Services vorgestellt.

In Tabelle 5.1 werden die Analyseergebnisse je Untersuchungsobjekt nochmals übersichtlich zusammengestellt.

Situation OEM	Problematik	Situation SW-Hersteller
<b>Unternehmenskultur:</b> Organisationsaufbau, kulturelle Werte der Belegschaft		
<ul style="list-style-type: none"> <li>- neu geschaffene Bereiche mit Mitarbeitern aus SW-Firmen, welche andere Kultur aufweisen</li> <li>- Belegschaft aus dem HW-Bereich lebt dagegen bereits etablierte Kultur und betrachtet agile Methoden als Projektmanagement</li> <li>- funktionale Organisationsstruktur mit hierarchischem Aufbau</li> <li>- agile SW-Entwicklung und QM sind eigenständige Funktionsbereiche</li> </ul>	<ul style="list-style-type: none"> <li>- funktionaler Aufbau begrenzt Zusammenarbeit zwischen Entwicklung und QM auf wenige Schnittstellen</li> <li>- kulturelle Differenzen zwischen Mitarbeitern aus dem HW-Bereich und dem SW-Bereich</li> </ul>	<ul style="list-style-type: none"> <li>- holokratische Organisation in Zirkel und Rollen</li> <li>- agile Unternehmenskultur wird von agiler Organisationsstruktur unterstützt</li> <li>- Qualität wird durch die Rolle des Testmanagers im agilen Team berücksichtigt</li> </ul>
<b>Kundenorientierung:</b> Aufnahme von Kundenanforderungen, Entstehung neuer Produktideen		
<ul style="list-style-type: none"> <li>- geteilte Rolle: PO besitzt technische Sichtweise und QM besitzt Endkunden-Perspektive</li> <li>- agiles Team nimmt regelmäßig Kundenanforderungen durch PO auf</li> <li>- QM nur einmalig in Anforderungsaufnahme eingegliedert</li> <li>- Demand Management nimmt Ideen von unterschiedlichen Quellen auf</li> </ul>	<ul style="list-style-type: none"> <li>- Nutzerperspektive wird durch fehlende Integration des QMs nicht berücksichtigt</li> </ul>	<ul style="list-style-type: none"> <li>- PO ist ebenfalls einziger Kundenkontakt – allerdings unproblematisch, da sich befragte SW-Hersteller in Lieferantenrolle befinden</li> <li>- Auftraggeber tritt mit Problemstellung an SW-Unternehmen heran</li> </ul>

<b>Ständige Verbesserung</b>		
<ul style="list-style-type: none"> <li>- Teamübergreifend: Lessons Learned Sitzungen nach Releaszeitpunkten finden alle sechs bis zwölf Wochen statt</li> <li>- Teamintern: Product Reviews, etc.</li> <li>- QM ist nicht in teaminterne Meetings eingebunden</li> </ul>	<ul style="list-style-type: none"> <li>- Frequenz von Lessons Learned Sitzungen sind an lange Releaseabstände gekoppelt</li> <li>- Nutzerperspektive kann nicht in agile Entwicklung einfließen</li> </ul>	<ul style="list-style-type: none"> <li>- Anwendung von täglichen Meetings, Retrospektiven und Produkt Reviews innerhalb des Teams</li> <li>- Cross-team- und Organisations-Retrospektiven</li> </ul>
<b>Mitarbeiterorientierung (MAO): Fortbildung, Kompetenzen, Mitarbeiterumfeld</b>		
<ul style="list-style-type: none"> <li>- MAO wird als Mitarbeiterorientiertes Führen verstanden</li> <li>- Regelmäßige Fortbildung von Mitarbeitern</li> <li>- Besetzung von Schnittstellen mit cross-funktionalen Teams</li> <li>- Einstellung von Mitarbeitern mit Berufserfahrung im Bereich SW-Entwicklung</li> <li>- Neugestaltung der Arbeitsbereiche wird in Strategie aufgenommen</li> </ul>	<ul style="list-style-type: none"> <li>- K.A.</li> </ul>	<ul style="list-style-type: none"> <li>- Führung wird in Rollen aufgeteilt und Scrum Master ist verantwortlich für Fortbildung</li> <li>- Soziale Aspekte wie Arbeitsumfeld spielen bedeutende Rolle</li> </ul>
<b>Führung: Führungsstil, Rolle des QMs, Hierarchie</b>		
<ul style="list-style-type: none"> <li>- Wandel der Führungsrolle zu semi-agiler Führung</li> <li>- Hybridform in der Entwicklung: Übergeordnet, zentraler, hierarchischer Führungsstil der Weisungsbefugnis hat; SW-Bereich mit Rollenverteilung im agilen Team</li> <li>- Auch hierarchiearmer Ansatz "Schwarm" in anderen Bereichen verfügbar</li> <li>- QM besitzt keine eigene Rolle im agilen Team, sondern ist separate Abteilung</li> </ul>	<ul style="list-style-type: none"> <li>- Starke Differenzen zwischen hierarchiegetriebener Führung und autonomer Teams</li> <li>- Fehlende Rolle des QMs im agilen Team limitiert die Integration</li> </ul>	<ul style="list-style-type: none"> <li>- Dezentral verteilte Führung im agilen Team, weiterhin auch Hierarchie zu finden</li> <li>- Klassische QM Rolle nicht vorhanden, sondern Testmanager oder Qualitäts-beauftragter, welche ins agile Team integriert sind</li> </ul>
<b>Umgang mit Fehler: Fehlerkultur</b>		
<ul style="list-style-type: none"> <li>- Im HW-Bereich: Geringe Toleranz für Fehler, da lange Entwicklungszeiten und hohe Folgekosten</li> <li>- Im SW-Bereich: Hohe Toleranz für Fehler, da schnelle Behebung durch kurzzyklische Entwicklung möglich</li> <li>- QM nimmt Fehlertoleranz in Arbeitsweise auf</li> <li>- Wartung angesichts komplexer SW-Lieferanten-Struktur nicht klar geregelt</li> </ul>	<ul style="list-style-type: none"> <li>- Im After-Sales ist Fehleranzahl im SW-Bereich essenziell höher als im HW-Bereich</li> <li>- Langfristige Wartung und Updates für gesamten Automobillebenszyklus notwendig</li> </ul>	<ul style="list-style-type: none"> <li>- Wartung und Updates werden von Entwicklungsbereich verantwortet</li> </ul>
<b>Prozessuales Vorgehen: Planungshorizont, Aufbau der SW-Entwicklung, teamübergreifende Abstimmung und Koordination, Anzahl agiler Methoden, Kopplung HW-SW</b>		

<ul style="list-style-type: none"> <li>- Etablierte HW-Entwicklung: Sequentiell aufgebaut, langfristige Planung</li> <li>- QM im HW Bereich durch Schnittstellen ausreichend integriert</li> <li>- Neue SW-Entwicklung: inkrementeller Ablauf mit schneller, Releaseplanung mit kurzfristiger Detailplanung</li> <li>- QM nicht ausreichend auf kurze Zyklen abgestimmt</li> <li>- SW-Entwicklung besteht aus großer Anzahl interner und externer Entwicklerteams</li> <li>- Heterogene Anwendung von agilen Methoden</li> <li>- Starke Kopplung von SW an HW-Entwicklungszyklus, weshalb Anforderungsabstimmung zu frühem Zeitpunkt notwendig ist, geplante Lösungsstrategie ist erweitertes HW-Angebot</li> </ul>	<ul style="list-style-type: none"> <li>- QM prozessual nicht auf kurze Entwicklungszyklen angepasst</li> <li>- Struktur der SW-Entwicklung erschwert Abstimmung zwischen agilen Teams</li> <li>- Heterogene Methoden bedeuten unterschiedliche Prozesse, Rollen und Verantwortlichkeiten, was einen qualitätsseitigen Eingriff erschwert</li> <li>- HW-Kopplung bedeutet Einschränkung bzgl. der Berücksichtigung von Änderungswünschen</li> </ul>	<ul style="list-style-type: none"> <li>- Qualitätsbeauftragter ist Teil des agilen Teams</li> <li>- Problematik in End-to-End Phase wird durch regelmäßige, teamübergreifende Abstimmung angeleitet und durch den Qualitätsbeauftragten begegnet</li> <li>- Testphase bereits in zyklisches Vorgehen integriert und durch Entwickler ausgeführt</li> </ul>
<p><b>Visuelles Management: Kommunikationswege, Visuelle Objekte</b></p>		
<ul style="list-style-type: none"> <li>- SW-Bereich zeigt wenig Dokumentation, da viel über mündliche Absprachen geregelt wird</li> <li>- Fehlende einheitliche Testing-Plattformen innerhalb des SW-Bereichs und zwischen SW- und HW-Bereich</li> </ul>	<ul style="list-style-type: none"> <li>- Bild der Unverbindlichkeit seitens agiler Methoden entsteht</li> <li>- Vorurteile gegenüber agilen Methoden entstehen</li> <li>- Enormer Koordinations- und Kommunikationsaufwand in agil arbeitenden Bereichen samt Schnittstellen</li> <li>- Intransparenz bzgl. Testing Historie zwischen verschiedenen Bereichen und Schnittstellen</li> </ul>	<ul style="list-style-type: none"> <li>- Regelmeetings innerhalb und außerhalb des Teams</li> <li>- Extreme Kultur der Offenheit und des Austauschs</li> <li>- Digitale Tools, die bei der team- und hierarchieübergreifenden Kommunikation helfen, einheitliche Testing Plattformen</li> </ul>
<p><b>Lieferantenmanagement – Vertragsart, Lieferantennetzwerk, Risiken</b></p>		
<ul style="list-style-type: none"> <li>- Weite Teile der SW-Entwicklung sind extern</li> <li>- PO je nach Vertragsart intern und agiles Team extern vergeben</li> </ul>	<ul style="list-style-type: none"> <li>- Fehlendes internes SW-Knowhow durch Auslagerung</li> <li>- Qualitätsaspekt nur beschränkt vertraglich festlegbar</li> <li>- Intransparentes Lieferantennetzwerk</li> <li>- Gefahr der klassisch, starren Kunde-Lieferanten-Beziehung mit Lasten- und Pflichtenheft</li> <li>- QM nicht regelmäßig in Kommunikation zum SW-Lieferanten involviert</li> </ul>	<ul style="list-style-type: none"> <li>- Starke Zusammenarbeit mit externen Lieferanten</li> </ul>

Tabelle 5.1: Zusammenfassung der Analyseergebnisse (Eigene Darstellung, Forschungsstudie QSK der TU Berlin.)

## 5.2.1 Rahmenbedingungen für die Rolle des Qualitätsmanagers

Bei der Recherche der Literatur zu agilen Arbeitsweisen konnte das Aufgabengebiet des Qualitätsmanagers keiner vorgefundenen Rolle im agilen Team zugeordnet werden.<sup>[106]</sup> Auch die Empirie zeigte, dass diese Rolle im IT-Bereich unbekannt ist (vgl. Anhang 22, Interview 4). Die Empirie zeigte, dass das QM denselben funktionalen Aufbau besitzt, wie das, was für die Hardwareentwicklung der Fall ist. Die Folge sind fehlende Schnittstellen und Regelprozesse, welche für ausreichende Abstimmung zwischen QM und agilem Team sorgen. In prozessualer Hinsicht wurde gezeigt, dass das QM somit nicht regelmäßig in die schnelleren Entwicklungszyklen integriert ist und der Informationsaustausch nicht stattfinden kann. Hinsichtlich der Verbesserung und Kundenorientierung konnte festgestellt werden, dass die Nutzerperspektive seitens des QMs nicht in Entwicklungsprozesse einfließt.

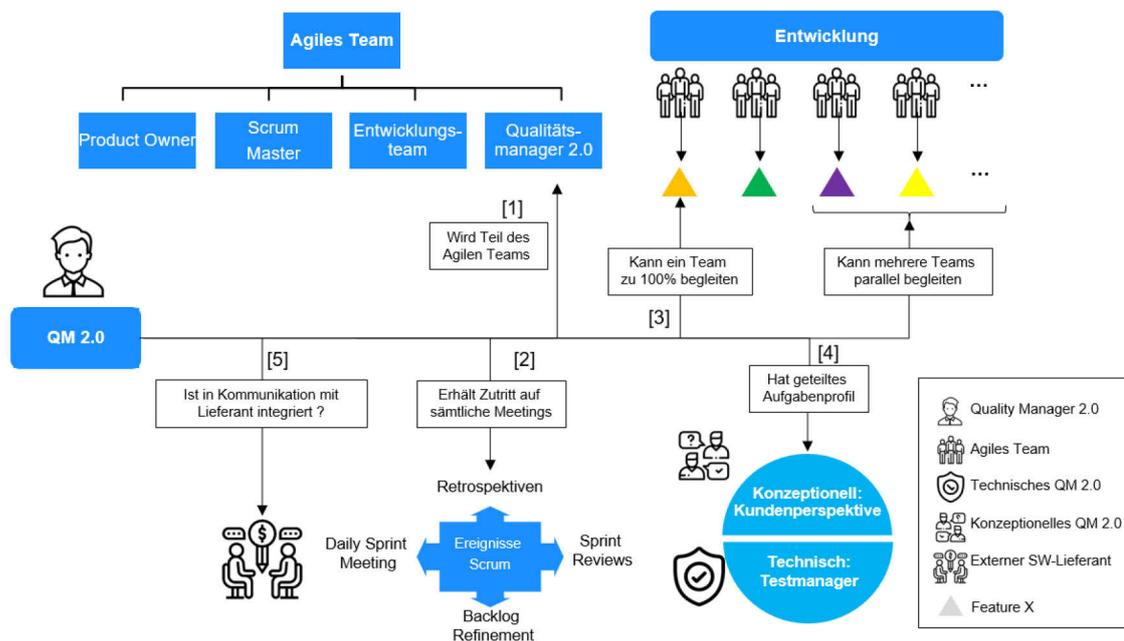


Abbildung 5.1: Einbindung der Rolle QM 2.0 in die Entwicklungsarbeit (Quelle: Eigene Darstellung, Forschungsstudie QSK der TU Berlin.)

Einer der Interviewpartner sieht die Notwendigkeit, die Rolle des QMs in das agile Team zu integrieren, um so einen engen Austausch zu ermöglichen (vgl. Anhang 22, Interview 4). Ringbauer<sup>[118]</sup> versteht eine Integration des QMs durch eine unterstützende Rolle im agilen Team ebenfalls als Option, wie das QM in die agile Methodik eingebunden werden kann. Weiter schlägt sie vor, die Rolle mit der des Scrum Masters zu verbinden.

Um den Austausch zwischen QM und agiler Entwicklung zu intensivieren, soll die erste Erweiterung der Rolle des QMs 2.0 also die Integration ins agile Team sein (Abb. 5.1).**[1]**

Damit bekommt der Qualitätsmanager 2.0 Einsicht in die Zeremonien und Ereignisse der agilen Arbeitsweise. Im Falle von Scrum sind das die Daily Sprints Meetings, Sprint Refinements, Retrospektiven und Reviews (Abb. 5.1).**[2]** Die Anwesenheit bzw. die Möglichkeit, die Regelmeetings zu besuchen, ermöglichen dem Qualitätsmanager 2.0,

Anforderungen und Anmerkungen ins Team einzusteuern sowie deren aktuelle Probleme und Themen zu verstehen.

In Anlehnung an die Rolle des POs, welcher oftmals mehrere Projekte gleichzeitig verfolgt, wird anhand des Umfangs und der Komplexität des Software-Projekts für jeden Einzelfall entschieden, ob der Qualitätsmanager mehrere agile Teams oder nur ein Entwicklungsteam gleichzeitig begleitet (Abb. 5.1).[3]

Die Empirie zeigte, dass der klassische Qualitätsmanager im Softwarebereich eher als qualitätssichernde Instanz in einer technischen Richtung verstanden wird. Der Testmanager kann neben den technischen Aufgaben auch Koordinations- und Managementaufgaben umfassen (vgl. Anhang 22, Interview 3). Im Automobilbereich wurde das QM vielmehr in einer konzeptionellen Rolle vorgefunden, welche die Kundenperspektive einnimmt (vgl. Anhang 22, Interview 2). Das QM 2.0 soll zukünftig die Rolle des qualitätssichernden Testmanagers und des kundennahen Qualitätsmanagers sinnvoll verbinden (Abb. 5.1).[4]

Wie ermittelt wurde, besteht die Softwareentwicklung zu weiten Teilen aus externen Lieferanten, welche durch Verträge eingebunden werden (vgl. Anhang 22, Interview 2). Innerhalb von Vertragsverhandlungen formulieren sowohl der PO als auch der Qualitätsmanager 2.0 Anforderungen, welche erfüllt werden müssen. Der PO definiert die technischen Anforderungen und der Qualitätsmanager 2.0 qualitätskritische und funktionale Anforderungen. Auch im Nachhinein wird durch diese beiden Rollen als Team die regelmäßige Kommunikation zum Lieferanten sichergestellt. Somit wird garantiert, dass eine kontinuierliche Abstimmung im Erstellungsprozess stattfindet (Abb. 5.1).[5]

### **5.2.2 Technische Rolle des Qualitätsmanagers**

In technischer Hinsicht begleitet der Qualitätsmanager 2.0 den PEP im Team durch das Management von Testing-Aktivitäten. Dabei gibt er Methoden wie Pair Programming oder Peer Reviews an das Entwicklungsteam weiter und sorgt dafür, dass diese verstanden und angewendet werden. Somit bindet er die Entwickler in die Testaktivitäten mit ein, wie das in agilen Methoden ohnehin der Fall sein sollte (Abb. 5.2).[6]

Außerdem schließt er sich zu festgelegten Zeitpunkten zwischen den Releases mit den Qualitätsmanagern der anderen Teams zusammen und sorgt damit für einen regelmäßigen, teamübergreifenden Austausch (Abb. 5.2).[7] Ziel dieser Regelkreise ist die produktübergreifende Koordination von Integrationstests, Regressionstests, der Automatisierung von Tests (vgl. Anhang 22, Interview 6) und Smoke Tests (vgl. Anhang 22, Interview 3) (Abb. 5.2).[8]

Dort auftretende Probleme und Fehler werden von ihm aufgenommen und an das agile Team weitergegeben. Im Sprint Review wird es hineingesteuert, um dort vom PO ins Product Backlog aufgenommen zu werden. Damit ist die Umsetzung der aufgenommenen Maßnahmen seitens der Entwicklung gesichert (Abb. 5.2).[9]

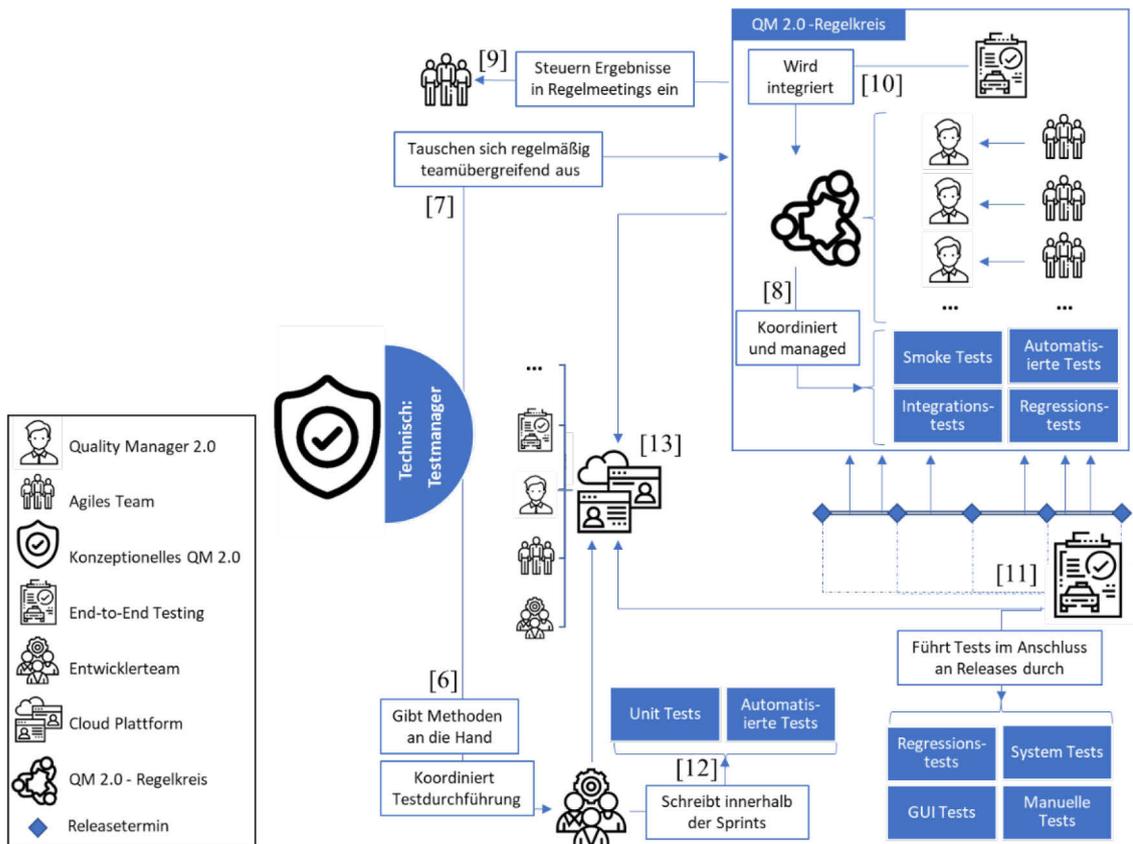


Abbildung 5.2: Technische Rolle des Qualitätsmanager 2.0 (Quelle: Eigene Darstellung, Forschungsstudie QSK der TU Berlin.)

Im QM 2.0-Regelkreis sind End-to-End Tester, welche die einzelnen Softwaremodule zum ersten Mal in ihrer Gesamtheit betrachten, ebenfalls anwesend (vgl. Anhang 22, Interview 1). Dadurch wird der aktuellen Situation von fehlenden Absprachen zwischen den Modulen (vgl. Anhang 22, Interview 1) entgegengewirkt, indem Anmerkungen bereits früh in den PEP hineingesteuert werden können. Dadurch ist es möglich, Verschwendung in Form von nachträglichen Änderungen oder Verschiebungen auf andere Releasezeitpunkte zu vermeiden (vgl. Anhang 22, Interview 1) (Abb. 5.2).**[10]**

Im Anschluss an Releasezeitpunkte beginnt das End-to-End Testing, welches über alle Systeme testet und manuelle Tests, Regressionstests und GUI-Tests anwendet.**[11]** Das vorgestellte Vorgehen hat an dieser Stelle das Potential, diese umfangreiche Phase zu reduzieren, indem weniger Tests durchgeführt werden müssen. Zusätzlich kann der Aufwand durch Nacharbeit in den einzelnen agilen Teams reduziert werden.

Das Aufgabengebiet des QMs 2.0 umfasst ausschließlich das Management und die Koordination von Tests. Die operative Umsetzung, sowie das Schreiben von Tests in Form von Unit Tests (vgl. Anhang 22, Interview 1) und Teile der Testautomatisierung (vgl. Interview 2) obliegen weiterhin dem Entwickler.**[12]**

Die durchgeführten Tests sowie auftretende Probleme werden auf einer Cloud-Plattform aufgenommen und dokumentiert. Das hilft, den großen Kommunikationsbedarf der agilen Arbeitsweise abzudecken. Die Test- und Kommunikationsplattform dient als Wissensplattform, indem für aufgetretene Fehlerbilder bereits Tests auf breiter Basis verfügbar gemacht werden. Außerdem ist die Plattform für alle Beteiligten entlang der Testpyramide nach Martin Fowler einsehbar und zugänglich. Das heißt, Informationen können sowohl von allen Fachbereichen (Entwicklern, Qualität, dem gesamten agilen Team und End-to-End Testing) eingesehen werden (vgl. Anhang 22, Interview 4). Aber auch das Management und Schnittstellen zum HW-Bereich können damit gelegt werden (vgl. Interview 4; vgl. Interview 2). Das schafft zum einen Vertrauen, dass Qualität berücksichtigt wird (vgl. Interview 4), und es verhindert, dass Tests doppelt durchgeführt werden. Die beschriebene Cloud-Plattform orientiert sich an einer digitalen Wissensplattform, wie sie auch im Lean Management bekannt ist.<sup>[119]</sup> Die Informationen sollen dabei möglichst in visueller Form dargestellt und ausgewertet werden, um gemäß dem agilen Ansatz sowie im Lean Management beschriebenen Methoden, Informationen schnell und übersichtlich zu teilen.<sup>[13]</sup>

### 5.2.3 Konzeptionelle Rolle des Qualitätsmanagers 2.0

In konzeptioneller Hinsicht übernimmt das QM 2.0 die Position des Kunden und unterstützt somit das agile Team bei der Entwicklung von kundennahen Produkten.

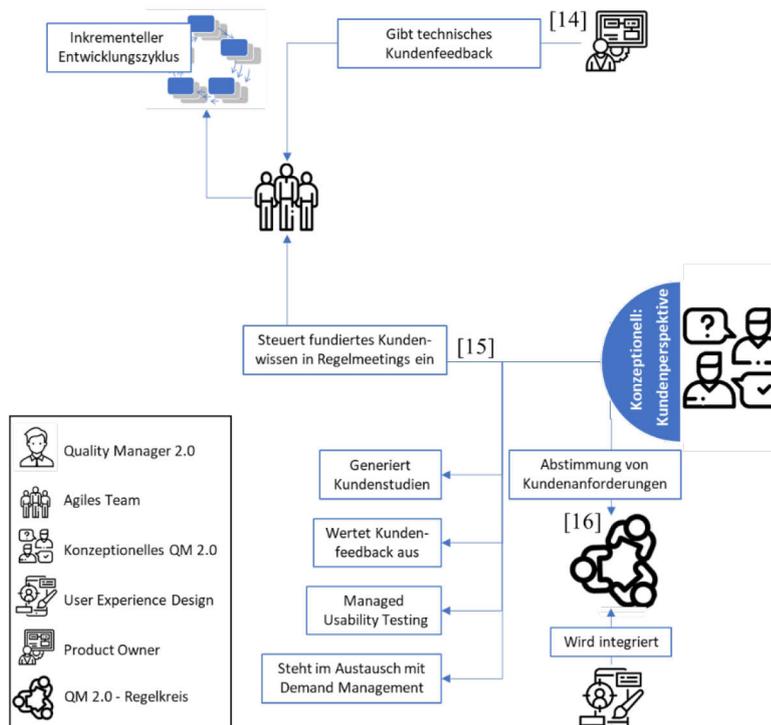


Abbildung 5.3: Konzeptionelle Rolle des Qualitätsmanagers 2.0 (Quelle: Eigene Darstellung, Forschungsstudie QSK der TU Berlin.)

Laut Definition soll der PO der Vertreter des Kunden sein und somit seine Interessen bzgl. der Funktionalität an das Produkt in den PEP einbringen.<sup>[120]</sup> Das geschieht durch

die Aufnahme von Anforderungen im Backlog und die systematische Umsetzung dieser im Verlauf des Projekts.

Die ISO 9000-Reihe verlangt die Erfüllung oder ein Übertreffen der Kundenanforderungen, lässt aber offen, wer der Kunde nun eigentlich ist (vgl. DIN EN ISO 9000). Bei den OEMs arbeiten sowohl externe Software-Unternehmen als auch interne Entwicklungsabteilung mit einem PO, welcher der Entwicklungsabteilung im Automobilunternehmen zugehörig ist (vgl. Interview 2; vgl. Interview 5). Die Product Owner erhalten die Informationen bzgl. neuer Produkte und Features vom Demand Management (vgl. Interview 1). Eine direkte Verbindung zum Endkunden besitzen die PO also nicht. Er vertritt somit im PEP die Anforderung der technikseitigen Entwicklung und nicht die des Endkunden (Abb. 5.3).[14]

Nach Angaben eines Interviewpartners fließen schon heute Kundenstudien seitens des QMs in die SW-Entwicklung ein, allerdings nicht in einem Regelprozess (vgl. Interview 2) Es erscheint logisch, dieses Kundenfeedback umfangreicher und prozessual gestützt zu verankern.

Mit den neuen Schnittstellen zum agilen Team kann das QM 2.0 mit seiner umfassenderen Sichtweise auf Kundenwünsche und die einzelnen Funktionalitäten im Connectivity-Spektrum innerhalb von Produkt-Reviews regelmäßig einwirken. Demnach wird ausgewertetes Kundenfeedback aus Befragungen und Marktstudien regelmäßig in die Softwareentwicklung eingebunden, wie von agilen Methoden gefordert. Der Qualitätsmanager 2.0 steht in seiner Position in regelmäßigen Kontakt mit dem Demand Management. Im Rahmen neuer Produktideen und Features kann er deren Feedback und produktübergreifendes Wissen ebenfalls in die Softwareentwicklung einbinden. Weiteres Kundenfeedback generiert er durch Usability-Tests in Form von Crowd Testing, was in Zukunft durch erweiterte Crowd Testing-Aktivitäten ergänzt werden kann (vgl. Interview 2).[15]

In seiner Position als Kundenanforderungsexperte tauscht der Qualitätsmanager 2.0 sich, wie schon in technischer Hinsicht, mit seinen Kollegen aus, um Synergieeffekte zu nutzen (vgl. Interview 2). Diese Plattform bietet die Möglichkeit, die Produkte hinsichtlich Designs und Funktionalität bereits in frühen Phasen abzustimmen. Die Abstimmung fokussiert den Frontend-Anteil, welcher für den Kunden später auch ersichtlich ist (vgl. Interview 1). In diesen Regelmeetings sind ebenfalls spezialisierte User Experience Designer anwesend, welche die konzeptionelle Sicht in der SW-Entwicklung umsetzen (vgl. Interview 3).[16]

Einige der in vorherigem Abschnitt aufgedeckten Probleme der Ansätze können nicht von der Rolle des Qualitätsmanagers gelöst werden, sondern benötigen ein Handeln des Managements auf einer höheren Hierarchiestufe. Im Rahmen der Unternehmenskultur wird ebenfalls die Führungskultur verstanden, da eine enge Bindung zwischen beiden besteht.<sup>[121]</sup>

### 5.3 Agile Organisationsstruktur schaffen

In Kapitel 5.2 wurde auf die zugrunde liegende Organisationsstruktur eingegangen und gezeigt, dass sich Softwarehersteller Organisationsformen mit flachen hierarchischen Strukturen bedienen. Marrold<sup>[122]</sup> sieht die Veränderung der Aufbau- und Ablauforganisation als unausweichliche Herausforderung für Unternehmen, um Marktentwicklungen schnell und flexibel zu adaptieren.

Eine flache Hierarchie hilft, Entscheidungen dezentral und daher zeitnah zu treffen. Damit entsprechen sie vielmehr der Geschwindigkeit von agilen Prozessen mit ihrem inkrementellen Charakter und der Berücksichtigung von dynamischen Anpassungen.

Im hierarchischen Aufbau sitzen die Entscheidungsträger entlang des kaskadenförmigen Aufbaus. Folglich können Entscheidungen nicht direkt von den direkten Beteiligten getroffen werden, sondern wandern entlang der Hierarchiestufen bis zur entsprechenden Person. Lange Entscheidungswege und hiermit verbunden lange Wartezeiten behindern die Geschwindigkeit von agilen Methoden. Zur Verdeutlichung hat ein Sprint eine Dauer von wenigen Wochen, indem ein Softwareinkrement in ein funktionstüchtiges Format entwickelt wird. Eine Entscheidungsdauer von wenigen Tagen würde folglich bereits einen Großteil des Sprints verzögern.

Am Softwareentwicklungsprozess sind mehrere Fachbereiche beteiligt. Organisatorisch sind diese in Funktionsbereiche untergliedert, was die Anzahl an Entscheidungswegen erhöht. Des Weiteren verursacht der funktionale Aufbau ein Silodenken und -handeln der einzelnen Funktionsbereiche.<sup>[123]</sup> Im agilen Team wird crossfunktional und ohne Hierarchie gearbeitet, was das Denken im Team fördert. Verbesserungsbestrebungen werden vom Gedanken der Gesamtsystemoptimierung geleitet und nicht von der Optimierung von Budget oder Prozessen einzelner Kostenstellen bzw. Abteilungen. Die funktionale Organisationsstruktur fördert einen Interessenkonflikt durch eigene Ziel- und Interessenverfolgung.<sup>[124]</sup>

Alternative Organisationsansätze mit wenig Hierarchie und direkten Entscheidungswegen wurden sowohl im Rahmen der Empirie als auch bei der Literaturrecherche ermittelt (vgl. Interview 2; vgl. Interview 3).<sup>[125]</sup>

In kurzfristiger Hinsicht kann die Organisation von agilen Teams in bereits existierenden Schwarmkonstellationen die funktionale Aufstellung im Konzern übergehen. In dieser bereits etablierten Arbeitsform arbeitet ein Projektteam aus Mitarbeitern verschiedener Fachbereiche hierarchieübergreifend und autonom zusammen (vgl. Daimler AG, 2016).

Langfristig gesehen ist eine Organisation der agilen Teams um Produkte in sogenannten Zellen sinnvoll (S. Abb. 5.4). In dieser produktzentrierten, zellularen Organisation ist die Hierarchie sehr flach ausgeprägt, sowohl organisatorisch also auch darin, wie sie von den Mitarbeitern gelebt wird. Teams organisieren sich um ein Produkt oder Projekt und agieren möglichst autonom, da benötigte Funktionen in einer „Produktzelle“ vorhanden sind. Innerhalb der Produktzellen organisieren sich die Teams selbst innerhalb ihrer Rollen und benötigen keine klassische Autorität, die sie anweist. Mit Sicht auf das

Unternehmen interagieren die einzelnen Produktzellen untereinander und können auch auf zentrale Fachbereiche, falls vorhanden, zurückgreifen. Obwohl es Linienmanager mit Mitarbeiterverantwortung gibt, macht der organische Aufbau eine ausgeprägte Führungsebene, welche auf dem Führungspyramiden-Prinzip beruht, obsolet.<sup>[126]</sup>

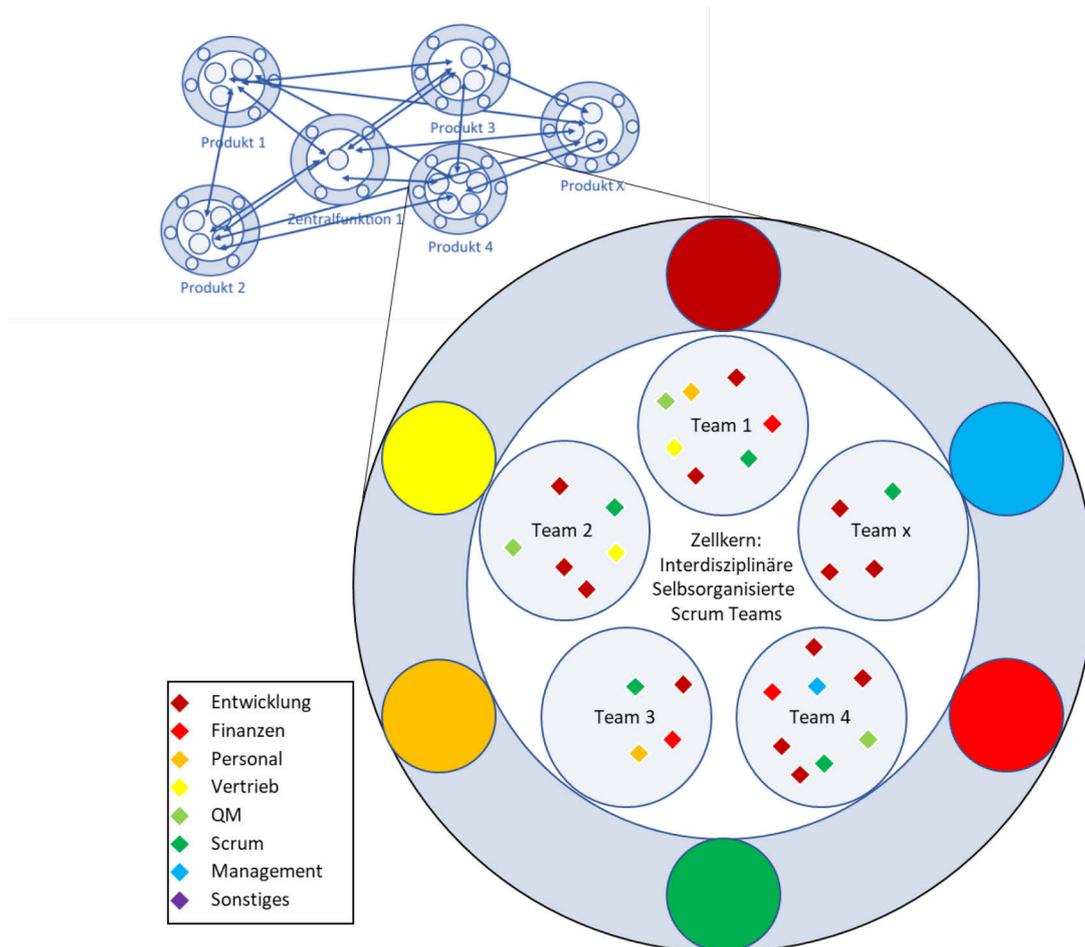


Abbildung 5.4: Produktzelle und zellulare Organisation<sup>[126]</sup>

Auch eine holokratische Aufstellung des Unternehmens oder von Teilen des Unternehmens wäre langfristig denkbar, da diese Agilität schafft und vorteilhaft im volatilen Umfeld ist. Jedoch ist die Einführung ein langfristiges Unterfangen, welches auch oftmals am Widerstand oder dem Unverständnis der Mitarbeiter scheitert.<sup>[122]</sup>

Die Holokratie ist ein Organisationsystem, welches anstelle von Funktionen Zirkel und Unterzirkel definiert, welche sich beliebig aus Themen unterschiedlicher Funktionsbereiche zusammensetzen können. Das kleinste Element in diesem System aus Zirkeln ist nicht der Mitarbeiter, sondern eine Rolle, welcher Aufgaben, Funktionen und Autoritäten zugeteilt werden. Es können von einem Mitarbeiter auch mehrere Rollen übernommen werden. In dieser Organisationsform gibt es wenig Hierarchie, da es keine Teams und Abteilungen mit direktem Weisungsbefugten gibt. Die Strukturen in der

Holokratie erlauben es, Rollen und Zirkel je nach Anforderung des Umfeldes dynamisch und kontinuierlich zu verändern. Eine feste Struktur gibt es also nicht.<sup>[122]</sup>

Die geschilderten sowie weitere hierarchiearme, agile Organisationsformen werden von Jeromin<sup>[125]</sup> gegenübergestellt und diskutiert.

### **Unternehmenskultur weiterentwickeln**

Neben der Organisationsstruktur liegt es im Macht- und Verantwortungsbereich der Führung, die bestehende Unternehmenskultur weiterzuentwickeln. Der Willen des Managements ist wesentlicher Faktor für eine langfristig erfolgreiche Etablierung neuer Aspekte in der Unternehmenskultur.<sup>[122]</sup> Maximini (2018) warnt im Zusammenhang einer ausbleibenden Transformation der vorherrschenden Kultur vor einem Kampf der Kulturen im Unternehmen.

Die in Kapitel 5.2 angestellten Untersuchungen ergaben, dass die Eingliederung von neuen Kompetenzen im Softwarebereich bereits zu Unverständnis zwischen einzelnen hardwarelastigen und softwarelastigen Fachbereichen führt. Zum einen liegen diese darin begründet, dass das Vorgehen in der agilen Arbeitsweise nicht verstanden wird (vgl. Interview 2). Zum anderen ist es auf die unterschiedlichen Aspekte der Unternehmenskultur, welche von den Mitarbeitern gelebt werden, zurückzuführen (vgl. Anhang 22, Interview 3).

OEMs haben erkannt, dass die Unternehmenskultur an das dynamische Umfeld angepasst werden muss und erste Schritte zum entsprechenden Kulturwandel eingeleitet (vgl. Daimler AG 2016).

Wie in Tabelle 5.1 gezeigt, vereint die neue Führungskultur in gleichen Maße Werte und Prinzipien aus dem agilen Umfeld und dem Lean Management Umfeld. Die Wichtigkeit von Aspekten wie Befähigung der Mitarbeiter, Übergabe von Verantwortung, Experimentierfreude, Lernbereitschaft und der Wille, aus Fehlern zu lernen, ist in gleicher Weise bei den Automobilbauern als auch der Literatur zu agiler Unternehmenskultur zu entnehmen (vgl. Daimler AG 2016).<sup>[122][126]</sup>

In der Unternehmenspraxis ist es modern, das Wort Agilität zu nutzen, und doch wissen wenige, was es bedeutet, und interpretieren es auf eigene Weise, weshalb es die FAZ zu einem Unwort des Jahres 2018 kürte (vgl. Köhn 2018; Seedig 2018). Auch die Empirie zeigte, dass die Aufklärung von Mitarbeitern an der Schnittstelle zur agilen SW-Produktion hilft, Problemen vorzubeugen (vgl. Interview 3). Folglich muss das Management für ein breites, einheitliches Verständnis über kulturelle Aspekte, aber auch zum großen Teil über Prozesse der agilen Arbeitsweise in der Belegschaft sorgen.

Bei der Vermittlung sollte aufgezeigt werden, wie eng sich agile Arbeitsweise an den Prinzipien des Lean Managements orientiert. In Kapitel 5.2 Prozessuales Vorgehen wurden hierzu die Leitgedanken, welche der Prozessgestaltung zugrunde liegen, aufgezeigt. Ein Unverständnis zwischen den Mitarbeitern ist also nicht in unterschiedlichen Grundsätzen zu sehen, sondern es fehlt die genaue Kenntnis, wie

agile Methoden vorgehen. Im Zuge der Aufklärung über das Vorgehen agiler Methoden müssen weiter vorherrschende Vorurteile über agiles Arbeiten bekämpft werden.<sup>[106][127]</sup>

## 5.4 Zusammenfassung und Fazit

Die agilen Arbeitsweisen werden in der Automobilindustrie als Projektmanagementmethode verstanden, was dazu führt, dass kulturelle Werte der Belegschaft aus dem Softwareumfeld nicht ausreichend berücksichtigt werden. Der Bereich Connectivity befindet sich noch im frühen Stadium des Aufbaus und es wurden bzw. werden Mitarbeiter mit einem softwarelastigen Background eingestellt. Die Werte und Vorstellungen der Neueinstellungen stehen im Kontrast zu den Mitarbeitern, der bereits bestehenden hardwarelastigen Abteilungen, was in Unverständnis zwischen den Bereichen resultiert. Das unterschiedliche Mindset der Mitarbeiter und ihre gewohnte agile Arbeitsweise werden vom Automobilbereich nicht durch eine Anpassung der Organisationsstruktur mit kurzen Entscheidungswegen und flacher Hierarchie berücksichtigt. Die Untersuchungen zeigten, die agile Arbeitsweise in Form der Softwareentwicklung sowie das QM werden in die bereits bestehende funktionelle Organisation aufgenommen. Problematisch zeigte sich in dieser Hinsicht, dass das QM nicht ausreichend in die schnelleren agilen Entwicklungsprozesse integriert ist. Softwarehersteller neigen zu agilen Organisationsformen, wie z. B. der Holokratie, welche schnelle Entscheidungsfindung durch schwach ausgeprägte Hierarchien ermöglicht und damit die agile Kultur unterstreicht.

Der Kunde und dessen Anforderungen sind im traditionellen QM und agilen Methoden ausgeprägt verankert. Die agilen Arbeitsweisen realisieren den Kundenfokus durch die regelmäßige Einbindung von Kundenfeedback in die Softwareerstellung, wohingegen im Hardwarebereich die Kundenanforderungen punktuell zu Beginn der Entwicklung einfließen. In der Softwareentwicklung im Connectivity-Bereich ist der Kunde der PO, welcher mit internen oder externen agilen Teams zusammenarbeitet. Problematisch zeigte sich in dieser Hinsicht, dass der PO im Funktionsbereich Entwicklung tätig ist und in dieser Position primär die technischen Anforderungen vertreten kann. Folglich können die Anforderungen des Endkunden, welcher die Connectivity-Produkte nutzt, durch den PO nicht an das agile Team weitergegeben werden. Das QM besitzt durch seine Funktion eine Übersicht der Connectivity-Dienste und ermittelt die Anforderung des Nutzers durch Kundenstudien. In dieser Rolle lässt es aktuell bereits Endkundenanforderungen unregelmäßig in die Softwareentwicklung einfließen. Bei den befragten Softwareherstellern übernimmt die Rolle des POs ebenfalls die Kundenperspektive im agilen Team. Die befragten Unternehmen hatten allerdings die Stellung eines Lieferanten und somit war der Auftraggeber auch gleichzeitig der Kunde, was die Frage nach der Kundenrolle obsolet machte.

Der Bedeutung von kontinuierlicher Verbesserung wird sowohl in QM-Systemen als auch agiler Arbeitsweise durch zyklisch verlaufende Prozesse gemäß dem PDCA-Zyklus Rechnung getragen. Im untersuchten Connectivity-Bereich konnten Verbesserungsbestrebungen in Form von crossfunktionalen Lessons-Learned-

Sitzungen, welche alle acht bis zwölf Wochen stattfinden, nach Releases erkannt werden. Ob und wie dort identifizierte Probleme durch Maßnahmen umgesetzt werden, konnte nicht geklärt werden. Zur Verbesserung innerhalb der agilen Entwicklung finden nach Sprintende Produkt-Reviews statt. Die erwähnte Trennung der Kundenperspektive, in technische Anforderungen und Nutzeranforderungen auf unterschiedlichen Rollen, limitiert das Verbesserungspotential in Produkt-Reviews auf technische Aspekte. Das QM partizipiert nicht in Regelmeetings des agilen Teams, weshalb Feedback hinsichtlich der Nutzeranforderung nicht eingebracht wird. Zusätzlich zu den bekannten Regelmeetings konnte bei den Softwareherstellern der crossfunktionale Austausch über Abteilungs- oder Unternehmens-Retrospektiven festgestellt werden. In Kombination mit einer agilen Organisationsstruktur können hier teamübergreifende Probleme durch die Anpassung von Rollen und Verantwortlichkeiten gelöst werden.

Die Empirie zeigte, dass sowohl Mitarbeiter der OEMs als auch jene im agilen Unternehmen - regelmäßig - ihren Tätigkeitsgebieten angemessen weitergebildet werden. Die Struktur der Fortbildung für Softwareentwickler im Bereich Connectivity wurde nicht gesondert betrachtet. Die Steuerung von Fortbildungsmaßnahmen wird durch die sorgfältige Einstellung von erfahrenem Personal ergänzt. Des Weiteren wurde erkennbar, dass Schnittstellenfunktionen durch Teams mit differentem Expertenprofil besetzt werden. In Anlehnung an soziale Aspekte der Mitarbeiterorientierung, wie von agilen Methoden behandelt, wurden bei einem Automobilhersteller die Umgestaltung der Arbeitsräume in der Unternehmensstrategie verankert. In der Untersuchung dieser Arbeit wurde stark auf die fachlichen Kompetenzen sowie die Ausbildung der Mitarbeiter fokussiert. Folglich wurden im Rahmen der Empirie keine Unterschiede in der generellen Mitarbeiterzufriedenheit aufgenommen. Auch die Möglichkeiten von Weiterentwicklung in Form von Positionswechsel, Führungskarriere oder zusätzlichen Aufgaben wurde nicht abgedeckt. Abschließend wurde in Hinsicht auf die Mitarbeiterorientierung kein Problembereich identifiziert, welcher zu Konflikten führt.

Bei einem führenden Automobilhersteller wurde die Notwendigkeit der zukünftigen Anpassung der klassischen Führungsrolle erkannt und in die Unternehmensstrategie integriert. Obwohl diese weiterhin als hierarchische Instanz zu verstehen ist, sind einige Tendenzen zu erkennen, welche den Leitsätzen der Agilität entsprechen. So werden crossfunktionale Teams hervorgehoben, welche in Schwarmkonstellationen frei von Hierarchie sind und autonom arbeiten können. Mit der Agilität wird ein Prinzip aufgenommen, welches eine offene Unternehmenskultur fordert, die neue, innovative Lösungen in Betracht zieht. Lösungen abseits alter Wege werden also vom Top-Management unterstützt. Aktuell werden die agilen Arbeitsweisen in die hierarchische Struktur aus Funktionsbereichen und mit klassischen Führungspositionen integriert. Das Rollenmodell, welchem sich agile Methoden bedienen, ist im jeweiligen Entwicklungsprojekt zu finden, ordnet sich allerdings unter das hierarchische System. Es existiert also aktuell eine Hybridform aus klassischer hierarchischer Führung und agilen Rollen. Das QM wird in agilen Methoden nicht mit einer Rolle berücksichtigt. Daraus folgt, dass die Zusammenarbeit mit dem agilen Team nicht prozessual definiert

ist. Die fehlende Integration des QMs als Rolle macht sich im Automobilbereich durch fehlende Schnittstellen und unregelmäßige Einbindung in das agile Team bemerkbar. Die Empirie in der SW-Industrie zeigte auch hier kein vollkommen hierarchiefreies Modell, allerdings ist die Verteilung der Macht auf Rollen dort bereits weiter fortgeschritten. Das klassische QM wird hier als Testmanagement begriffen und als eigenständige Rolle in das agile Team aufgenommen.

Die Theorie zeigte eine Übereinstimmung hinsichtlich einer offenen Einstellung gegenüber Fehlern in QM-Systemen und den agilen Prinzipien. Fehler werden als Chance zur Verbesserung anerkannt. In der Praxis zeigte sich bei den OEMs, dass das QM im SW-Bereich eine höhere Toleranzschwelle für Fehler aufweist, als es im HW-Bereich zu verzeichnen ist. Grund dafür ist die Möglichkeit, Fehler, welche noch beim Kunden entstehen, durch das Einspeisen von Updates oder Wartungen zu lösen. Hierdurch entstehen dem Unternehmen keinerlei Kosten durch Rückrufe, Umbaumaßnahmen in der Werkstatt oder Reputationsschäden. Neben Kosten spielt der Faktor Geschwindigkeit eine Rolle, da durch die schnellen Entwicklungszyklen der agilen Arbeitsweise Updates zeitnah generiert werden. Die Empirie zeigte, dass die Anzahl von SW-Fehlern, die beim Kunden auftreten, bemerkenswert höher ist, als die Fehler von HW-Teilen nach Verkauf. Ob diese auf die Fehlerkultur im SW-Bereich, die charakteristische SW-Architektur oder die fehlende Abstimmung zwischen den agilen Teams zurückzuführen ist, konnte nicht geklärt werden. In diesem Zusammenhang zeigten sich Probleme in Bezug auf die langfristige Verantwortlichkeit für SW-Fehler verschiedener Features. Diese Verantwortlichkeit ist aktuell im Automobilbereich nicht geklärt, weshalb Widerstände zwischen Fehlermanagement und SW-Entwicklung aufkommen. Darauf aufbauend zeigte sich die Notwendigkeit der langfristigen Wartung und Update-Generierung hinsichtlich des Automobillebenszyklus. Das Ausmaß dieser Herausforderung scheint aufgrund des relativ langen Produktlebenszyklus des Automobils noch nicht absehbar zu sein.

Agile Methoden orientieren sich nah an den prozessualen Gestaltungsprinzipien des Lean Managements. Das One-Piece-Flow, Pull-Prinzip, Taktung und Kanban werden in den agilen Prozessen berücksichtigt, was einen Aufbau gemäß dem Qualitätsdenken deutlich macht. In prozessualer Hinsicht zeigte die Empirie, dass es die kurzen, abgeschlossenen Zyklen in der SW-Entwicklung dem QM schwer machen, sinnvoll durch vordefinierte punktuelle Eingriffe, wie aus der HW-Entwicklung bekannt, einzuwirken. Außerdem konnte festgestellt werden, dass die agile SW-Entwicklung einen äußerst heterogenen Aufbau besitzt, sowohl strukturell als auch bzgl. der Anwendung von Methoden. Hinsichtlich der Struktur besteht die SW-Entwicklung aus internen und größtenteils externen Teams. Die externe Seite setzt sich wiederum aus einer Lieferantenbasis bestehend aus Start-ups, SW-Herstellern und Tochterfirmen zusammen. Hinsichtlich angewandeter Methoden werden im SW-Feld unterschiedliche agile Verfahrensmodelle wie Scrum, Kanban oder anderes verwendet. Diese unterscheiden sich in ihrer Methodik und Aufbau teilweise deutlich voneinander. Die beschriebene Vielschichtigkeit stellt das QM vor weitere Schwierigkeiten. Weitere

Problematik, welche sich für eingebettete Systeme als charakteristisch erweist, ist die Bindung der SW-Anforderungen an die Anforderungsaufnahme von HW-Komponenten und damit an langwierige HW-Entwicklungsprozesse. Wie gezeigt wurde, beschränkt die langfristige HW-Planung die SW-Entwicklung hinsichtlich der Anforderungsänderungen bereits in einer frühen Phase. Um die starke Bindung zwischen den HW- und SW-Komponenten zu trennen, werden in der Automobilindustrie Plattformstrategien angewendet, welche die Smartphone-Industrie zum Vorbild haben. Die SW-Hersteller gliedern einen Qualitätsbeauftragten ins agile Team ein und schaffen so die Abstimmung mit der Entwicklung innerhalb schneller Zyklen. Auch hier herrscht Heterogenität durch die Einflechtung von Lieferanten und verschiedener Methoden. Auf Probleme, welche dadurch entstehen, wurde nicht tiefer gehend eingegangen. Allerdings wurde auf eine intensive Abstimmung zwischen den einzelnen Entwicklerteams hingewiesen, die durch den Qualitätsbeauftragten koordiniert wird. Des Weiteren wurde deutlich, dass die Qualitätssicherung neben dem Qualitätsbeauftragten bzw. Testmanager auch seitens der Entwickler durch das Schreiben von Tests abgedeckt wird. Die Testphase ist ein Teil jedes Sprints, womit die Entwicklung kontinuierlich in das Testing integriert ist.

Wie in den agilen Werten beschrieben, favorisieren agile Methoden die direkte Kommunikation anstelle der Dokumentation. Ein Austausch von Informationen wird also bevorzugt über den Dialog abgehandelt als über eine ausführliche Dokumentation des Vorgehens und der Ergebnisse. Wie bei SW-Herstellern festgestellt wurde, spielt daher die regelmäßige Kommunikation eine bedeutende Rolle. Was innerhalb des Teams durch verschiedenste Regelmeetings gut funktioniert, zeigte in der Untersuchung Herausforderung an den Schnittstellen zu anderen Bereichen. Die Qualitätssicherung weist Schwachstellen besonders bei der teamübergreifenden Arbeit auf, was sich in einer hohen Fehlerdichte in der End-to-End Phase zeigt. Verstärkt wird die fehlende Abstimmung durch fehlende zentrale Plattformen, durch welche Ergebnisse, aus bereits durchgeführten Tests, zwischen den Teams und Abteilungen kommuniziert werden können. Den an der Qualitätssicherung Beteiligten, wie Entwickler, Testmanager und End-to-End Tester, mangelt es somit an Transparenz über die Historie bereits angewendeter Tests, was zu Mehrarbeit führt. Des Weiteren hinterlässt fehlende Abstimmung auf teamübergreifender Ebene, verbunden mit der minimalistischen Dokumentation an den Schnittstellen zur agilen Entwicklung, ein Bild, welches die agile Arbeitsweise als unverbindlich darstellt. Wie aus der Literatur bekannt, entstehen Vorurteile, die die agile Arbeitsweise als chaotisch beschreiben. Die Antwort der SW-Hersteller auf mangelnde teamübergreifende Abstimmung ist die kontinuierliche Kommunikation über Teamgrenzen hinweg, welche durch Qualitätsbeauftragte regelmäßig hineingesteuert werden. Um die Kommunikation zu intensivieren, konnten ebenfalls einheitliche, digitale Tools wie Jira identifiziert werden, welche für einen gemeinsamen Informationsstand sorgen.

Die Entscheidungsfindung wird, wie von QM-Systemen gefordert, durch die Anwendung von Kennzahlen gestützt. Mithilfe von Story Points wird die Arbeit schätzbar gemacht und liefert damit Optionen für die Planung von Sprints und Releases. Methoden wie

Sprint Burndown - und Release Burndown-Grafiken zeigen dabei den Fortschritt innerhalb des Sprints bzw. im Projekt auf. Darauf aufbauend deckte die Analyse keine Differenzen zwischen dem QM und agiler Arbeitsweise auf.

Weite Teile der SW-Entwicklung sind an externe SW-Firmen ausgelagert. Dieses Bild ist sowohl bei OEMs als auch bei befragten SW-Herstellern zu finden. Der SW-Entwicklungsbereich in den Automobilfirmen ist noch relativ jung und weiterhin am Wachsen. Nichtsdestotrotz besteht die Gefahr von langfristig fehlenden internen SW-Kompetenzen. Die Lieferanten werden mittels verschiedener Vertragsarten in die SW-Entwicklung eingegliedert. Je nach Vertragskonstellation gibt es einerseits enge Formen der Zusammenarbeit zwischen OEM und andererseits gibt es noch die klassischen Formen mit Lasten- und Pflichtenheft, welche die agile Arbeitsweise stark einschränken. Enge Formen der Zusammenarbeit weisen sich durch einen PO beim Kunden aus, welcher regelmäßig Feedback an das agile Team liefert. Qualitätskriterien können zwar vertraglich festgelegt werden, im Hinblick auf Anwendungen allerdings nur beschränkt. Die Notwendigkeit einer regelmäßigen Abstimmung mit dem Lieferanten hinsichtlich der Qualitätskriterien nach Vertragsschluss wird also deutlich. Diese Abstimmung findet aktuell durch den PO statt, welcher in dieser Rolle technisches Feedback einfließen lassen kann. Das QM dagegen ist nicht in die weitere Zusammenarbeit über Unternehmensgrenzen involviert, womit Qualitätskriterien nicht erneut einfließen können. Weiterhin problematisch erwies sich die beschriebene Herausforderung der langfristigen Wartung und Updateversorgung im Zusammenhang mit SW-Produkten, welche vom Lieferanten entwickelt wurden. Bei fehlendem internen Know-how wurde deutlich, dass für die Bereitstellung von Updates klare Verantwortlichkeiten geschaffen werden müssen.

Die Beantwortung der ersten Forschungsfrage zeigt, dass die agilen Methoden die einzelnen Aspekte von QM-Systemen berücksichtigen und in einer eigenen Art interpretieren. Dabei gibt es einige Anlehnungen an das Lean Management und dessen Prinzipien, was zu Überschneidungen in deren Fundament führt. In Bezug auf die Situation bei OEMs sind die größten Differenzen zu agilen Denkansätzen in der Art der Führung, dem Aufbau der Organisation sowie dem Ablauf der Entwicklung zu sehen. Die vorgefundene IST-Situation: ein hierarchischer Führungsstil, ein funktionaler Aufbau der Organisation und eine sequentielle wasserfallartig verlaufende Entwicklung mit langem Planungshorizont. Dagegen erfordert die agile Methodik: Dezentrale Führungsinstanzen, agile Organisationsstrukturen mit kurzen Entscheidungswegen und eine inkrementelle Entwicklung, welche die Berücksichtigung von Anforderungsänderungen erlaubt. Das QM für den SW-Bereich bleibt in seiner Struktur unverändert und folgt dem QM im HW-Bereich, was den Anforderungen der agilen Arbeitsweise nicht gerecht wird. Die Folgen sind fehlende Abstimmung und Integration des QMs in die agile Prozessstruktur, was als primäres Problem identifiziert wurde.

Die Rolle des Qualitätsmanagers wurde in der Rolle des QMs 2.0 in das agile Team integriert und seinen Kompetenzen auf deren Arbeitsweise abgestimmt. Den wenigen Schnittstellen zum agilen Team, welche sich in Problemen hinsichtlich der Abstimmung zwischen QM und agilem Team zeigten, wurde durch die Integration des

Qualitätsmanagers in das agile Team begegnet. Auf diese Weise hat der Qualitätsmanager die Möglichkeit, an den Regelmeetings während der schnellen zyklischen Entwicklungsphasen teilzunehmen und Anforderungen direkt in die Prozesse der agilen Methodik einfließen zu lassen. Abstimmung, welche vorher in sehr langen Abständen vorzufinden war, kann nun entlang des gesamten PEP kontinuierlich stattfinden. Wie das für PO oder Scrum Master der Fall sein kann, soll auch der Qualitätsmanager 2.0 in der Lage sein, mehrere Projekte gleichzeitig zu coachen. Je nach Umfang und Komplexität des Projektes kann der Qualitätsmanager mehrere Teams unterstützen. Dadurch bekommt er eine Übersicht über verschiedene Funktionen, was die Abstimmungen auf crossteam Ebene positiv beeinflusst. Die interne SW-Entwicklung stellt einen kleinen Teil der Gesamtwertschöpfung dar. Der Großteil findet im externen Umfeld statt und liegt somit beim Lieferanten. Das Einflussgebiet des Qualitätsmanagers wird auf das externe Umfeld ausgeweitet, um auch die Möglichkeit zu gewinnen, auf die Zusammenarbeit mit Lieferanten Einfluss zu nehmen. In Vertragsverhandlungen sollen Qualitätsaspekte der SW mithilfe des Qualitätsmanagers festgelegt werden. Eine umfassende Absicherung und Beschreibung mittels Verträge sind in der agilen SW-Entwicklung nicht sinnvoll, weshalb der Fokus auf der kontinuierlichen Zusammenarbeit mit dem externen agilen Team liegt. Während der Entwicklungsarbeit bildet aktuell ausschließlich der PO die Schnittstelle zum Lieferanten ab. Der Austausch über technische Spezifikationen besteht damit bereits, jedoch fehlt die konzeptionelle Sichtweise auf die Kundenwünsche, welche in der Funktion des Qualitätsmanagers abgebildet ist. Um diese in die zukünftige Regelkommunikation aufzunehmen, bildet der Qualitätsmanager 2.0 gemeinsam mit dem PO die Schnittstelle zum SW-Lieferanten. Weitere Anpassung, die der Qualitätsmanager 2.0 enthält, ist eine Aufteilung in die technische und die konzeptionelle Rolle.

In seiner technischen Ausrichtung übernimmt der Qualitätsmanager 2.0 die Rolle eines Testmanagers, der die Durchführung von Tests entlang des gesamten PEP koordiniert und leitet. In der agilen Entwicklung coacht er die SW-Entwickler bei der Anwendung von Methoden wie Peer Programming, Peer Reviews und gemeinsame Verantwortlichkeit für geschriebenen Code. Zwischen den agilen Teams koordiniert er die Termine für gemeinsame Testphasen und bindet hierfür alle notwendigen Parteien wie End-to-End Testing etc. ein. Die späte End-to-End Testphase mit hohem Aufwand und hoher Fehleranzahl wird durch die regelmäßige Abstimmung schlanker gemacht. Zweite Regelkommunikation bildet ein produktübergreifender Regelkreis zwischen den Testmanagern in den agilen Teams. Während in dieser Runde Testaktivitäten abgestimmt werden, kommen auch konzeptionelle Themen zur Ansprache, was an die konzeptionelle Rolle des Qualitätsmanagers 2.0 anknüpft. Die koordinativen Aktivitäten des Qualitätsmanagers 2.0 werden zuletzt durch eine digitale Plattform entlang des gesamten PEP ergänzt. Mit der Plattform soll die Testhistorie für alle Parteien ersichtlich gemacht werden und sie soll ebenfalls als Sprachrohr zum Management und der HW-Entwicklung dienen.

Die konzeptionelle Rolle des Qualitätsmanagers 2.0 spannt die Brücke zu einem SW-Produkt, welches den Nutzerwünschen entspricht. Die Funktion entspringt der Problematik, dass der PO allein die Nutzeranforderungen nicht mit seinem Kompetenzprofil abdecken kann und diese dadurch in der agilen SW-Entwicklung benachteiligt werden. Die Sichtweise auf die Endkundenerwartungen erhält der Qualitätsmanager 2.0 durch Kundenstudien und Usability-Tests sowie deren Auswertungen. Diese bringt er als Mitglied des agilen Teams innerhalb von Product Reviews als Anforderungen in die agile Entwicklung ein. Das regelmäßige Product Review eignet sich hervorragend zum Einsteuern von Nutzerfeedback, da es auf einem Prototyp des Produkts basiert. Nutzerfeedback wird bei internen Teams und außerdem in Kooperation mit dem PO bei externen Teams eingebracht. Um den Features den Charakter eines abgestimmten Gesamtkonzepts zu geben, tauschen sich die Qualitätsmanagers 2.0 in ihrer konzeptionellen Rolle, wie schon in der technischen, im Qualitätsmanager 2.0 Regelkreis aus. Hier werden die Funktionen und das User Interface diskutiert und abgestimmt. Die Ergebnisse können anschließend ins agile Team hineingesteuert werden. Gemäß den Anforderungen der inkrementellen Entwicklungszyklen können die Anforderungen in regelmäßigen Abständen berücksichtigt werden.

Die Arbeit zeigte auch Herausforderungen, deren Lösung nicht im Machtbereich der Rolle des Qualitätsmanagers liegt, sondern im Rahmen einer Veränderung des QM-Systems, die von der Unternehmensführung umgesetzt werden sollte. Eine starre Organisationsform, welche Änderungen schwer zulässt und mit steilen Hierarchien und langen Entscheidungswegen zu kämpfen hat, limitiert agile Arbeitsweisen und muss langfristig durch agile Organisationsformen ersetzt werden. Die agilen Werte und Prinzipien werden nicht aktiv in die Unternehmenskultur aufgenommen, da agile Methoden mit dem Charakter einer Projektmanagementmethode gleichgesetzt werden. Die bereits etablierte „Wasserfall“-Kultur muss Schritt für Schritt mit agilen Werten ergänzt bzw. ersetzt werden, um eine Zwei-Klassen-Kultur zu verhindern. Der geplante potentielle Wandel der Unternehmenskultur muss in der Belegschaft kommuniziert werden und als langfristiges Unterfangen in die Unternehmensstrategie aufgenommen werden. Kurzfristig muss über Agilität und agiles Arbeiten aufgeklärt werden. Vorurteile und eigene Interpretationen behindern die Zusammenarbeit und können sich zu Konflikten entfalten.

Die zweite Forschungsfrage wurde mit der Herleitung eines Modells, welches den zukünftigen agilen Qualitätsmanager 2.0 beschreibt, beantwortet. Das Modell integriert den Qualitätsmanager 2.0 durch seine Aufnahme als Rolle im agilen Team besser in deren Arbeitsweise. Es verschafft dem Qualitätsmanager durch regelmäßige Regelabstimmungen den Einblick in die agile Entwicklung. Der Austausch berücksichtigt nun die schnellen Zyklen und ist nicht länger auf wenige Zeitpunkte begrenzt. Auf diese Weise kann er regelmäßig seine Kompetenzen in technischer und konzeptioneller Sichtweise durch Anforderungen in die Entwicklung einfließen lassen. Über Unternehmensgrenzen hinweg bildet der Qualitätsmanager 2.0 in Kooperation mit dem

PO die Schnittstelle zum SW-Lieferanten. Gemeinsam steuern sie so ein umfassendes Anforderungsgefüge aus technischer und endkundenorientierter Perspektive in die externe Entwicklung. Teamübergreifende und produktübergreifende Abstimmung koordiniert der Qualitätsmanager 2.0 durch Regelkreise, welche die Qualitätsmanager 2.0 aus den einzelnen Teams sowie involvierte Funktionsbereiche umfassen. Auf diese Weise kann Wissen und Informationen bzgl. Testaktivitäten und Nutzerwünschen schnittstellenübergreifend im agilen SW-Entwicklungsprozess geteilt werden.

## 5.5 Ausblick

Die Interviews aus Teilen des Connectivity-Bereichs untersuchen das Vorgehen bei einem deutschen Automobilbauer und decken somit nicht das Gesamtbild der Branche ab. Um demnach Aussagen über das Gesamtfeld zu tätigen, sind weitere Interviews bei unterschiedlichen Automobilbauern und Abteilungen/Teams durchzuführen. Die Herausforderungen in diesem Feld stellen alle Automobilbauer vor Probleme, weshalb ein Interesse dieser an gemeinsamen Untersuchungen und Lösungen bestehen sollte. Die Untersuchungen zeigen den Status quo bei einem etablierten globalen Player und sind somit aussagekräftig für große Unternehmen, welche bereits historischen Charakter im Automobilbau besitzen. Im Gegensatz dazu gibt es neue internationale Wettbewerber wie Tesla oder XPENG Motors, welche sich auf intelligente Elektrofahrzeuge fokussieren. Die Prozesse und Unternehmensstrukturen sind im Vergleich zur deutschen Automobilindustrie noch im Anfangsstadium und lassen jedoch flexiblere Strukturen vermuten. Das vernetzte Automobil ist auch dort kein Fremdwort und es können sicherlich weitere Erkenntnisse aus deren Strategien gewonnen werden.

Der untersuchte agile SW-Hersteller, welcher befragt wurde, ist ein kürzlich eingegliedertes Tochterunternehmen eines Automobilherstellers. Das Unternehmen ist in seiner Struktur und Arbeitsweise nach Eingliederung weitestgehend unverändert geblieben. Wie festgestellt wurde, zeigen Lieferanten deutliche Unterschiede in der Interpretation der Kundenperspektive. Weitere Untersuchungen bei agilen Unternehmen, welche direkten Nutzerkontakt besitzen, können somit weitere Forschungserkenntnisse in Bezug auf die Einbindung von technischen Anforderungen und Nutzerfeedback liefern. Des Weiteren kann die Untersuchung der dort vorzufindenden Organisationsstruktur, Ausbildung von Hierarchie und Unternehmenskultur weitere Erkenntnisse bzgl. der Auswirkungen auf Anwendung agiler Methoden aufzeigen.

Der festgestellte hohe Grad externer Wertschöpfung verbunden mit der hohen Anzahl an Fehlern bei Integration von SW-Komponenten seitens verschiedener Entwicklerteams zeigt die Bedeutung eines systematischen Managements sowie Integration von Lieferanten. Im Bereich der Beschaffung von HW-Komponenten ist bekannt, dass Lieferanten in einem systematischen Ansatz die Phasen der Analyse, Bewertung, Klassifizierung und Entwicklung durchlaufen.<sup>[128]</sup> Außerdem helfen unabhängige Zertifizierungen, wie die ISO 9000-Reihe, einheitliche Qualifikationsstandards bei Lieferanten festzulegen. Die homogene Struktur aus Start-ups und generell unterschiedlichen Unternehmensgrößen lässt die Vermutung zu, dass

dort weitreichende Unterschiede in den Prozessen und Methoden zu erwarten sind. In diesem Zusammenhang ist auch hinsichtlich Qualitätsstandards eine unterschiedliche Handhabung zu erwarten. Einheitliche Qualifikationsstandards sowie -zertifizierungen gibt es für die unterschiedlichen Unternehmenstypen nicht. Im Zusammenhang mit Lieferanten wird die Frage nach der Rolle des POs aufgeworfen. Er vertritt die Interessen des Kundenunternehmens regelmäßig im Laufe der Entwicklung und stellt nach Vertragsverhandlungen die Schnittstelle zum Lieferanten dar. Was sind Erfolgsfaktoren bei der Zusammenarbeit zwischen PO (Kundenunternehmen) und restlichem agilen Team (Lieferant)? Welche notwendigen Qualifikationen ergeben sich hieraus für den PO? Und sind diese in einer Person/Rolle zu vereinen? Die Position des POs wird als essenziell betrachtet, da sie in der Literatur als einzige Schnittstelle über Unternehmensgrenzen hinweg beschrieben wird.

Die langfristige Wartung in Verbindung mit einem hohen Anteil an SW-Entwicklung macht die Wichtigkeit eines geregelten Prozesses klar. Verantwortlichkeiten für aufkommende Fehler müssen geklärt werden. Sowohl für die interne als auch die externe SW-Entwicklung. Die aktuell fehlende SW-Kompetenz bzw. -Kapazitäten müssen kurzfristig durch Überlegungen geeigneter Prozesse kompensiert werden, welche Lieferanten bei der Erstellung von Updates einbindet. Im Anbetracht der geschilderten Langfristigkeit und der Bedeutung für die Sicherheit im Straßenverkehr muss für diese Aufgabe auch interne Lösungen generiert werden.

## 6 Strategien für die Frühphase der zukünftigen Zusammenarbeit digitale Dienste

Aufgrund der zuvor beschriebenen Zusammenhänge lässt sich eine steigende Tendenz zur Kooperation zwischen Start-ups und OEMs im Bereich der Connectivity beobachten.

### 6.1 Zielsetzung

Das Ziel dieses Kapitels ist es, einen Leitfaden für die Gestaltung von Kooperationen zwischen Start-ups und OEMs im Bereich Connectivity zu entwickeln. Ein Schwerpunkt liegt auf der Untersuchung von Qualitätsaspekten in der Frühphase des Lifecycles digitaler Dienste. Mit Blick auf die zu untersuchenden Qualitätsaspekte ist eine spezifische Betrachtung des situativen Kontexts des gesamten Ökosystems notwendig. Das Hauptziel der Arbeit ist, neue Erkenntnisse in diesem noch wenig erforschten Feld zu generieren. Dazu wurde die folgende, der Arbeit zugrundeliegende Frage formuliert: Welche grundsätzlichen qualitätsspezifischen Aspekte sind für erfolgsversprechende Kooperationen zwischen OEMs und technologieorientierten Start-ups im Feld Connectivity von zentraler Bedeutung? Die Operationalisierung der übergeordneten Fragestellung erfolgte durch die Formulierung nachfolgend gelisteter Unterforschungsfragen, welche konzeptionell und empirisch beantwortet werden sollen:

1) *Wie arbeiten Start-ups im Bereich Connectivity?*

Zunächst ist zu klären, welche Art von Start-ups im Bereich der Connected Cars potenzielle Kooperationspartner für OEMs sind. Weiterhin soll die Arbeitsweise der identifizierten Start-ups durch eine literaturgestützte Untersuchung dargestellt werden.

2) *Wie ist das Qualitätsmanagement im Kooperationskontext verankert? Welche Rolle spielt Qualitätsmanagement für diese Start-ups?*

Für die Beantwortung der Unterfrage erfolgt eine literaturgestützte Analyse kooperationsrelevanter Qualitätsaspekte. Die untersuchten Kooperationen sind im spezifischen Kontext neuer Geschäftsfelder um das Connected Car angesiedelt. Somit sind Erkenntnisse bezüglich der Rolle des Qualitätsmanagements in diesem spezifischen Bereich notwendig. Zur weiteren Spezifizierung der zentralen Forschungsfrage in der Dimension Qualität wurde die Unterfrage speziell in Bezug auf Start-ups formuliert.

3) *Welches sind die wesentlichen Einflussfaktoren auf Kooperationen?*

Mit dieser Unterfrage soll erfasst werden, welche Faktoren die untersuchten Kooperationsvorhaben maßgeblich beeinflussen. Dabei stehen Erfolgsfaktoren und Hindernisse von Kooperationen im Vordergrund. Die Erkenntnisse bezüglich der Frage sollen empirisch durch Interviews gewonnen werden.

## 6.2 Vorstellung bisheriger Studien sowie Umfragen

Im Rahmen einer Literaturanalyse wurden für die durchzuführende Untersuchung relevante Erkenntnisse zusammengetragen. Die in Tabelle 6.1 zusammengefassten Ergebnisse zeigen, dass formale Prozess- und Qualitätsaspekte keine hohe Priorität haben. Hierfür lassen sich mehrere Ursachen benennen. Bereits zuvor wurde die Ressourcenknappheit erwähnt, unter welcher Start-ups agieren. Start-ups nutzen ihre knappen Ressourcen, um ihre Entwicklungsgeschwindigkeit zu maximieren, während Prozess- und Qualitätsmanagement zweitrangig sind. Ziel ist es, möglichst schnell ein Produkt mit minimalen Anforderungen und Eigenschaften zu realisieren. Ein solches Produkt wird auch als „Minimum Viable Product“ (MVP), bezeichnet und beschreibt die erste minimale funktionsfähige Iteration eines Produktes.<sup>[57]</sup>

Ergebnis der Erhebung	Art der Erhebung	Quelle
Eine Mehrheit der Start-ups im Bereich Software adaptiert keine Best-Practice-Methoden, vor allem weil diese als Standards für größere Unternehmen wahrgenommen werden.	Qualitative Studie (Interviews)	Sánchez-Gordón/Rory 2016 <sup>[129]</sup>
Start-ups haben nicht die notwendige Zeit, um entsprechende Engineering-Standards zu implementieren.	Qualitative Studie (Interviews)	
Grundsätzlich könnte die Software-Dokumentation optimiert werden, um das gemeinsame Verständnis für das Produkt und die Lösungen zu verbessern.	Qualitative Studie (Interviews)	Nascimento, L./Travassos, G., 2017 <sup>[130]</sup>
Start-ups aus dem Bereich Software arbeiten agil und haben keine festen Rollenbeschreibungen während der Entwicklung.	Qualitative Studie (Interviews)	Hof, J., 2019 <sup>[131]</sup>
Für Start-ups hat die Entwicklungsgeschwindigkeit Priorität, Qualitätsaktivitäten wie bspw. Testen werden auf ein Minimum reduziert.	Qualitative Studie (Interviews, Beobachtungen)	Gralha, C. et al., 2018 <sup>[132]</sup>
Die meisten Start-ups greifen auf einfache Dokumentationsmöglichkeiten für Anforderungen zurück.	Qualitative Studie (Interviews)	Melegati, J./Goldman, A., 2016 <sup>[133]</sup>
Start-ups folgen Vorgehensmodellen nicht strikt. Meistens werden Bestandteile aus bspw. Scrum für die eigene Entwicklung angepasst (Sprint).	Qualitative Studie (Interviews)	Giardino, C. et al., 2014 <sup>[134]</sup>
Mitarbeiter in den Entwicklungsteams von Start-ups tragen eine hohe Eigenverantwortung und nehmen mehrere Rollen im Entwicklungsprozess ein.	Qualitative Studie (Interviews)	
Qualitätsaspekte spielen im Vergleich zur Entwicklungsgeschwindigkeit eine deutlich untergeordnete Rolle.	Qualitative Studie (Interviews)	
Scrum ist die meist genutzte Entwicklungsmethode im agilen Umfeld.	Quantitative Umfrage	VersionOne, 2017 <sup>[135]</sup>

Scrum ist unter den agilen Methoden die beliebteste.	Quantitative Umfrage	Komus, A./ Kuberg, M., 2015 <sup>[136]</sup>
--	----------------------	--

Tabelle 6.1: Qualitätsmanagement in Start-ups (Quelle: Eigene Darstellung, Forschungsstudie QSK der TU Berlin.)

Entwicklungen in Start-ups sind im hohen Maße von den Ansichten und Fähigkeiten des Entwicklungsteams abhängig. Das Vorgehen in Start-ups ist nicht durch formale Prozesse geprägt, vielmehr nutzen sie agile Vorgehensmodelle wie insbesondere Scrum. Dabei werden diese nicht strikt befolgt, sondern im Sinne des Entwicklungsteams adaptiert. Diese Praktik spiegelt die hohe Eigenverantwortung wider, unter welcher agile Entwicklungsteams arbeiten. Somit sind qualitätssichernde Maßnahmen wie Dokumentation, Anforderungs- und Änderungsmanagement und Testen in hohem Maße von den Ansichten und Prioritäten der Entwickler abhängig. Aufgrund der beschriebenen Zeitknappheit lässt sich vermuten, dass entsprechende Maßnahmen nicht systematisch geplant und durchgeführt werden.

### 6.2.1 Beschreibung des Datenerhebungsinstruments und des Erhebungsablaufs

Zur Beantwortung der im Kapitel 6.1 gestellten Fragen wurde ein aus 18 Items bestehender Fragebogen konzipiert (s. Anhang 13). Um relevante Erkenntnisse zu den Start-ups zu gewinnen, wurden insgesamt vier Themenbereiche definiert. Im ersten Abschnitt wurden zunächst allgemeine Fragen zum Unternehmen gestellt. Dies soll es ermöglichen, die Stichprobe zu beschreiben, die der empirischen Untersuchung zugrunde liegt. Das darauffolgende Themenfeld bezog sich auf den Ist-Status des QMs der befragten Unternehmen. Dabei sollte bspw. erfasst werden, ob diese über QM-Zertifizierungen oder QM-Stellen verfügen. Anschließend wurden Fragen bezüglich der genutzten Vorgehensmodelle behandelt. Hierbei standen den Probanden die relevantesten iterativen und sequenziellen Modelle zur Auswahl. Weiterhin konnten nicht aufgeführte Vorgehensmodelle manuell hinzugefügt und Kombinationen mehrerer Vorgehensmodelle als Antwortmöglichkeiten ausgewählt werden. Der vierte Themenblock umfasste Fragen zur konkreten Umsetzung der Entwicklungsprozesse. Dabei sollten die Probanden die Systematik der Entwicklungsprozesse in ihren Unternehmen auf einer fünfstufigen Likert-Skala bewerten. Als Hilfestellung wurden unter den Fragen konkrete Beispiele für systematische Prozesse genannt.

Ein generelles Problem des Datenerhebungsprozesses mittels schriftlicher Befragung besteht darin, einen möglichst hohen Rücklauf zu erzielen. Mit der Intention, die Teilnahme an der Evaluation zu steigern, wurde der Fragebogen inhaltlich kompakt gestaltet. Die Fragen wurden möglichst einfach und leicht verständlich gehalten. Einleitende Worte informierten die Probanden über das Ziel der Untersuchung. Es wurde zudem auf die kurze Bearbeitungsdauer von etwa 10 Minuten und die Wahrung der Anonymität hingewiesen. An dieser Stelle ist zudem anzumerken, dass die exakte Formulierung von Fragen eine hohe Bedeutung für die Validität und Reliabilität einer

Erhebung hat. Aus diesem Grund wird empfohlen, Pretests durchzuführen.<sup>[137]</sup> Nach dem ersten Entwurf des Fragebogens in englischer und deutscher Sprache wurde daher das einheitliche Verständnis der Fragen und die Dauer der Befragung mittels Pretest getestet. Dafür stellten sich drei Probanden aus der Softwarebranche zur Verfügung. Im Anschluss erfolgte auf Grundlage der Pretests die Erstellung der finalen Versionen des Fragebogens.

Die Zielgruppe der empirischen Befragung bildeten technologieorientierte Start-ups aus dem Bereich connected Cars. Die relevante Grundgesamtheit eindeutig zu bestimmen, ist vor dem Hintergrund vieler Neugründungen und Start-ups, welche den Betrieb wieder einstellen, nicht möglich. Um dennoch eine geeignete, repräsentative Stichprobe zu ermitteln, wurden eine umfassende Recherche durchgeführt und verschiedene Internetquellen mit relevanten Informationen gesichtet. Im Fokus stand die Identifizierung von Unternehmen, welche digitale Dienste rund um das connected Car anbieten könnten. Das trifft insbesondere auf spezialisierte Technologie- und Softwareunternehmen zu. Die Daten zur Identifizierung relevanter Start-ups wurden größtenteils über Crunchbase bezogen. Dabei handelt es sich um eine 2007 gegründete Plattform, welche Daten zu privaten und öffentlichen Unternehmen bereitstellt. Die Validität der dort gespeicherten Daten wurde anhand anderer Studien geprüft, wobei eine Prüfung der Datenqualität ergab, dass die Unternehmensdaten im Einklang mit entsprechenden Studien standen.<sup>[137]</sup> Für die Segmentierung standen in Crunchbase verschiedene Filter zur Verfügung. So konnte sichergestellt werden, dass die Unternehmen jünger als zehn Jahre waren und thematische Schnittpunkte im Feld der connected Cars aufweisen. Insgesamt wurden so 366 Start-ups identifiziert.

Der Fragebogen war im Zeitraum November 2018 bis Februar 2019 online verfügbar und wurde in Form eines Links zur Umfrage zum einen via E-Mail oder über Formulare auf der Website an die 366 Start-ups versandt und zum anderen über universitäre und private Kontakte an relevante Start-ups herangetragen. Insgesamt wurden 26 Fragebögen in deutscher und sieben Fragebögen in englischer Sprache ausgewertet. Die Ergebnisse wurden mithilfe von Excel zusammengefasst und ausgewertet. Die Auswertung erfolgte auf deskriptiver Ebene. Auf tiefergehende statistische Analysen wurde aufgrund der geringen Rücklaufquote und methodenbedingter Verzerrungseffekte verzichtet.

## **6.2.2 Datenerhebung und Auswertung der Ergebnisse**

Im Folgenden werden die wesentlichen Ergebnisse der Umfrage dargestellt. Dabei werden die Themen bisherige Kooperationen mit etablierten Unternehmen, Qualitätsmanagement in Start-ups, Qualitätsmanagement-Rollen, Referenzmodelle, Vorgehensmodelle und Entwicklungsprozesse in Start-ups unterschieden. An das Kapitel schließt die Ergebnisdiskussion an. (Forschungsstudie am Fachgebiet QSK der TU Berlin.)<sup>[20]</sup>

### **1. Bisherige Kooperationen mit etablierten Unternehmen**

Die Frage bezüglich bisheriger Kooperationserfahrungen wurde nicht weiter präzisiert. Somit werden alle Unternehmen erfasst, welche bereits erste Berührungspunkte mit etablierten Unternehmen haben. Abbildung 6.1 zeigt, dass bereits 85 Prozent der befragten Start-ups in Kontakt mit etablierten Unternehmen standen.



Abbildung 6.1: Bisheriger Kooperationsstand von Start-ups

## 2. Qualitätsmanagement in Start-ups

Die Ergebnisse zu QM-Zertifizierungen in Start-ups sind in Abbildung 6.2 zu sehen. Etwa 77 Prozent der Start-ups gaben an, dass sie noch keine Zertifizierungen aufweisen. Allerdings gaben rund 19 Prozent an, dass bereits eine Einführung geplant ist. In insgesamt 4 Prozent der Unternehmen sind laut der Befunde QM-Zertifizierungen vorhanden. Es lässt sich feststellen, dass 76 Prozent der Probanden, die in einem über fünf Jahre alten Start-ups arbeiten, angaben, dass ihr Unternehmen QM-Zertifizierungen aufweist oder sich diese aktuell in der Planungsphase befinden.



Abbildung 6.2: QM-Zertifizierung in Start-ups

Weiterhin wurde gefragt, ob bereits feste Stellen für QM-Themen vorhanden oder in Planung sind. Die Ergebnisse in Abbildung 6.3 zeigen, dass dies bei rund 40 Prozent der befragten Start-ups der Fall ist. Insgesamt gaben vier Prozent der Probanden an, dass die Unternehmen über feste QM-Stellen verfügen.

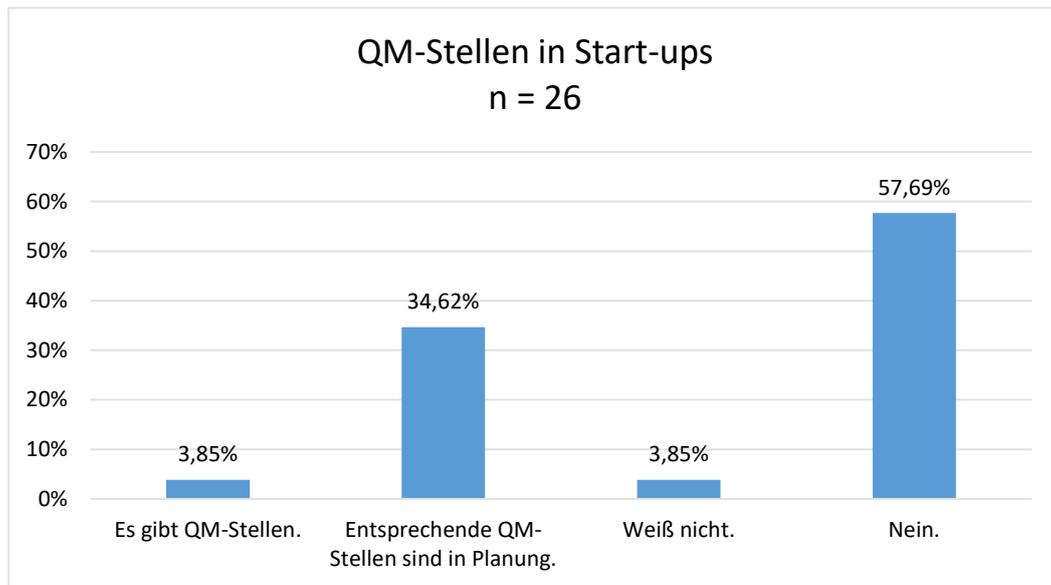


Abbildung 6.3: QM-Stellen in Start-ups (Eigene Darstellung)

### 3. Qualitätsmanagement-Rollen

Zuvor wurde gezeigt, dass innerhalb der befragten Start-ups erst wenige feste QM-Stellen existieren. Im Gegensatz zur vorherigen Verteilung zeigt sich, dass dieses Thema innerhalb der Entwicklungsteams stärker wahrgenommen wird. So gaben, wie in Abbildung 6.4 zu sehen, insgesamt knapp zwei Drittel an, dass entsprechende Rollen verteilt sind oder die Schaffung dieser Rolle in Planung ist. Ein Teilnehmer fügte beschreibend hinzu, dass die Entwicklungsarbeit im Rahmen von Retrospektivmeetings reflektiert wird.

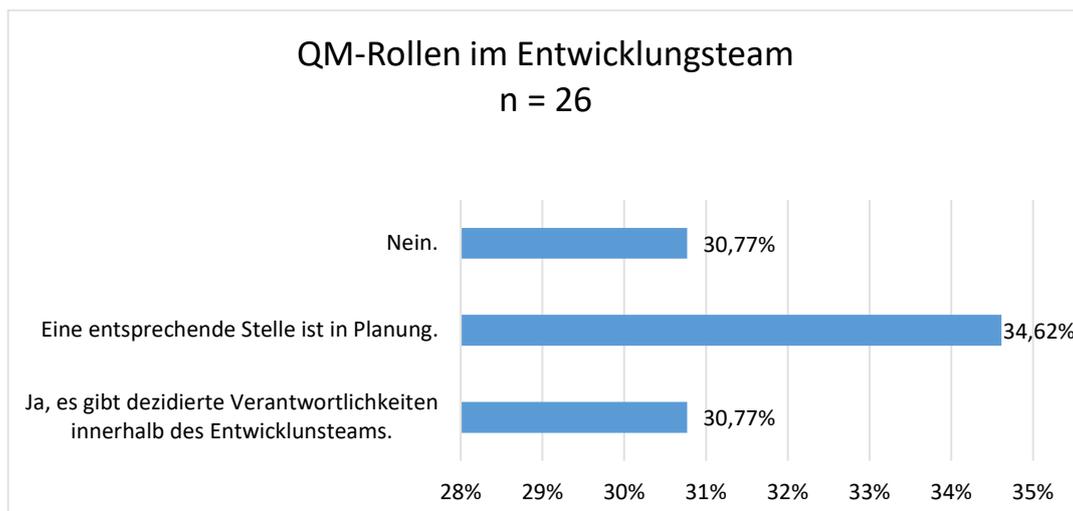


Abbildung 6.4: QM-Rollen in Entwicklungsteams (Eigene Darstellung)

### 4. Referenzmodelle

Die abschließende Frage thematisierte die Anwendung von Referenzmodellen in der Softwareentwicklung. Es ist zu erwähnen, dass keine konkreten Reifegrade abgefragt wurden. Es sollte in Erfahrung gebracht werden, wie viele Start-ups bereits Berührungspunkte mit Referenzmodellen aufweisen. Das Ergebnis ist in Abbildung 6.5 zu sehen und zeigt, dass über 40 Prozent der teilnehmenden Start-ups auf gängige Referenzmodelle (Square, Automotive Spice, CMMI) zurückgreifen. Dabei ist Automotive Spice mit etwa 20 Prozent unter den Probanden am verbreitetsten.

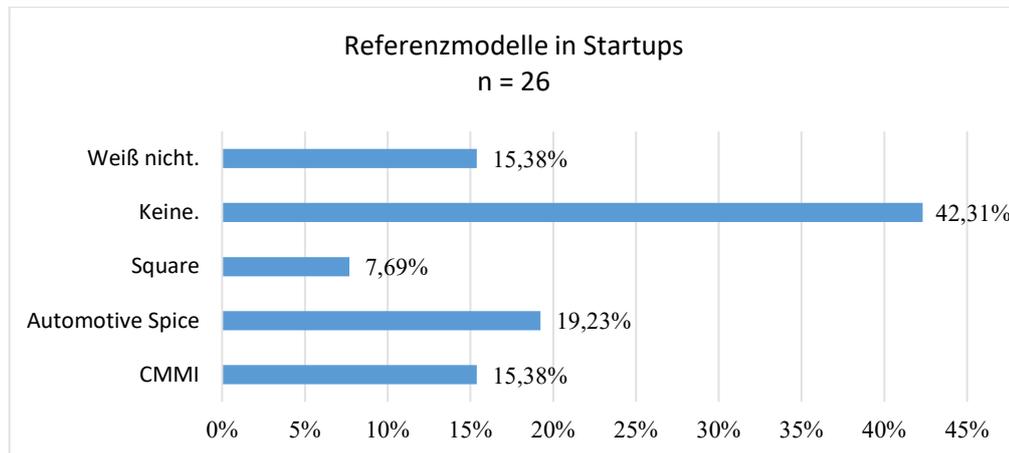


Abbildung 6.5: Referenzmodelle in Start-ups (Eigene Darstellung)

#### 5. Vorgehensmodelle in Start-ups

Da Unternehmen in der Praxis oftmals mehrere Vorgehensmodelle nutzen, war es den Probanden bei der Frage nach dem verwendeten Vorgehensmodell möglich, mehrere Antworten auszuwählen. Einerseits konnten Aktivitäten aus unterschiedlichen Modellen zu einem sogenannten Hybridmodell kombiniert, andererseits Modelle projektspezifisch ausgewählt werden. Es wurde zuvor bereits auf die hohe Relevanz von Scrum hingewiesen. Abbildung 6.6 verdeutlicht, dass Scrum mit über 47 Prozent das häufigste Vorgehensmodell bei den Teilnehmern ist. Insgesamt zeigt sich, dass iterative und wenig formalisierte Vorgehensmodelle deutlich stärker vertreten sind als sequenzielle.

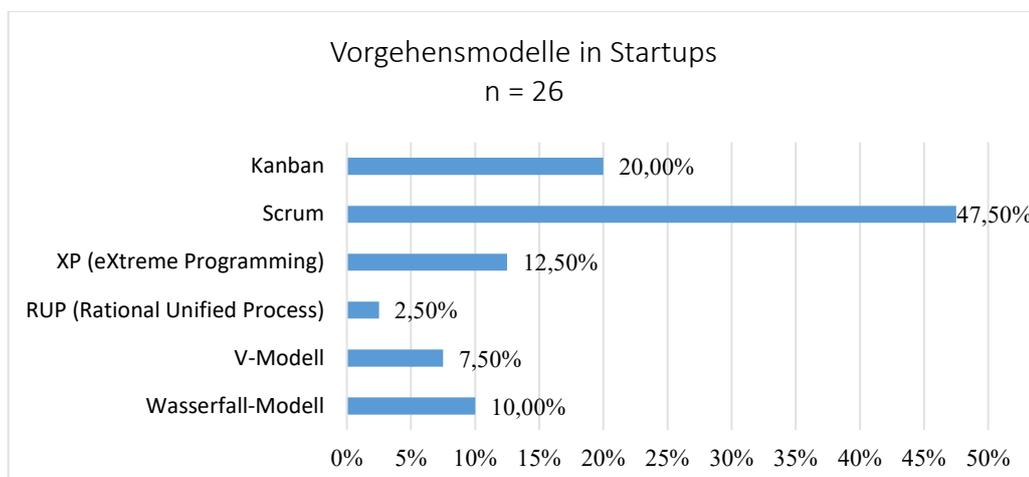


Abbildung 6.6: Vorgehensmodelle in Start-ups (Eigene Darstellung)

## 6. Entwicklungsprozesse in Start-ups

Die Fragen zum Themenbereich Entwicklungsprozesse in Start-ups dienen dazu, die Prozesse zu identifizieren und zu bewerten, welche die Start-ups in der Softwareentwicklung anwenden. Insgesamt wurden fünf Fragen zur Softwareentwicklung gestellt. Die Teilnehmer konnten jeweils auf einer diskreten Skala von eins (stimme überhaupt nicht zu) bis fünf (stimme voll und ganz zu) antworten. Zunächst wurde gefragt, ob die Entwicklung systematisch evaluiert wird, um Potenziale für Verbesserungen aufzudecken. Wie in Abbildung 6.7 zu sehen ist, ergab sich ein gemischtes Bild. So gaben zwar 70 Prozent der Start-ups an, bereits entsprechende Beurteilungen des eigenen Vorgehens vorzunehmen, doch dies geschieht nicht besonders regelmäßig.

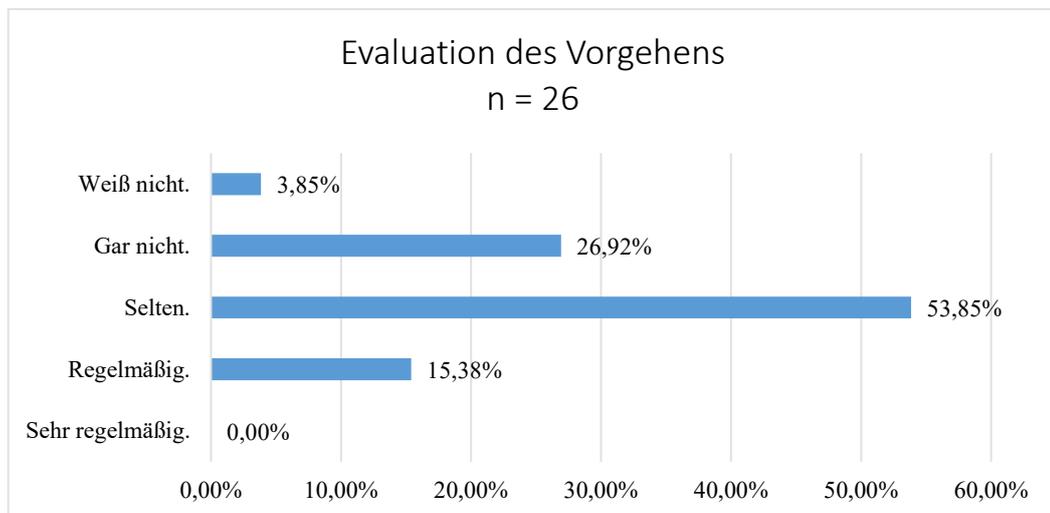


Abbildung 6.7: Evaluation des Vorgehens in Start-ups (Eigene Darstellung)

### 6.2.3 Diskussion der Ergebnisse

Die Erhebung sollte Erkenntnisse zu den Arbeitsweisen und zum Status des QMs in Start-ups liefern. Inferenzstatistische Analysen waren aufgrund der geringen Fallzahl nicht zweckdienlich, weshalb die Auswertung der in Kapitel 6.2 vorgestellten Erkenntnisse auf deskriptiver Ebene stattfindet. Im arithmetischen Mittel wiesen die befragten Start-ups ein Alter von 3,57 Jahren auf, damit waren sie um etwa zwei Jahre jünger als der Durchschnitt der zugrundeliegenden Stichprobe. Der hohe Anteil von Start-ups (85 %) mit Kooperationserfahrungen kann dadurch erklärt werden, dass vernetzte Dienste im Auto mit neuen Herausforderungen in den Bereichen Cloud-Computing, Big Data, Internet der Dinge, 186 und Data Security einhergehen. Vor diesem Hintergrund ergeben sich Potenziale für innovative Lösungskonzepte, welche insbesondere Chancen für neue Unternehmen bieten. Die Ergebnisse zur allgemeinen Charakterisierung der Teilnehmer lassen sich wie folgend zusammengefasst:

- Der Großteil der untersuchten Start-ups arbeitet im B2B-Bereich und verfügt bereits über Kooperationserfahrungen mit etablierten Unternehmen.
- Qualitätsmanagement hat zurzeit noch keine hohe Priorität in Start-ups.

- Agil arbeitende Teams definieren kaum feste Rollen für Qualitätsaspekte während der Entwicklungsarbeit.
- Start-ups setzen auf agile Methoden, insbesondere auf Scrum.
- Qualitätssichernde Aktivitäten sind im hohen Maß vom Entwicklungsteam abhängig.

Zusammenfassend kann festgehalten werden, dass die untersuchten Start-ups bereits erste Berührungspunkte zu kooperationsrelevanten Qualitätsthemen wie beispielsweise dem Branchenstandard Automotive Spice haben. In der konkreten Entwicklungsarbeit sind qualitätssichernde Aktivitäten jedoch vor allem von den handelnden Personen in den überwiegend agilen Entwicklungsteams abhängig.

### 6.3 Industrie-Interviews

Um ein tieferes Verständnis für die unternehmerischen Entscheidungen bezüglich Kooperationsvorhaben zu erlangen, wurden im Verlauf der Dissertation Industrie-Interviews durchgeführt.

#### Interviewfragen

Als Interviewform wurde das explorative, systematisierende Experteninterview gewählt. Diese Form dient nach Bogner, Littig und Menz dazu, individuelles Deutungswissen, Sachinformationen sowie praxisbasiertes Handlungs- und Erfahrungswissen zu erlangen, und eignet sich somit für die Beantwortung der Frage im Kapitel 6.1.<sup>[139]</sup> Es wurden sechs Einzelinterviews geführt. (s. Anhang 23) Die Themenfelder, die in den Interviewfragen angesprochen wurden, lassen sich wie folgt zusammenfassen:

- 1) Wie wird die Qualitätssicherung innerhalb der Kooperation gestaltet?
- 2) Wie werden Differenzen zwischen den Unternehmenskulturen überwunden?
- 3) Welches sind die kritischen Erfolgsfaktoren für die Kooperation mit Start-ups/etablierten Unternehmen?
- 4) Welche Faktoren erschweren die Kooperation mit Start-ups/etablierten Unternehmen?

<b>Funktion bzw. Position</b>	<b>Unternehmen</b>	<b>Form</b>
Mitarbeiter im Einkauf digitaler Services	OEM	mündlich
Teamleiter im Bereich IT-Innovationen	Einzelhandel	schriftlich
Mitarbeiter im Bereich Innovations- und Kooperationsmanagement	OEM	mündlich
Projektmanager im Bereich Kooperationsmanagement	OEM	schriftlich
Manager im Bereich IT-Innovationen	Beratungsunternehmen	mündlich
Teamleiter im Einkauf digitaler Services	OEM	mündlich

Tabelle 6.2: Übersicht der Interviewpartner (Quelle: Eigene Darstellung)

## 6.4 Ergebnisse und Ergebnisdiskussion

Mit Blick auf die Interviews Fragen (Anhang 13) werden nachfolgend die zuvor dargestellten Themenfelder nacheinander diskutiert. In Bezug auf die Frage, welche Rolle das Qualitätsmanagement in Kooperationen spielt, kann dabei zunächst festgestellt werden, dass der Begriff des QMs innerhalb der untersuchten Kooperationen kontext- und situativ geprägt ist. Viele Experten sprachen von einem kooperationspezifischen Vorgehen. Allerdings lässt sich konstatieren, dass die Implementierung qualitätssichernder Maßnahmen ansteigt, insbesondere bei längerfristigen Kooperationen und in der Entwicklung von sicherheitskritischen Produkten. Bei diesen Kooperationen fordern die OEMs die Einhaltung von Vorgaben und Richtlinien. Als Beispiele können Dokumentationspflichten oder festgeschriebene Qualitätsaudits genannt werden. Teilweise wurde auf Seiten von Start-ups ein mangelndes Verständnis für den Themenbereich IT-Sicherheit konstatiert. Um die Kooperation im Hinblick auf die Qualität erfolgreich zu gestalten, sei eine enge Zusammenarbeit notwendig. Diese kann beispielsweise durch die Bildung von gemeinsamen Teams realisiert werden.

Somit kann festgestellt werden, dass kooperationsrelevante Qualitätsthemen keiner strategischen Planung unterliegen. In diesem Zusammenhang lässt sich vermuten, dass die QM-Fachbereiche nicht ausreichend in agile Projekte eingebunden werden. Des Weiteren kann davon ausgegangen werden, dass bei den untersuchten Kooperationen ein Mangel an Know-how über qualitätsspezifische Themen herrscht. Somit könnte zum einen der Aufbau von spezifischem Know-how und zum anderen die agile Ausrichtung des QMs von OEMs ungenutztes Potenzial bieten, um Kooperationen mit Start-ups besser zu unterstützen. In diesem Zusammenhang ist auch eine grundsätzliche Entwicklung hin zu einer agileren Unternehmenskultur innerhalb von OEMs zu empfehlen.

In Bezug auf die Hindernisse, welche innerhalb von Kooperationen identifiziert werden können, lässt sich zusammenfassend sagen, dass viele Experten auf fehlendes Verständnis als Kooperationshindernis verwiesen. Konkret wurde beispielsweise ein Mangel an Wissen seitens Start-ups über die bürokratischen Hürden bei OEMs genannt. Darüber hinaus wurden fehlende Vertrauensverhältnisse als Ursache für ein Nichtzustandekommen von Kooperationen genannt. Mit Blick auf kulturelle Differenzen wurde vor allem auf Diskrepanzen innerhalb der gelebten Vorgehensweisen und Prozesse hingewiesen. Auch die Knappheit von Ressourcen seitens der Start-ups birgt Konfliktpotenzial, welches insbesondere die Vertragsgestaltung erschwert. Diesbezüglich ergibt sich ein Spannungsfeld zwischen knappen Ressourcen bei den Start-ups und dem Ziel der OEMs, Kosten einzusparen. Die meisten Start-ups haben keinen internen Zugriff auf Juristen, weshalb die Prüfung von Verträgen als Hindernis genannt wurde. Aus interner Sicht eines OEMs wurden vor allem zu lange Entscheidungswege, in deren Folge es zu Zeitverzögerungen kommt, als Hindernisse genannt.

Es ist anzunehmen, dass einige der beschriebenen Hindernisse durch eine offene und transparente Kommunikation überwunden werden können. Hierfür ist es wichtig, ein gemeinsames Kooperationsverständnis mit eindeutigen Zielen zu formulieren. Die Kooperationspartner sollten vorab Verständnis für die Situation des anderen entwickeln, um so mögliche Missverständnisse zu vermeiden. OEMs sollten ihre internen Strukturen anpassen, um größere Zeitverzögerungen zu vermeiden. Als konkretes Beispiel wurde der aufwendige Einkaufsprozess genannt. Des Weiteren könnte ein Überwinden von langen Entscheidungswegen und unflexiblen Richtlinien eine agil ausgerichtete Kooperation positiv beeinflussen.

Welches sind die Erfolgsfaktoren von Kooperationen? Die Ergebnisse der Interviews zeigen deutlich, dass eine offene und faire Kommunikation zwischen den Kooperationspartnern als wichtiger Erfolgsfaktor angesehen wird. Hierfür ist ein jederzeit faires und realistisches Feedback wichtig. Weiterhin wurden gute Erfahrungen mit dem Einsatz von Vermittlern gemacht, welche über beidseitiges Know-how verfügen. Start-ups könnten zum Beispiel einen Branchenexperten einstellen, um so bereits in der Anbahnungsphase mögliche Hindernisse zu beseitigen. Ein weiterer Punkt ist der Aufbau eines gemeinsamen Verständnisses. Ziele und Vorstellungen im Hinblick auf die Kooperation sollten jederzeit fair und transparent vereinbart werden. Dabei sollten die Partner Rücksicht auf die wirtschaftlichen und bürokratischen Rahmenbedingungen des Partners nehmen. Bei Start-ups wären das knappe finanzielle und personelle Ressourcen, OEMs wiederum unterliegen bürokratischen Hürden und internen Anforderungen, welche von Kooperationspartnern teilweise ein hohes Maß an Geduld erfordern. An dieser Stelle ist auch die von den Experten hervorgehobene Kompromissbereitschaft zu betonen. Die verschiedenen Kulturen lassen sich zusammenführen, wenn eine transparente und faire Kompromissbereitschaft herrscht. Weiterhin wurde aus Sicht von OEMs mehrfach die Wichtigkeit eines internen Sponsors erwähnt. Eine starke Kooperationsunterstützung aus der Führungsebene ermöglicht einen stärkeren Zugriff auf finanzielle und personelle Ressourcen. Vor diesem Hintergrund sollten die Kooperationsziele mit grundsätzlichen Unternehmenszielen vereinbar sein.

Zusammenfassend lassen sich verschiedene Handlungsfelder für eine erfolgreiche Gestaltung von Kooperationen erkennen. Es ist essenziell, gemeinsame Ziele zu formulieren, damit beide Partner deutlich erkennbar von der Kooperation profitieren. Weiterhin sollten Schlüsselpositionen mit kompetenten Mitarbeitern besetzt werden, welche über beidseitiges Know-how verfügen. Aufgrund des hohen Stellenwertes von Kommunikation zwischen den Partnern muss ein transparenter, regelmäßiger und professioneller Austausch möglich sein. An dieser Stelle wurden gute Erfahrungen mit physischen Austauschformaten gemacht. Ein regelmäßiger Austausch ist auch eine Maßnahme, um kulturelle Differenzen zu überwinden. Intern ist zu beachten, dass Kooperationen unter Berücksichtigung strategischer Ziele eines Unternehmens initiiert werden sollten. Des Weiteren ist die Gewinnung wichtiger interner Unterstützer aus dem höheren Management als Erfolgsfaktor zu betrachten.

## **Erfolgsfaktoren und Handlungsempfehlungen**

- Schaffung eines gemeinsamen Kooperationsverständnisses
- Gestaltung einer offenen und vertrauensvollen Kommunikation
- Mitbringen von grundsätzlicher Kompromissbereitschaft
- Gewinnung interner Unterstützer für Kooperationsvorhaben
- Förderung einer agilen Kultur im Unternehmen
- Entwicklung eines Qualitätsbewusstseins durch Start-ups

Die wissenschaftliche Literatur diskutiert bereits Ansätze für eine agile Gestaltung des QMs. Es hat sich gezeigt, dass sich Branchenstandards wie etwa Automotive Spice mit agilen Entwicklungsansätzen vereinbaren lassen. Weiterhin konnte auf Grundlage der Interviews festgestellt werden, dass Qualitätsthemen in Kooperationen im hohen Maße von kooperationspezifischen Kriterien abhängen. Eine feste Einbindung des QMs von OEMs konnte nur für die Entwicklung sicherheitskritischer Software bestätigt werden. Es lässt sich vermuten, dass ein gezieltes Vorgehen bezüglich der Qualität bei der kooperativen Entwicklung unkritischer Softwarekomponenten noch nicht in geeignetem Maße stattfindet. Somit entscheiden über Qualitätsthemen in Kooperationen oftmals einzelne Mitarbeiter, welche die vorhandenen Ressourcen einplanen müssen. In diesem Kontext ist eine Erweiterung des QMs von OEMs um agile Ansätze zu empfehlen. Auf diese Weise kann sichergestellt werden, dass entsprechende Qualitätsmaßnahmen sich nicht negativ auf die Entwicklungsgeschwindigkeit auswirken.

## **6.5 Kritische Einordnung, Ausblick und Zusammenfassung**

Die betrachtete Thematik unterliegt einer hohen Dynamik. Die im Rahmen dieses Kapitels beschriebenen Marktentwicklungen und Strategien sind unter dieser Einschränkung als aktuelle Bestandsaufnahme einzuordnen. Weiterhin ist anzumerken, dass aufgrund des relativ unerforschten Gegenstands ein explorativer Forschungsansatz gewählt wurde. Kritisch ist in diesem Zusammenhang zum einen die geringe Rücklaufquote der quantitativen Erhebung zu erwähnen, auf deren Grundlage repräsentative Aussagen bezüglich der Grundgesamtheit nicht möglich sind. Zum anderen spiegelt die qualitative Erhebung vor allem die Sichtweise von OEMs auf Kooperationen mit Start-ups wider. Die qualitativ gewonnenen Erkenntnisse sind somit lediglich als erste Annäherung an das Untersuchungsfeld einzuordnen und sollten nicht als allgemeingültig betrachtet werden. Trotz der beschriebenen Limitationen bieten die Ergebnisse Anhaltspunkte, welche für die grundsätzliche Gestaltung von Kooperationen relevant sind. Es empfiehlt sich aber einerseits weitere quantitative Studien mit einer größeren Stichprobe durchzuführen und andererseits die qualitativen Studien auszuweiten, wobei insbesondere relevante Start-ups einzubeziehen sind.

Themenspezifisch erscheint eine nähere Betrachtung der digitalen Transformation von OEMs besonders interessant. Es stellt sich die Frage, wie diese die Wertschöpfung im Bereich Connectivity gestalten wollen. In Bezug auf Kooperationen ist anzunehmen, dass dies maßgeblichen Einfluss auf die Koordinationsform der untersuchten

Kooperationen zwischen Markt und Hierarchie haben wird. Hieraus ergeben sich Implikationen für deren konkrete Ausgestaltung. Somit könnte eine umfassende Einordnung relevanter Kooperationen im Kontinuum von Markt und Hierarchie weitere forschungsrelevante Erkenntnisse liefern.

Das Ziel dieses Kapitels bestand in der Entwicklung eines Leitfadens für die Gestaltung von Kooperationen zwischen OEMs und technologieorientierten Start-ups im Bereich Connectivity. Für die Beantwortung der Forschungsfrage wurden neben einer theoretischen Fundierung des Untersuchungsgegenstands empirische Erhebungen vorgenommen. Abschließend konnten ausgehend von den theoretischen und empirischen Erkenntnissen Handlungsempfehlungen für die Gestaltung von Kooperationen im Bereich der digitalen Dienste vernetzter Fahrzeuge abgeleitet werden. Erfolgreiche Kooperationen basieren vor allem auf der Festlegung von gemeinsamen strategischen Zielen. Diese sollten auf Basis einer partnerschaftlichen Entwicklungsarbeit, welche stets auf Augenhöhe stattfindet, realisiert werden. Dabei können kulturelle Differenzen durch ein hohes Maß an Kompromissbereitschaft, Transparenz und Offenheit überwunden werden. Weiterhin ist in diesem Zusammenhang ein regelmäßiger Austausch zwischen den Kooperationsakteuren wichtig.

Die Automobilhersteller stehen vor der Herausforderung, auf eine durch die voranschreitende digitale Transformation zunehmende Marktdynamik angemessen zu reagieren. Vor diesem Hintergrund ist einerseits die strategische und zukunftsgerichtete Selektion passgenauer Kooperationspartner zu nennen, andererseits spielt die Entwicklung agiler Strukturen und dynamischer Fähigkeiten im eigenen Unternehmen eine wichtige Rolle. In diesem Kontext ist zudem die Entwicklung einer agilen Qualitätskultur zu empfehlen. Die beschriebenen Fähigkeiten sind maßgeblich, um die zukünftige Wettbewerbsfähigkeit zu gewährleisten. Es geht darum, Entwicklungszeiten zu verkürzen und flexibel auf neue Kundenanforderungen reagieren zu können. Diese Fähigkeiten gelten als wichtige Erfolgsfaktoren im Geschäftsfeld digitaler Dienste. Somit können Automobilhersteller durch eine zukunftsgerichtete Auswahl und Gestaltung von Kooperationsvorhaben sowie angemessene interne Rahmenbedingungen die Transformation vom Produktanbieter zum Dienstleistungsunternehmen erfolgreich gestalten.

## 7 Agiler Fehlermanagementprozess für digitale Dienste

Die Quellenlage im Bereich agile Softwareentwicklung allgemein sowie in den Bereichen des Qualitätsmanagements in agilen Projekten, agile Reifegradmodelle und weitere, an das Fehlermanagement angrenzende Bereiche ist sehr umfangreich. Speziell im Bereich des agilen Fehlermanagements von Apps und digitalen Produkten für die Automobilindustrie ist die Quellenlage jedoch nicht ausreichend. Das Fehlermanagement von digitalen Produkten in agilen Umgebungen ist ein aktuelles Thema, mit dem sich die Automobilindustrie auseinandersetzt und mit besonderen Herausforderungen verbunden ist.

### 7.1 Ergebnisse der Interviews Leitfaden Fehlermanagement

Um einen agilen Fehlermanagementprozess für digitale Produkte zu entwerfen und die aktuellen Strukturen, Entwicklungen, Prozesse und Probleme zu verstehen, wurden im Rahmen dieser Arbeit drei anonyme qualitative Interviews (s. Anhang 14) mit Industrievertretern durchgeführt. Es konnten zwei Experten aus Qualitätssicherungsabteilungen digitaler Produkte von zwei deutschen Automobilherstellern sowie ein Experte aus einem deutschen Telekommunikationsunternehmen befragt werden. Ziel der Experteninterviews war es, einen praktischen Bezug zur Problematik Fehlermanagement in agilen Umgebungen herzustellen. Dabei lag der Fokus darauf herauszufinden, welche Rolle agile Methoden und Softwaretests bei der Entwicklung von digitalen Produkten spielen, wie der Fehlermanagementprozess derzeit abläuft und vor welchen Problemen die Unternehmen bei der Behebung von Fehlern stehen. Außerdem sollte die Rolle von Lieferanten während der Entwicklung und des anschließenden Betriebes thematisiert werden. Da keine Verschwiegenheitserklärung unterschrieben wurde sowie die Befragung telefonisch erfolgte, muss an dieser Stelle darauf hingewiesen werden, dass die Interviews auf Wunsch der Interviewpartner weder aufgenommen wurden noch anonym veröffentlicht werden.

Im Anhang 14 befindet sich eine Liste von Beispielfragen, die in Vorbereitung auf das Interview erstellt und den Interviewpartnern vor den Interviews zugesandt wurden. Die Fragen sind als Leitfaden zu verstehen und wurden im Laufe des Gesprächs angepasst. Im Folgenden werden die Kernaussagen aus den Interviews dargelegt. Dabei ist natürlich zu beachten, dass Antworten von Experten aus zwei deutschen OEMs nur einen kleinen Ausschnitt darstellen und nicht im wissenschaftlichen Sinne verallgemeinerbar sind. Inhaltlich lassen sich die Ergebnisse wie folgt zusammenfassen:

- 1) Ein Großteil der Entwicklung von digitalen Anwendungen findet gemeinsam mit externen Lieferanten bzw. mit eng angebundenen Tochterunternehmen statt, die nahezu komplett in die Unternehmensstruktur integriert sind. Schwierigkeiten in Bezug auf Kommunikation und/oder Abstimmungsprobleme während der Entwicklung bzw. des Betriebes, wie sie in der Regel mit externen Partnern auftreten, spielen eher keine Rolle.

- 2) JIRA, das Fehlermanagementtool für agile Projekte, war allen Interviewpartnern vertraut. (JIRA ist eine Webanwendung zur Fehlerverwaltung, Problembehandlung und operativem Projektmanagement, die von Atlassian entwickelt wurde.)
- 3) Agile Entwicklungsmethoden werden vor allem im Bereich der Frontendentwicklung angewendet, wobei keine Trennung von Entwicklung und Betrieb mehr erfolgt.
- 4) Je komplexer die Anwendungen bzw. digitalen Produkte sind und je mehr Backendsysteme miteinander kommunizieren, desto schwieriger sind agile Methoden im Bereich der Entwicklung und Fehlerbehebung anzuwenden.
- 5) Softwaretests sind eine zentrale Säule der Qualitätssicherung in der Entwicklungsphase, finden jedoch zu selten auf Gesamtsystemebene, sondern oft nur recht isoliert und produktspezifisch in den Abteilungen statt.
- 6) Die zentrale und wichtigste Aussage, die allen Interviews zu entnehmen war, ist, dass die große Herausforderung darin besteht, eine Fehlerallokation und -zuordnung auf die Backendsysteme bzw. Schnittstellen zu erreichen, um eindeutige Verantwortlichkeiten bzw. Zuständigkeiten zuweisen zu können.

Die Ergebnisse der Interviews wurden beim Entwurf des Fehlermanagementprozesses berücksichtigt. Sie haben gezeigt, dass in den betroffenen deutschen Autofirmen ein beträchtlicher Teil der Softwareentwicklung digitaler Anwendungen in agilen Umfeldern und unter Einsatz moderner agiler Methoden erfolgt. Dabei verwenden die Teams unterschiedliche Test- und Dokumentationssoftware und arbeiten im Sinne agiler Vorgehensmodelle in kleinen, unabhängigen und eigenverantwortlichen Teams. Bei digitalen Anwendungen, die auf wenige Backendsysteme zurückgreifen und damit relativ unabhängig von der Gesamtsysteminfrastruktur sind, funktioniert dieser Ansatz, auch mit Blick auf einen effektiven Fehlermanagementprozess, sehr gut. Das trifft auf Apps zu, die keine oder kaum Auswirkungen auf die Hardware im Fahrzeug haben und als reine Smartphoneanwendung funktionieren. In der Regel sind solche digitalen Produkte dem Marketing zugeordnet. Die Teams, die für diese Art von Anwendungen verantwortlich sind, können ihre eigene Infrastruktur aufbauen, müssen nicht auf historisch gewachsene IT-Architekturen und -Verantwortlichkeiten Rücksicht nehmen und Schnittstellen zu anderen Systemen und Backend-Bereichen nicht oder nur wenig beachten. Tritt ein Fehler auf, kann die Ursache in der Regel schnell gefunden und behoben werden. Sobald Apps bzw. digitale Anwendungen jedoch mit dem Fahrzeug und mit Server- und/oder Cloudsystemen interagieren („Embedded Apps“), findet eine Vielzahl von Kommunikationsprozessen zwischen den Backendsystemen statt. Der Kunde sieht die Anwendung im Frontend, alle Prozesse, die im Hintergrund ablaufen, werden als Backend bezeichnet. Diese Backendsysteme werden von unterschiedlichen Teams betreut und sind oft historisch gewachsen (Stichwort: Legacy System). Es sind demnach bei Entwicklung und Betrieb von „Embedded Apps“ viele Schnittstellen, Umgebungen und komplexe Systeme zu beachten.

In allen Experteninterviews wurde bemängelt, dass auftretende Fehler aufgrund der komplexen IT-Gesamtinfrastruktur während des Betriebes nicht bzw. nur unter großem Aufwand einzelnen Systemen, Komponenten und damit auch Zuständigkeiten zugeordnet werden können. Ohne eine entsprechende Fehlerallokation bzw. -zuordnung

ist eine Fehlerbehebung jedoch nicht möglich. Der Schlüssel für eine schnelle Fehlerzuordnung während des Betriebes liegt in der Entwicklungsphase, speziell im Bereich der automatisierten Softwaretests. Bei agilen Entwicklungsmethoden ist die Abstimmung und Gesamtorchestrierung der Softwaresysteme, Teams und Organisationsbereiche besonders wichtig. Dabei muss eine gute Kommunikation, crossfunktionale Zusammenarbeit und eine abgestimmte Herangehensweise gewährleistet sein, um schnittstellen- und bereichsübergreifende Prozesse und Kommunikation zu ermöglichen. DevOps liefert hier einige hilfreiche Ansätze, wobei die Verschmelzung von Entwicklung und Betrieb eine wichtige Voraussetzung ist. Sind die Entwickler selbst für die Überwachung und den Betrieb ihrer Systeme verantwortlich, führt dies von vornherein dazu, dass während der Entwicklung bzw. dem Schreiben von Code Methoden und Tools in Codeform implementiert werden, die ein Tracken der Performance und eine schnelle Fehlerallokation ermöglichen.

Wie bereits erwähnt, ist neben einer guten Gesamtorganisation der einzelnen agilen Entwicklerteams der Bereich der Softwaretests von zentraler Bedeutung. Jede Softwareanwendung, das heißt auch digitale Dienste und Apps, müssen getestet werden. In den klassischen Vorgehensmodellen, wie Wasserfall- oder V-Modell, finden Tests meist nicht automatisiert und zu einem späten Zeitpunkt im Entwicklungsprozess statt. Agile Entwicklungsmethoden ermöglichen das Testen parallel zur Entwicklung. Dabei handelt es sich jedoch häufig um Tests auf Modul- bzw. Codeebene, die den Entwicklern ein schnelles Feedback geben. Im Gesamtsystem werden Anwendungen jedoch auch in agilen Projekten erst spät implementiert und getestet. Hier treten oftmals Fehler auf, die schwer zu beheben sind, da eine Vielzahl komplexer Abhängigkeiten zwischen den einzelnen Servern und Backendsystemen besteht. Daher muss eine Möglichkeit geschaffen werden, einzelne Anwendungen und Funktionen früher im Gesamtsystem zu testen und auftretende Fehler so zu dokumentieren, dass sie später eine Grundlage bilden, um im Betrieb schnell Fehlerursachen ermitteln zu können. Das Zusammenführen von Entwicklung und Betrieb, IaC und ein höchstmöglicher Grad an automatisiert ablaufenden Prozessen können die Testautomation so weiterentwickeln, dass kleine Entwicklerteams am Ende ihrer Sprints die Inkremente im Gesamtsystem testen können. Hier steht nicht die Fehleranalyse des Codes an sich im Mittelpunkt, diese findet täglich statt, sondern die Anbindung an andere Backends.

Wenn die Anwendungen in einem virtualisierten Gesamtsystem getestet werden, sollten Fehlerbilder, -ursachen und -behebungsmaßnahmen gesammelt und verknüpft werden können. Diese Verknüpfung, eventuell allein das Aufzeigen von Backendsystemen, die bei einem Prozess wie „Tür-öffnen-per-Smartphone“ beteiligt sind, kann einen Mehrwert darstellen und später die Fehlerbehebung während des Betriebes erleichtern. Die Bedeutung einer entsprechenden Dokumentation nimmt in solchen komplexen Umgebungen aber nicht ab. Um dabei nicht in alte nicht-agile Methodiken zu verfallen und einen Großteil der Arbeitszeit an die Dokumentation zu verlieren, sollte die Fehlerdokumentation weitgehend automatisiert erfolgen, indem Logfiles, Fehlerbilder und Fehlerhistorie eines Prozesses zusammengefügt werden.

## 7.2 Fehlerklassifizierung für digitale Dienste

Die während der Entwicklungsphase sowie der Nutzungsphase beim Endkunden erkannten Fehler enthalten relevante Informationen für die anschließenden Fehlerbeseitigungsmaßnahmen. Ein effektiver Fehlerbeseitigungsprozess dient zur Wiederherstellung der Produkt- und Prozessqualität. Ein zweckmäßiges Klassifizierungsmodell verarbeitet die Informationen der Fehlererfassung, um die Fehler zu priorisieren. Ein effektives Einleiten von Fehlerbehebungsmaßnahmen setzt eine hohe Dokumentationsqualität voraus. Eine toolunterstützte Fehlererfassung und ein datenbank- und webbasiertes System zum Defektmonitoring (s. Anhang 3: Watchlist) soll die geforderten Qualitätsansprüche gewährleisten. Eine möglichst detaillierte Fehlerbeschreibung mit unternehmensspezifischen Attributen, Hardcopies, Logfiles oder Datenkonstellationen ist bei der Fehlerursachenfindung hilfreich. Für eine hohe Dokumentenqualität sind die folgenden Informationen ausschlaggebend:<sup>[52]</sup>

- ein aussagekräftiger, eindeutiger Fehlertitel
- eine zugeordnete Fehlernummer oder Fehler-ID
- Datum der Entdeckung
- Programmbezeichnung, Versionsnummer oder Entwicklungsgrad
- detaillierte, genaue Fehlerbeschreibung
- Informationen oder Bedingungen zur Reproduzierbarkeit
- mögliche Seiteneffekte des Fehlers (Wirkung)
- Fehlerbearbeitungsstatus
- Historie der Fehlerbearbeitung
- Verantwortlicher Bearbeiter
- Hardwarekonfiguration
- Betriebssystem
- Angaben zur geplanten Fehlerbeseitigung: Zeit, Umsetzung, Maßnahme
- weitere Fehlerattribute

### Fehler-ID-Phasenmodell

Wesentliche Fehlerinformationen können in der Fehler-ID erfasst werden. Die Fehlercodes sind in einer Auswahlliste vorgegebenen (s. Abb. 7.1). Die Codes der Fehler-ID sind in einem separaten Fehlerkatalog ausgeschlüsselt (s. Anhang 15). Der standardisierte Fehlerkatalog kann im Rahmen der Qualitätssicherung inhaltlich durch zusätzliche Positionen erweitert werden, wobei dies ausschließlich durch befugte Personen einer Organisationseinheit erfolgt, die feste Regeln zu befolgen haben. Der Fehlerkatalog mit den entsprechenden Codes deckt alle bekannten Fehler ab und dient als Grundlage für die Auswertung von Kundenrezensionen.

#### 1) Download

Bevor die Nutzungsphase einer Anwendungssoftware mit einem funktionalen Mehrwert für den Kunden beginnt, muss dieser die Applikation herunterladen und auf dem Endgerät installieren. Die Schnittstelle für diesen Vorgang bildet ein Vertriebskanal (u. a.

Google Play Store, App Store), der in Abhängigkeit vom mobilen Betriebssystem des Smartphones ausgewählt wird. Im entsprechenden Vertriebskanal muss nach dem Titel der Applikation gesucht werden. Um Download sowie Installation zu starten, müssen bestimmte Voraussetzungen erfüllt sein. Dazu zählen unter anderem ein gültiges Nutzerkonto auf der Vertriebsplattform, eine vorhandene Datennetzverbindung, ausreichender Speicherplatz auf dem Endgerät, Mindestanforderungen an das mobile Betriebssystem oder die Zustimmung zu Nutzungsbedingungen und einer Datenschutzerklärung. In dieser Phase haben auftretende Fehler und Störungen einen vollständigen Nutzungsausfall zur Folge. Mit der abgeschlossenen Installation gilt diese erste Phase als beendet.

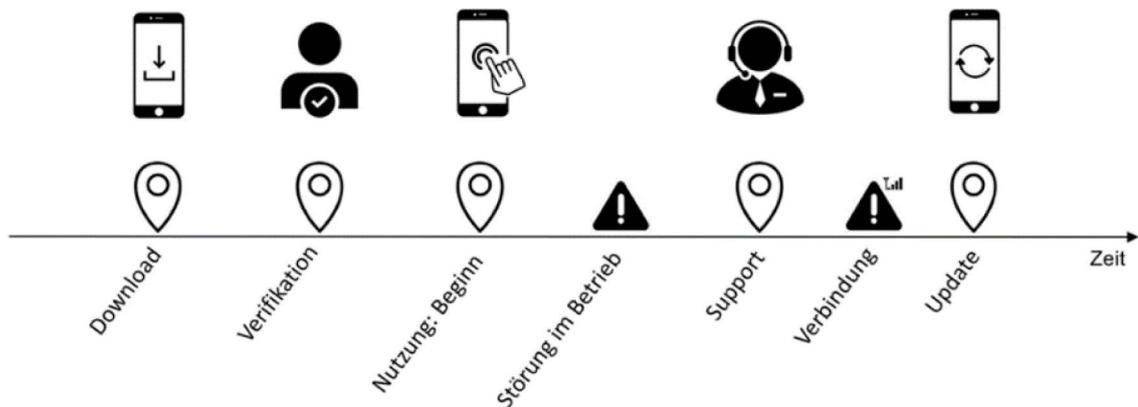


Abbildung 7.1: Fehler-ID Phasenmodell (Quelle: Eigene Darstellung)

## 2) Verifikation

Nachdem eine Applikation auf dem Endgerät installiert wurde, ist ein herstellerspezifischer Verifizierungsprozess zu durchlaufen. Er dient zur Absicherung eines gesetzeskonformen Betriebs und zur Erfüllung von sicherheitsrelevanten Aspekten. Sowohl die Prozessgestaltung als auch die notwendigen Maßnahmen zur Verifikation legt der Automobilhersteller individuell fest. Zur Verifikation kann die Registrierung mit Erstellung eines Nutzerkontos, die Anmeldung (Login), die Aktivierung oder Freischaltung von Fahrzeugkomponenten durch den Fahrzeughersteller oder die Zustimmung zu AGBs, Nutzungsbedingungen und Zugriffsrechten dienen. Für eine erfolgreiche Verifikation sollten die Prozessschritte transparent gestaltet werden. Allgemein findet in der Verifizierungsphase eine Legimitation mit persönlichen Kundendaten statt, um die mobile Anwendungssoftware nutzen zu können. Die Verifizierung kann ein kostenpflichtiges Abonnement oder aufpreispflichtige Sonderausstattungsmerkmale im Fahrzeug voraussetzen. Nach der erfolgreichen Verifikation beginnt die Nutzungsphase mit einem funktionalen Mehrwert für den Kunden. Auftretende Fehler führen in der Regel zum Nutzungsausfall.

## 3) Nutzung

Die Nutzungsphase beginnt in der Regel nach dem erfolgreichen Loginvorgang der Verifizierungsphase. Dem Nutzer stehen in dieser Phase Funktionen mit einem

Mehrwert zur Verfügung. Der funktionale Umfang einer Applikation ist ein herstellerepezifisches Produktmerkmal. Effektivität, Effizienz und Zufriedenheit mit der Zielerreichung von Funktionalitäten hängen von der Gebrauchstauglichkeit (Usability) ab. Wie das Interface einer Applikation gestaltet ist, beeinflusst die Gebrauchstauglichkeit oder Benutzerfreundlichkeit und damit die Anfälligkeit für Bedienerfehler. Abweichungen von den Kundenerwartungen haben dagegen einen unmittelbaren Einfluss auf die Kundenzufriedenheit. Zentrale Kundenanforderungen sind die Usability, die Funktionalität, die Zuverlässigkeit und nutzungsbedingte Randbedingungen. Zu letzteren zählen Voraussetzungen oder Erscheinungen während der Nutzung wie beispielsweise ein kostpflichtiges Abonnement, hohe Leistungsanforderungen an das Smartphone, ein erhöhter Datenvolumen- und Batterieverbrauch des mobilen Endgeräts, eine Verfügbarkeit nur für bestimmte mobile Betriebssysteme, eine ständig notwendige Eingabe von Logindaten beim Öffnen der App oder eingeblendete Werbung. Nach der Fehlerdefinition für mobile Anwendungssoftware (s. Kap. 3.4) ist ein Fehler die Nichterfüllung verfügbarer funktionaler Produktmerkmale sowie die Abweichung von Kundenerwartungen. Fehler haben eine negative Auswirkung auf die Kundenzufriedenheit. Demnach werden sowohl funktionale Erweiterungswünsche als auch Verbesserungsvorschläge als Fehler gewertet.<sup>[140]</sup>

#### 4) Störung im Betrieb

Während der Nutzungsphase von Apps können beim Endkunden Fehler und Störungen zu funktionalen Beeinträchtigungen führen. Der Grad der Beeinträchtigung hängt vom Fehler ab und kann nicht pauschalisiert werden. Bei der Fehlererfassung werden die Fehlercodes in Abhängigkeit der klassifizierten Funktionalitäten (s. Anhang 16) vergeben. Mögliche Fehler und Störungen im Betrieb sind fehlerhafte Push-Mitteilungen, nicht transparente Fehlermeldungen, Programmabstürze oder ein Datenverlust bei den personalisierten Einstellungen. Im Rahmen der Fehleranalyse werden diese Fehler lokalisiert, erfasst und effektive Maßnahmen zur Beseitigung sowie Wiederherstellung der Produkt- und Prozessqualität eingeleitet. Das Fehlerklassifizierungsmodell ordnet die erfassten Fehler hinsichtlich ihrer Priorität. Dabei werden Fehlerschwere, Anzahl betroffener Nutzer und Fehlerfrequenz berücksichtigt.

#### 5) Support

Mit zunehmendem Stellenwert digitaler Dienste zählt der Support zu den bedeutsamen Dienstleistungen. Der Support ist eine organisatorische Einheit eines Unternehmens, die für den Kunden lösungsorientierte Beratungstätigkeiten ausführt. Die Dienstleistungen des Supports sind ein fester Bestandteil des After-Sales-Management und können für den Kunden entgeltlich oder unentgeltlich sein. Die Qualität des Supports hat einen direkten Einfluss auf die Kundenzufriedenheit und die langfristige Kundenbindung. Die Kontaktaufnahme kann schriftlich oder telefonisch erfolgen. Die Kunden stellen bei der Kontaktaufnahme Anforderungen bezüglich Erreichbarkeit und Wartezeiten. Bei einer schriftlichen Anfrage wird eine Empfangsbestätigung erwartet. Die Gründe für die Kontaktaufnahme sind vielfältig, mitunter zählen dazu Informationsauskunft, Unterstützung bei Anwendungsproblemen und Störungs- oder Fehlerbeseitigung bei

auftretenden Abweichungen. Bei den Fehlercodes wird zwischen Störungs- und Fehlerbeseitigung, Unterstützung bei einem Anwendungsproblem und Informations- oder Fachauskunft unterschieden. Die Dienstleistung kann mittels technischer Instrumente zur Fernwartung erfolgen. Das Ergebnis des Supports hängt stark von der Kompetenz des Mitarbeiters wie auch den Tools zur Fehleranalyse und zur Wiederherstellung der funktionalen Dienstbereitschaft einer Anwendungssoftware ab. Die Bearbeitungszeit ist ein Leistungsmerkmal des Supports. Kunden fordern sowohl Transparenz bezüglich des Bearbeitungsstatus als auch Feedback bei erfolgreicher Fehlerbeseitigung.<sup>[141]</sup>

#### 6) Verbindung

Die technische Ausführung von Funktionen setzt eine Datenverbindung voraus, um Informationen und Befehle übertragen zu können. Beim Datenaustausch können Verzögerungen und Unterbrechungen auftreten. Die technische Infrastruktur für den Datenverkehr bietet der Mobilfunkstandard (UMTS). Die geografische Position des Senders und Empfängers ist ausschlaggebend für die Leistung beim Datenverkehr.

#### 7) Update

Mit einem Update wird der Quelltext einer Anwendungssoftware (halb-)automatisch aktualisiert. Solche Aktualisierungen erfolgen in der Regel zyklisch, führen Fehlerkorrekturen und Verbesserungen durch und können den Funktionsumfang einer Applikation ändern. Das Entwicklerunternehmen einer App bietet das Update im betriebssystemabhängigen Vertriebskanal an. Dort muss es heruntergeladen und installiert werden, dazu muss das mobile Betriebssystem die Installation unterstützen. Die Aktualisierung erfolgt in Versionen. Die Kunden stellen steigende Qualitäts- und Leistungsanforderungen (Kano-Modell). Aufgrund von komplexen Quelltextstrukturen kann es nach einer Aktualisierung zu Fehlern und Störungen kommen. Diese Abweichungen haben eine negative Auswirkung auf die Kundenzufriedenheit.

## 7.3 Modelldarstellung des Fehlerbeseitigungsprozesses

Für einen effektiven und effizienten Umgang mit Fehlern muss ein vierstufiges Grundschema beachtet werden:

- Standardisierte Erfassung und Dokumentation von Fehlern
- Einheitliche, zweckmäßige Klassifizierung und Analyse von Fehlern
- Effektive Fehlerbeseitigungsmaßnahmen
- Maßnahmen zur präventiven Fehlervermeidung oder frühzeitigen Fehlerentdeckung

### 7.3.1 Integration des Fehlerbeseitigungsmodells in den Produktlebenszyklus

Zur Absicherung und Steigerung der Kundenzufriedenheit sollte ein einheitliches Fehleranalysemodell in den Produktlebenszyklus einer mobilen Anwendungssoftware integriert werden. Es nutzt unterschiedliche Informationsquellen der Entwicklungs- und

Nutzungsphase hinsichtlich auftretender Fehler. Den Input liefern Kunden und Experten. Ein Experte ist ein Mitglied des Entwicklungsteams für mobile Softwareanwendungen. Die eruierten Fehler werden mit einem selbstentwickelten Tool (s. Anhang 17) standardisiert erfasst und dokumentiert. Die standardisierte Fehlererfassung und -dokumentation gewährleistet einen einheitlichen Qualitätsstandard. Ein zweckmäßiges Klassifizierungsmodell priorisiert die erfassten Fehler für einen anschließenden Fehlerbehebungsprozess und bewertet deren Auswirkungen auf die Kundenzufriedenheit. Im Fehlerbehebungsprozess wird ein standardisiert erfasster und zweckmäßig klassifizierter Fehler zu einem verantwortlichen Bearbeiter in der entsprechenden Fachabteilung kanalisiert. Das Ziel ist die Wiederherstellung oder Steigerung der Produktqualität und der Kundenzufriedenheit.

Im Rahmen der Fehleranalyse müssen auftretende Fehler und Störungen zunächst eruiert werden. Die nachstehenden Inhalte dienen als Informationsquelle:

1) Entwicklungsphase:

- Usability-Tests für die Bewertung der Gebrauchstauglichkeit
- Funktionale Testphase zur Sicherstellung von funktionalen Produktmerkmalen
- QM-Methoden zur Qualitätssicherung

2) Nutzungsphase (Schwerpunkt dieser Arbeit):

- Kundenrezensionen auf den Vertriebsplattformen für Apps
- Support-Tickets des Kundenservice
- Use-Case-Monitor im Feld, die mobile Softwareanwendungen nutzen

Usability-Testverfahren gliedern sich in experten- und nutzerbasierte Verfahren, wobei unterschiedliche Methoden zur Evaluation eingesetzt werden. Bei einer heuristischen Evaluation decken Experten Usability-Probleme auf. Nutzerbasierte Verfahren bewerten den Grad der Aufgabebearbeitung, den Prozentsatz von erfolgreich abgeschlossenen Aufgaben, die Anzahl von Fehlern während der Aufgabenbewältigung oder andere spezifische Indikatoren zur Qualitätsmessung. Bei der Vorgehensweise von Usability-Testverfahren wird zwischen Beobachtungs-, Interview- und Fragebogenverfahren sowie weiteren spezifischen Methoden unterschieden. Das Ziel von Usability-Testverfahren ist, die effektive, effiziente und zufriedenstellende Zielerreichung zu bewerten. Die Usability-Tests sollten in der frühen Entwicklungsphase erfolgen, um Usability-Probleme frühzeitig zu identifizieren, zu beheben und die Fehlerfolgekosten möglichst gering zu halten (formative Evaluation). In Abhängigkeit des Entwicklungsfortschritts wird zwischen einer formativen und summativen Evaluation unterschieden.<sup>[142]</sup> In einer funktionalen Testphase wird eine entwickelte Anwendungssoftware vor Veröffentlichung auf die Nichterfüllung von verfügbaren funktionalen Produktmerkmalen untersucht. In einem Lastenheft sind die funktionalen Produktmerkmale einer Anwendungssoftware als Anforderung festgelegt. Qualitätsmethoden sollen Produkt- und Prozesseigenschaften in der Entwicklungsphase absichern und verbessern. Der Einsatz von QM-Methoden zielt auf Maßnahmen zur Vermeidung oder Beseitigung möglicher und aufgetretener Fehler ab (u. a. FMEA, FTA, Ishikawa, QFD, SPC, 8D). Die bekannten QM-Methoden müssen auf ihre

Übertragbarkeit für die App-Entwicklung geprüft werden. Während der Entwicklungsphase erkannte Fehler sollten allgemein im Rahmen der Qualitätssicherung dokumentiert und untersucht werden.<sup>[142]</sup>

Das Bewertungssystem der Vertriebsplattformen für Apps setzt sich aus Wertungen und Rezensionen zusammen. Eine schriftliche Rezension ergänzt die numerische Bewertung eines Produktes. Die Automobilhersteller können mit ihrer Hilfe die Fehler und Störungen beim Kunden eruieren. Im Kapitel 7.3 werden Kundenrezensionen ausgewertet. Während der Nutzungsphase können Kunden den Support bei auftretenden Fehlern und Störungen telefonisch oder schriftlich kontaktieren. Ein Kunde kann eine schriftliche Anfrage über unterschiedliche Instrumente kanalisieren, dazu zählen Twitter, Messenger, Community und E-Mail (s. Anhang 18). Je nach verfügbaren technischen Systemen und Fehler findet eine Fehleraufnahme oder -behebung statt. Im Falle einer Fehleraufnahme sollte der Fehler anschließend an einen verantwortlichen Bearbeiter übermittelt werden. Ein Ticketsystem kann die Übermittlung koordinieren.

Nach der Veröffentlichung der mobilen Anwendungssoftware beginnt die Testphase im Feld. Mit geeigneten Prüfverfahren können Experten eine App im Feldversuch bezüglich auftretender Fehler und Störungen untersuchen. Erkannte Fehler sollten analysiert und zeitnah behoben werden, um den Schaden für den Kunden zu begrenzen. Eine bedeutende Grundlage des zukünftigen Erfolgs der Automobilhersteller ist die Steigerung der Kundenzufriedenheit.

### **7.3.2 Fehlererfassungsmaßnahmen in der IT-Branche**

Digitale Wertschöpfungsstufen und Geschäftsmodelle bilden bereits heute eine Kernkompetenz der IT-Branche. Die Unternehmen erreichen im numerischen Benchmarking-Leistungsvergleich eine hohe Kundenzufriedenheit. Die digitale Expertise der IT-Branche soll zur Steigerung der Kundenzufriedenheit in die Automobilindustrie einfließen. Die IT-Branche wird auf übertragbare Maßnahmen zur Steigerung der Kundenzufriedenheit untersucht. Für eine erweiterte Fehlerermittlung können zusätzliche Informationsquelle erschlossen werden:

#### (1) Entwicklungsphase:

- GitHub-Experten-Absicherungsmaßnahmen
- Issue-Ticket-System zur Fehlerverwaltung
- Fehlerbehebungs-Bewertungssystem zur Leistungsbewertung der verantwortlichen Bearbeiter

#### (2) Nutzungsphase:

- Feedback-Fragebögen des Automobilherstellers
- Community-Experten auf strukturierten Kundenplattformen
- Numerische Auswertung von Trackingdaten der Nutzer

GitHub ist ein Anbieter für eine Entwicklungsplattform zur ergebnisorientierten „Open Source“- oder „Business“-Softwareentwicklung. Die Struktur von GitHub erlaubt es, standort- und zeitzoneunabhängige Entwicklungsteams zu bilden. Die Teammitglieder

sind miteinander vernetzt und arbeiten gemeinsam sowie zielorientiert an der Entwicklung von Softwarecodes. Zur Qualitätssicherung werden Änderungen am Code erst nach einem Freigabeprozess vorgenommen. Dabei überprüfen sich die Teammitglieder (Experten) gegenseitig. Fehlerhafte Codes werden durch die Kontrollmaßnahmen zuverlässig erfasst, beseitigt und können im fortlaufenden Entwicklungsprozess verfolgt werden. Zahlreiche IT-Unternehmen aus Silicon Valley oder anderen Bereichen der IT-Branche nutzen die Plattform für ihre Softwareentwicklung (u. a. airbnb, IBM, SAP, PayPal, Spotify oder Bloomberg).<sup>[144]</sup>

Ein Issue-Ticket-System dient zur Qualitäts- und Leistungssteigerung im Fehlerbeseitigungsprozess und wird in der Regel im Support eingesetzt. Nach einer Fehleraufnahme werden die daraus folgenden Aufgaben in ein Ticketsystem übertragen. Dieses koordiniert die Verwaltung, Überwachung und Dokumentation von Aufgaben zur Fehlerbeseitigung. Ein Ticket wird zu einem verantwortlichen Bearbeiter kanalisiert. Dort findet die Fehlerbeseitigung statt. Mit dem Issue-Ticket-System entsteht eine Schnittstelle zwischen Fehleraufnahme und Fehlerbehebung. Der Bearbeitungsstatus einer Aufgabe ist transparent und kann digital überprüft werden. Die Maßnahmen zur Fehlerbeseitigung sind bei einer Anwendungssoftware komplex und können oftmals nur über Workflows bewältigt werden. Dabei handelt es sich um arbeitsteilige Aktivitäten, die abteilungsübergreifend erfolgen können.<sup>[141]</sup>

Das Einführen eines Fehlerbehebungs-Bewertungssystems dient zur Leistungsbewertung der Fehlerbehebung. Der für die Fehlerbehebung verantwortliche Bearbeiter stellt einen internen Lieferanten dar, der anhand von Leistungsindikatoren bewertet werden kann. Eine vergleichbare Bewertung findet bereits heute im Lieferantenmanagement statt. Im Einkauf wird beispielsweise die Leistung der Lieferanten nach dem Datum des Wareneingangs gemessen und im SAP-Tool bewertet. Für die Fehlerbehebung von Apps der Automobilindustrie müssen zweckmäßige Leistungsindikatoren entwickelt werden. Ein möglicher Kennwert ist beispielsweise die Bearbeitungszeit der Fehlerbehebung. Mit einer Leistungsbewertung und einer anschließenden Auswertung können Maßnahmen zur Leistungssteigerung definiert werden.

Mit digitalen Feedbackfragebögen können Kunden Produkte und Dienstleistungen eines Automobilherstellers bewerten. In der IT-Branche werden bereits heute Feedback-Bewertungen eingesetzt, um sowohl die Kundenzufriedenheit als auch die Kundenanforderungen zu erfassen (s. Anhang 19). Der Kunde kann über unterschiedliche Kanäle erreicht werden. Ein digitaler Feedback-Fragebogen kann demnach beim Öffnen der Applikation eingeblendet werden. Eine weitere Möglichkeit ergibt sich nach Kontaktaufnahme des Supports in Form einer kundenbezogenen E-Mail. Feedbackfragebögen stellen eine weitere Informationsquelle dar, um Fehler beim Endkunden in der Nutzungsphase zu eruieren.

Zahlreiche Unternehmen der IT-Branche bieten dem Kunden eine digitale Community-Plattform an. Sie bietet unter anderem Ankündigungen zu aktuellen Störungen,

Hinweise zur Installation und Nutzung, Empfehlungen sowie weitere Informationen. Der strukturelle Aufbau ähnelt sozialen Netzwerken wie Facebook. Aktive Nutzer sind über diese Plattform miteinander vernetzt und haben die Möglichkeit, Fragen zu stellen oder Fehler zu melden. Die Community-Plattform bezieht sich auf eine Dienstleistung oder ein Produkt. Das Nutzerverhalten des Kunden sollte mit einem Tracking-Management System überwacht werden. Tools können Datensätze zu ausgeführten Funktionen erfassen und dokumentieren. In einem Folgeprozess sollten die numerischen Datensätze statistisch ausgewertet werden, um den verfügbaren Funktionen durchschnittliche Ausführungen zuzuordnen. Kommt es bei der Überwachung der Datensätze zu Abweichungen, können diese auf Fehler zurückgeführt werden. Im Falle der Standheizungsfunktion kann beispielsweise eine rückläufige Anzahl von Ausführungen auf einen Fehler oder auf veränderte Witterungsbedingungen hinweisen.

Neben der standardisierten Erfassung und Dokumentation von Fehlern trägt eine frühzeitige Fehlererkennung zum Erfolg von digitalen Dienstleistungen in der Automobilindustrie bei. Um im Feld auftretende Fehler frühzeitig zu erkennen, sollte für das Fehlermonitoring eine eigens entwickelte Watchlist (vgl. Appbot, Adjust) verwendet werden. Sie erlaubt es, die Kundenzufriedenheit durch eine Auswertung der Leistungskennzahlen des Wertungssystems der Vertriebsplattformen numerisch zu überwachen. Veränderte Leistungskennzahlen können auf einen Fehler im Feld hindeuten. Der Grad der Veränderung dient als Indikator für die Methode „Trend und Prognose“ der frühzeitigen Fehlererkennung.

### **7.3.3 Auswertung von Kundenrezensionen**

Im Rahmen der Fehleranalyse werden Kundenrezensionen bezüglich auftretender Fehler beim Endkunden während der Nutzungsphase untersucht. Für eine einheitliche Fehlererfassungsqualität, d. h. eine vom Erfasser unabhängige und konstante Dokumentationsqualität, wird ein Tool zur standardisierten Fehlererfassung verwendet. Die gemäß Fehlerkatalog erkannten Fehlertypen werden dadurch standardisiert erfasst (s. Anhang 19). Allgemein weist jeder Automobilhersteller spezifische Fehler und Abweichungen auf. Bei zusammenhängenden Konzernmarken ist ein Muster bei den erfassten Fehlern zu erkennen. Die Anzahl von verfügbaren Kundenrezensionen variiert je nach Automobilhersteller, auch reagieren diese unterschiedlich auf Kundenrezensionen.

Ein zentrales Ziel der Fehleranalyse ist die zweckmäßige Klassifizierung der erfassten Fehler, um eine einheitliche und möglichst standardisierte Entscheidungshilfe für den Umgang mit Fehlern zu bieten. Um die beschränkten Kapazitäten in der Organisationseinheit effektiv einzusetzen, dient die Methode der Fehlerklassifizierung der effektiven Wiederherstellung beziehungsweise zur Steigerung der Kundenzufriedenheit. Die im Feld auftretenden Fehler und Abweichungen werden im Klassifizierungsmodell auf ihre Auswirkung auf die Kundenzufriedenheit bewertet, um anhand der Priorisierung Fehlerbeseitigungsmaßnahmen zielgerichtet, effizient und effektiv einzuleiten (s. Anhang 18). In der Übersicht (s. Anhang 20) werden die

standardisierten Fehler verschiedener Unternehmen erfasst. Diese sind nach Automobilhersteller, Mobilitätsdienstleister und weiteren Unternehmen kategorisiert. Aufgrund einer ungleichen numerischen Verteilung in den Kategorien wurde für die Vergleichbarkeit ein Referenzwert berechnet, der sich auf die prozentuale Häufigkeit betroffener Unternehmen bezieht. Im Vergleich zur IT-Branche treten bei den Automobilherstellern häufiger Fehler bei Verifikation, Störung im Betrieb und Verbindung auf. Die folgenden standardisierten Fehler sind bei den meisten Automobilherstellern vorzufinden:

- 1) Zeit Verbindungsaufbau (16/17)
- 2) Verbindungsaufbau/Datenaustausch/Kopplung (16/17)
- 3) Zuverlässigkeit (12/17)
- 4) Wiederholte Eingabe von Logindaten (11/17)
- 5) Fehler Funktion „Statusabfrage“ (8/17)
- 6) Fehler Anmeldung/Login (7/17)

Die Reihenfolge der genannten Fehler steht nicht im Zusammenhang mit der Auswirkung auf die Kundenzufriedenheit. Um diese in der Automobilindustrie zu steigern, sollten folgende Maßnahmen als Handlungsempfehlung aufgegriffen werden:

- Einholung der Zustimmung der Kunden zur Erhebung von Analyse-, Diagnose- und Nutzungsdaten
- Basic-Sessions zur Produkterklärung (Bedienerfehler reduzieren)
- Video-Verifikationsanleitung im Vertriebskanal (Verifikation)
- Fahrzeug-Empfangsanzeige (Verbindung)
- Integration von Feedbackfragebögen und Supportkontakt in Apps

Für eine toolunterstützte Fehleranalyse ist die Zustimmung der Kunden zur Erhebung von Analyse-, Diagnose- und Nutzungsdaten fundamental. Dabei wird im App-Betrieb ein paralleler Algorithmus zur Fehlererkennung ausgeführt. Wird ein Fehler identifiziert, sendet die Anwendungssoftware dem Entwickler automatisch einen Fehlerbericht zu. Der Kunde ist Dateneigentümer und kann seine Daten dem Automobilhersteller durch Einverständniserklärung für Analysezwecke offenlegen (vgl. Apple). Zahlreiche Unternehmen der IT-Branche nutzen derartige Systeme zur automatischen Fehleranalyse.<sup>[12]</sup>

Um Bedienerfehler während der Nutzungsphase zu reduzieren, Kundenerwartungen zu erfassen und den Vertrieb zu stärken, sollten Unternehmen den Kunden Veranstaltungen zur Produkterklärung anbieten (Basic-Sessions), in deren Rahmen die Produkte und die Dienstleistungen vorgestellt werden. Eine Produktvorstellung kann die Vorführung von verfügbaren Funktionen enthalten. Zahlreiche Unternehmen der IT-Branche bieten dem Kunden bereits heute eine auf das erworbene Produkt bezogene Veranstaltung an (u. a. Apple, o2 live store). Sie weisen meist einen Schulungscharakter auf, d. h. ein Experte stellt das Produkt oder die Dienstleistung vor und die Kunden besuchen diese Veranstaltung als Zuhörer. Um Bedienungsfehler während der Verifikation zu senken, sollten die Automobilhersteller den Kunden über den

Verifizierungsprozess aufklären. Eine Videoanleitung im Vertriebskanal kann den Kunden über den herstellerspezifischen Verifikationsprozess, bestehend aus gegebenenfalls Registrierung, Anmeldung/Login und Aktivierung/Freischaltung, informieren. Die Videoanleitung sollte ebenfalls Informationen über die definierten Anforderungen an das Fahrzeug enthalten, um Digital Services nutzen zu können. Eine in der App integrierte Fahrzeug-Empfangsstatusanzeige kann das Verständnis für situationsabhängige Abweichungen fördern. Kunden sind in der Regel mit der Empfangsanzeige vertraut. Sollte sich das Fahrzeug in einer Tiefgarage befinden, sind die digitalen Dienstleistungen aufgrund von mangelnder Netzabdeckung eingeschränkt oder nicht verfügbar. Dieser Empfangszustand würde in der App angezeigt werden.

## **7.4 Weiterentwicklung des agilen Fehlermanagementprozesses**

Im Folgenden wird ein Entwurf eines Fehlermanagementprozesses vorgestellt. Ein wesentlicher Bestandteil ist dabei ein zentraler, automatisiert gefüllter Fehlerspeicher, dessen Hauptaufgabe darin besteht, die Basisinformationen bereitzuhalten, auf deren Basis schnell entschieden werden kann, ob es sich bei einem Fehler um einen Fehler auf Gesamtsystemebene oder um einen Fehler handelt, der einem bestimmten Front- oder Backend zuzuordnen ist. Ziel ist es somit, Fehler schnell und zielgenau einem Verantwortungsbereich (verantwortlich im Sinne der Fehlerbehebung) zuweisen zu können.

### **7.4.1 Entwurf eines agilen Fehlermanagementprozesses**

Der zentrale Fehlerspeicher wird idealerweise automatisiert mit (vor allem technisch abgreifbaren) Informationen gefüllt. Dabei werden Logdateien der Server bzw. Backends, CRM-Informationen und die Funktionen der digitalen Anwendungen mit Fehlerbildern, die durch Softwaretest und während des Betriebes gesammelt werden, verknüpft. Um die Software ausreichend testen zu können, muss eine Testumgebung zur Verfügung stehen, die es den einzelnen Teams, bestehend aus Developer, Operator, Tester etc., ermöglicht, den Code und neue Funktionen ständig auch gesamtsystembezogen zu testen. Diese Umgebung sollte als „Infrastructure as Code“ die Konzepte des Continuous Testing, Continuous Deployment und Continuous Delivery ermöglichen.

Am Ende der jeweiligen Sprintzyklen können die Teams ihre Inkremente in der Gesamtsystem-Testumgebung testen. Die dabei entstehenden Fehlerbilder werden in den zentralen Fehlerspeicher aufgenommen, sodass dort eine Fehlerhistorie jeder App bzw. digitalen Anwendung in jeder Entwicklungsstufe entsteht. Da die Teams die auftretenden Fehler nach dem Testen beheben, können die Lösungen mit den Fehlerbildern verknüpft werden, sodass eine ganzheitliche Fehlerhistorie entsteht, die Fehlerbilder und Lösungsmaßnahmen über alle Teams hinweg verknüpft. Damit entsteht eine große Menge an Daten rund um das Thema Fehlerallokation und -lösung (bis hin zu Big-Data-Dimensionen). Mithilfe von KI können im zentralen Fehlerspeicher Muster

erkannt und durch Mustererkennung bei zukünftig auftretenden Fehlerbildern entsprechende wahrscheinliche Zuständigkeiten abgeleitet werden.

Der zentrale Fehlerspeicher benötigt speziell ausgebildete Systemarchitekten, die für die Gesamtorchestrierung der Front- und Backendlandschaft verantwortlich sind, die Fehlerursachen lokalisieren und Verantwortlichkeiten bestimmen. Ein weiterer Vorteil einer Testumgebung, die das Gesamtsystem abbildet, ist, dass die agil arbeitenden Teams Anbindungen und Schnittstellen schon früh während der Entwicklung beachten müssen. Das sorgt für eine Standardisierung des Entwicklungsprozesses über die Teams hinweg (Es entstehen damit auch gesamtsystembedingte Fehlermuster, die für frühe Phasen der Anwendungsentwicklung typisch sind).

Während der Entwicklung und Programmierung der digitalen Anwendung in kleinen Teams finden Softwaretest zunächst auf kleiner Ebene statt. Hier wird der Code auf Fehler überprüft. Eine systemübergreifende Kommunikation findet erst statt, wenn das erste funktionierende Inkrement der Anwendung vorgestellt wird. Da dies in der Regel am Ende eines Sprints der Fall ist, kann hier das lauffähige Inkrement mithilfe des Gesamtsystem-Testautomaten auf Fehler überprüft werden. Fehler, die hier auftreten, werden automatisch an den zentralen Fehlerspeicher übermittelt. Dieser legt eine Liste von Fehlerbildern an, wie sie beispielsweise beim Vorgang „Türöffnung via Smarphone“ auftreten. Die entsprechenden Fehler werden im nächsten Sprint vom Entwicklerteam analysiert und behoben. Die Dokumentation der Abstellmaßnahmen findet während der Sprintzyklen automatisiert im Product Backlog statt und wird dem entsprechenden Fehlerbild zugeordnet.

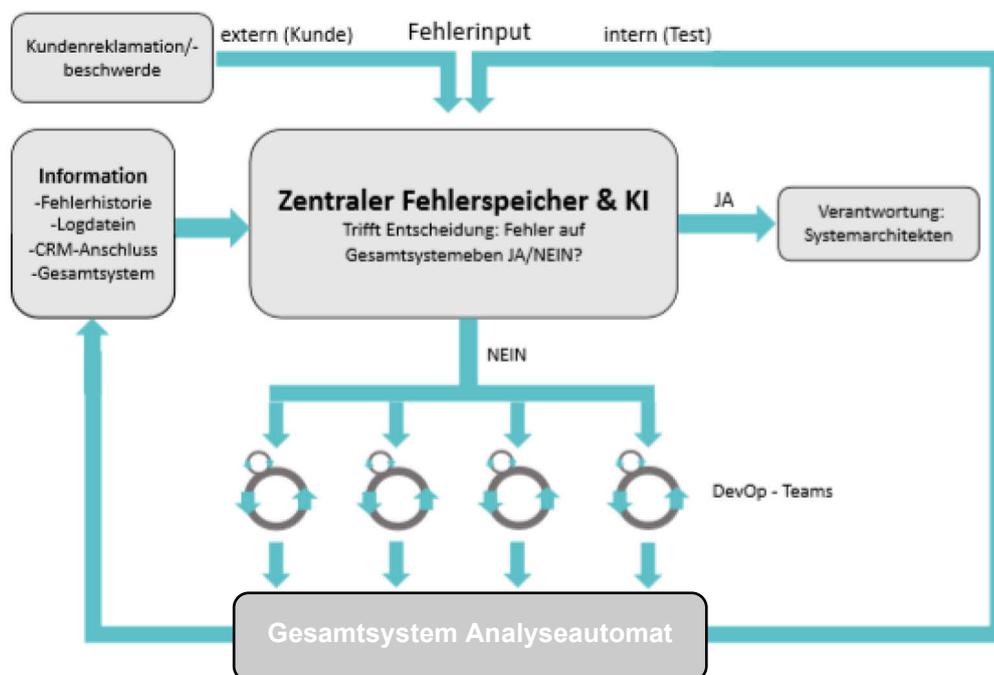


Abbildung 7: Entwurf eines agilen Fehlermanagementprozesses (Eigene Darstellung vgl. Forschungsstudie QSK der TU Berlin)<sup>[20]</sup>

Je mehr Fehler entstehen und je mehr Abstellmaßnahmen erfolgen, desto mehr Daten existieren, die mittels Mustererkennung durch KI ausgewertet werden können. Die Betrachtung der Logdateien von beteiligten Backendsystemen, Rüstzeiten und häufig auftretenden Fehlern an bestimmten Schnittstellen, die bei einem Vorgang „Türöffnung via Smarphone“ auftreten, ergeben ein ganzheitliches Fehlerbild. Wenn nun bei der Ausführung des Vorgangs „Türöffnung via Smarphone“ ein Fehler im Betrieb, d. h. bei einem Kunden, auftritt, kann auf einen großen Datenpool zurückgegriffen werden. Hier sollte eine Anbindung an das CRM-System eingerichtet werden, um schnell Informationen über den Kunden, wie Fahrzeug-ID, Betriebssystem, Version etc., zur Verfügung zu haben. Mit Hilfe der Fahrzeug-ID kann der Vorgang zeitlich eingegrenzt werden. Die entsprechenden Logdateien der Backends ergeben zusammen mit der Fehlerbeschreibung ein umfangreiches Fehlerbild, das mit den vorherigen Fehlerbildern verglichen werden kann, um so mithilfe der KI den Fehler zumindest auf ein bestimmtes System eingrenzen zu können und klare Verantwortlichkeiten abzuleiten.

Zusammenfassend lässt sich feststellen, dass der hier vorgestellte Fehlermanagementprozess darauf abzielt, die über einen längeren Zeitraum gewonnenen Daten zu nutzen, um eine schnelle Fehlerallokation während des Betriebes zu ermöglichen. Dabei muss ein möglichst hoher Automatisierungsgrad erreicht werden, der durch Unterstützung von KI ermöglicht werden kann. Eine Zukunftsvision, die durch weitere Entwicklungen im Bereich der KI und Testautomatisierung erreicht werden könnte, ist ein vollautomatisierter Fehlerabstellprozess (selbstreparierende Software).

#### **7.4.2 Allgemeine Ideen, Hypothesen und Ansätze**

Im folgenden Teil werden Ideen, Hypothesen und Ansätze aufgezeigt, die nicht direkt dem Thema agiles Fehlermanagement zugeordnet werden können, sich jedoch im Laufe der Arbeit ergeben haben und aus denen weitere Forschungsansätze entstehen könnten:

- 1) Als Vorbild für die Automobilindustrie im Bereich zukünftiger digitaler Entwicklungen kann ein Blick Richtung Apple nützlich sein. Apple optimiert und entwickelt die eigene Software in Abstimmung mit der Hardware. Dieses Konzept ist im Hinblick auf Maintenance, Performance und Qualität von Vorteil. Microsoft oder auch Google (Android) hingegen entwickeln Betriebssysteme und Software, die zum Großteil nicht auf eine bestimmte, eigenproduzierte Hardware abgestimmt sind, sondern stellt diese für ein breites Spektrum an Hardwarelieferanten (PCs und Smartphones) zur Verfügung. Ein Vergleich dieser unterschiedlichen Ansätze von Apple vs. Microsoft/Google, auch mit Blick auf zukünftige Entwicklungs- und Geschäftsmodelle im Bereich der Automobilindustrie, könnte interessante Einblicke liefern.
- 2) Die Umstellung auf und Verwendung von agilen Methoden wird in vielen Management- und Entwicklungsbereichen umgesetzt und vorangetrieben. Die rein agilen Herangehensweisen der Softwareentwicklung haben jedoch auch Nachteile in Bezug auf Termin- und Kostenüberschreitungen sowie beim Fokus auf das Gesamtsystem. Daher gibt es Ansätze für sogenannte hybride Strategien. Sie

versuchen die Vorteile der Wasserfallstrategie mit agilen Entwicklungsmethoden zu kombinieren. Softwareanbieter wie agosense stellen hier Tools zur Verfügung, die diesen Ansatz verfolgen und Plug-Ins für JIRA und weitere Fehlermanagementtools anbieten.<sup>[145]</sup> agosense ist ein Unternehmen, welches im Zuge der Literaturrecherche gefunden wurde. Es bietet sehr aktuelle Softwarelösungen an, die im Bereich des agilen Fehlermanagements mit Fokus auf Kunden- und Lieferantenbeziehung anzusiedeln sind. Ein Blick in das Whitepaper und zukünftigen Produkte, die agosense plant, sind empfehlenswert.

- 3) Im Rahmen der Arbeit wurde auf das Konzept des Continuous Deployment eingegangen, welches auf eine möglichst schnelle automatisierte Durchführung von Tests und Implementierung in die Produktivumgebung abzielt. Die Auslieferung in die Produktivumgebung könnte dabei gestaffelt stattfinden. So könnten komplett neue Funktionen, Apps oder digitale Dienste erst den eigenen Mitarbeitern und anschließend einer aufzubauenden Testcommunity als Beta-Version zur Verfügung gestellt werden. Diese Testcommunity könnte direkt Feedback zu den neuen Funktionen geben. Als Anreiz könnten hier kostenlose digitale Zusatzfunktionen, Werkstattbesuche etc. angeboten werden. Updates können dann sukzessive für einen immer größeren Kundenstamm zur Verfügung gestellt werden. Somit würden Fehler, die zu Beginn der Produktivsetzung auftreten, nur einen kleinen Teil der Kunden betreffen und ein Imageschaden ließe sich so vermeiden. Die entsprechenden Fehlerbilder könnten im Sinne des vorgestellten Fehlermanagementprozesses die KI mit Daten füttern.
- 4) Da sich Fehler im Bereich der digitalen Anwendungen nie gänzlich vermeiden lassen, spielen Sofortmaßnahmen eine wichtige Rolle. Dies könnte eine sofortige Mitteilung an den Kunden beim Auftreten eines Fehlers sein, die ihn darüber informiert, dass der Fehler registriert und dass ein Fehlerbericht an den Hersteller geschickt wurde, um das Problem möglichst schnell zu beheben. Dieses Vorgehen ähnelt einem Diagnosebericht, wie ihn auch Apple oder Microsoft anwenden. Dadurch, dass der Kunde direkt über die Registrierung des Fehlers informiert wird, können zum einen Imageschäden auf Kundenseite abgemildert werden, zum anderen gewinnt man Zeit, um den Fehler zu analysieren und zu beheben. Weiterhin könnte eine solche Maßnahme dazu beitragen, eine Überflutung der Beschwerdekanäle zu verhindern.

Im theoretischen Teil wurden Begriffe des Fehlermanagements definiert und Methoden des klassischen Fehlermanagements und im vierten Abschnitt klassische und agile Vorgehensmodelle der Softwareentwicklung vorgestellt. Dabei wurde ein Ansatz beschrieben, wie Softwarefehler gehandhabt werden können, die in Softwareentwicklungsprojekten auftreten, die nach der Scrum-Methode ablaufen. Außerdem wurde im vierten Abschnitt der DevOps-Ansatz vorgestellt. Dieser beinhaltet vielversprechende Methoden, die einen ganzheitlichen Ansatz liefern, um die Entwicklung und das Fehlermanagement von Software zu optimieren. Da das Thema sehr aktuell ist und wenig Literatur zu finden ist, auf die bei der Entwicklung des

Fehlermanagementprozesses zurückgegriffen werden konnte, wurden drei qualitative Fachinterviews durchgeführt.

## **7.5 Weiterentwicklung Reklamationsmanagement als Teil des gesamten Fehleranalyseprozesses**

Eine interne Forschungsstudie mit dem Thema Weiterentwicklung Reklamationsmanagement im Feld-Connectivity im Fachgebiet Qualitätsstrategie und Qualitätskompetenz an der TU Berlin wurde im Jahr 2019 bearbeitet.

Die zentrale Forschungsfrage und Zielsetzung des vorliegenden Kapitels lautet demnach: Wie sollte der Reklamationsprozess in der Automobilindustrie hinsichtlich der fortschreitenden Connectivity weiterentwickelt werden, um auch zukünftig die Wiederherstellung der Kundenzufriedenheit zu gewährleisten? Die Connectivity begegnet dem Reklamationsmanagement dabei mit drei zentralen Herausforderungen. Die erste resultiert aus einer vermehrten Provokation von Lieferantenfehlern durch stetig wachsende Softwareumfänge und -komplexitäten. Die zweite ist dem besonderen Umstand der zumeist späten Fehlerentdeckung beim Kunden geschuldet. Die dritte Herausforderung begründet sich letztlich in den besonderen Umständen, mit denen die Connectivity-Software an das Reklamationsmanagement herantritt. Diese bestehen vor allem in angepassten und die Möglichkeiten der Technologie nutzenden Abläufen.

Das nachfolgende Kapitel fasst alle bisherigen Erkenntnisse zu einem weiterentwickelten Reklamationsprozess in der Automobilindustrie zusammen. Die präsentierte Vorgehensweise soll den Herausforderungen der Connectivity mit entsprechenden Bestandteilen und Ausprägungen begegnen und somit letztlich die Wiederherstellung der Kundenzufriedenheit gewährleisten.

Zunächst wird die allgemeine Zielsetzung aufgegriffen. Im Anschluss erfolgt die Darstellung des weiterentwickelten Reklamationsprozesses. Sie wird durch Schaubilder und dazugehörige Beschreibungen vorgenommen. Den inhaltlich letzten Abschnitt bildet eine Betrachtung möglicher Herausforderungen und Risiken, die bei der Umsetzung und Anwendung des angestrebten Ablaufs auftreten könnten.

### **Zielsetzung**

Die allgemeinen Ziele des Beschwerdemanagements, welche in Abschnitt 4.5 genannt wurden, gelten gleichermaßen für den weiterentwickelten Reklamationsprozess in der Automobilindustrie. Damit steht eine Erhöhung des Unternehmensgewinns und eine gesteigerte Wettbewerbsfähigkeit im Zentrum der Bemühungen. Die dazu nötige Wiederherstellung der Kundenzufriedenheit durch eine schnelle und nachhaltige Abstellung aufgetretener Fehler (direktes Beschwerdemanagement) sowie die Qualitätsoptimierung der Produkte und Dienstleistungen (indirektes Beschwerdemanagement) stellen entscheidende Eckpfeiler auf dem Weg zur Umsetzung dar. Zusätzlich gilt es, die besonderen Herausforderungen und Chancen der Connectivity in einer geeigneten Art und Weise zu berücksichtigen.

## Prozessdarstellung

Auf den nächsten Seiten erfolgen die Darstellung und Beschreibung des weiterentwickelten Reklamationsprozesses in der Automobilindustrie. In diesem Zusammenhang wird die Umsetzung der jeweiligen Zielstellung verdeutlicht. Aus Gründen der Vergleichbarkeit wird der Gesamtprozess in die bekannten Teilschritte „Initialisierung, Problemlösungsmethode, Verifikation/Abschluss“ sowie „Beanstandung ablehnen“ unterteilt. Eine Neuerung stellt dagegen der Teilprozess „Produktrückruf“ dar. Zusätzlich findet die aus der Ursprungsdarstellung bekannte Modellierungssprache der ereignisgesteuerten Prozesskette Verwendung. Das vollständige Schaubild ist dem Anhang 11 beigelegt. Zentrale Neuerungen bzw. Unterschiede des Prozesses sind in der Gesamtdarstellung gestrichelt eingefasst.

Im Gegensatz zum bestehenden Prozess von VDA besitzt das nachfolgend präsentierte Schaubild mit dem Lieferanten, OEM und Kunden drei anstatt zwei Prozessteilnehmer. Diese Einteilung beruht auf den aus Abbildung 2.3 bekannten Partnern im Produktentstehungsprozess. Die Bezeichnung des Kunden in der Darstellung des VDA wird im Folgenden von der Bezeichnung des OEM abgelöst. Unter dem Begriff „Kunde“ ist im weiterentwickelten Reklamationsprozess der Endverbraucher zu verstehen.

### 7.5.1 Teilprozess Initialisierung

Der Teilprozess der Initialisierung und damit der Gesamtprozess beginnt mit dem Auftreten eines potenziellen Softwarefehlers. Die erste Änderung gegenüber der aktuellen Ablaufbeschreibung besteht in der vorgeschalteten Beschwerdestimulierung. Der VDA geht in der 2009 veröffentlichten Prozessbeschreibung von einer Fehlerentdeckung vor Auslieferung an den Kunden aus. Bei digitalen Diensten ist dies häufig jedoch nicht möglich. Aus diesem Grund muss der Prozess für die Kommunikation möglicher Fehlfunktionen vom Kunden an den OEM Sorge tragen. Der diesbezügliche Ablauf ist in dem untenstehenden Abschnitt dargestellt.

Wie in Abschnitt 4.5 verdeutlicht, besitzt die Maximierung der Beschwerdeanzahl eine enorme Wichtigkeit. Der weiterentwickelte Prozess nutzt in diesem Zusammenhang erstmals die Möglichkeiten der digitalen Dienste, indem er die automatische Sendung der Fehlerdaten an den OEM anstrebt. Dadurch wird die Beschwerdestimulierung in einem hohen Maße verbessert. Die Automobilhersteller sind nicht länger ausschließlich auf die Initiative ihrer Kunden angewiesen, um über Fehler im Feld unterrichtet zu werden. Die Anzahl der Reklamationen würde auf diese Weise maximiert werden, wodurch ein zentrales Ziel des allgemeinen Beschwerdemanagements erreicht wird. Die diesbezüglichen Hürden der technischen Voraussetzungen und Datenschutzbestimmungen wurden bereits umrissen. An dieser Stelle im Prozess wird erstmals deutlich, welche Möglichkeiten mit den digitalen Diensten einhergehen. Diese müssen vollumfänglich genutzt werden.

Als Ergebnis des Prozessschrittes hat der OEM die Fehlerdaten (manuell oder automatisch) erhalten. Es ist jedoch auch der Fall möglich, dass der Kunde eine automatische Datensendung ablehnt und zudem keine Kommunikation des Fehlers an den OEM betreibt. In diesem Fall würde die Fehlfunktion unentdeckt bleiben und der Gesamtprozess fände bereits frühzeitig ein Ende. Aus diesem Grund ist eine automatische Datenübermittlung anzustreben. Erwähnenswert sei an dieser Stelle noch, dass die Kommunikation zunächst ausschließlich an dem OEM vollzogen wird. Damit obliegt die Datenhoheit weiterhin dem Automobilhersteller.

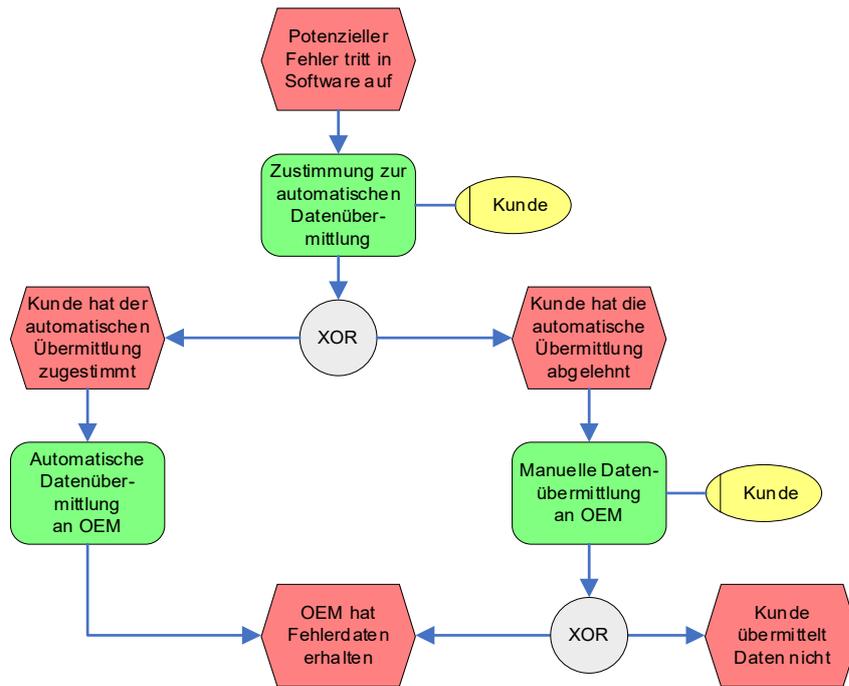


Abbildung 7.3: Übermittlung der Beanstandungsdaten an den OEM

Nach dem Erhalt der Fehlerdaten übermittelt der OEM die nötigen Informationen schnellstmöglich an den voraussichtlich verantwortlichen Lieferanten (S. Abb. 7.4). Auf diese Weise wird der Zulieferer frühzeitig in den Gesamtprozess eingebunden und ein beiderseitiges Verständnis der Problemstellung entsteht.

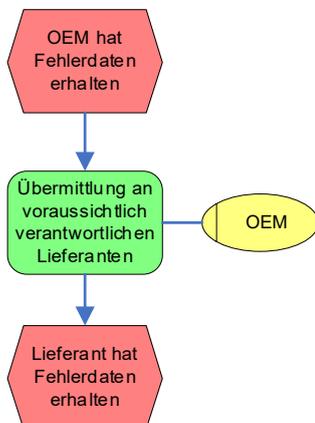


Abb. 7.4: Datenübermittlung an den Lieferanten

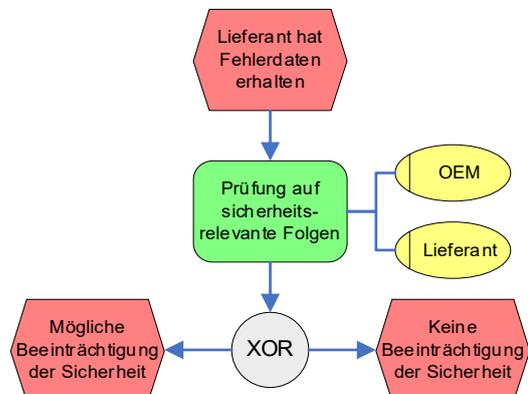


Abb. 7.5: Prüfung auf sicherheitsrelevante Folgen

Sobald der Lieferant die Daten erhalten hat, beginnt die parallele Arbeit entlang dreier Prozessstränge. Der wichtigste dieser Stränge sieht die Untersuchung des Fehlers auf womöglich sicherheitsrelevante Folgen vor (S. Abb. 7.5). Die Kontrolle wird über den gesamten Ablauf hinweg durchgeführt und hinsichtlich neuer Erkenntnisse stetig aktualisiert. Die diesbezügliche Vorgehensweise wurde von der Festlegung des Warenverwendungsentscheids im bestehenden Reklamationsprozess übernommen. Falls sicherheitsrelevante Folgen festgestellt oder nicht mehr ausgeschlossen werden können, ist ein unverzüglicher Produktrückruf durch den OEM nötig. (s. Abschnitt 7.5.5)

Gleichzeitig beginnt mit der Prüfung auf sicherheitsrelevante Folgen die Vorprüfung der Fehlerdaten. Wie die umseitige Abb. 7.6 verdeutlicht, arbeiten OEM und Lieferant auch während dieser Aufgabe eng zusammen. Im aktuellen Prozess von VDA wird dieser Schritt ausschließlich vom Zulieferer durchgeführt.

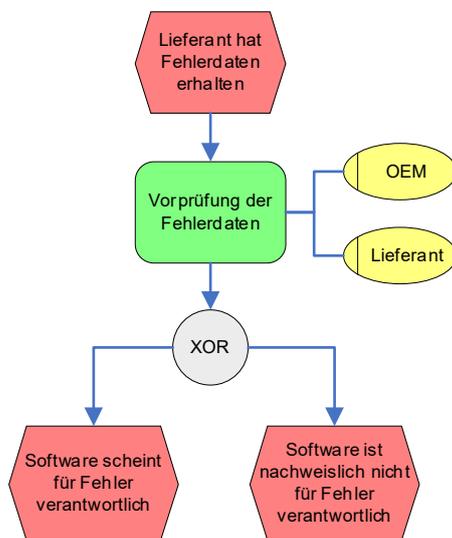


Abb. 7.6: Vorprüfung der Fehlerdaten

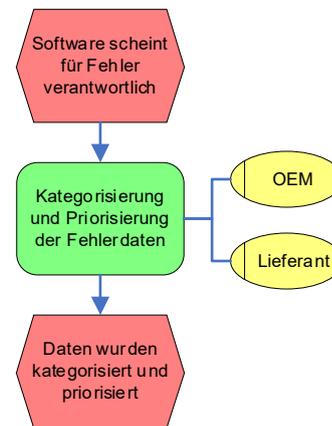


Abb. 7.7: Kategorisierung und Priorisierung der Fehlerdaten

Das Ziel der Vorprüfung besteht in der Bestätigung oder der Entlastung der zunächst verdächtigten Software. Hierbei besitzt der Lieferant eine Nachweispflicht gegenüber dem OEM. Kann der Zulieferer seine Beschuldigung nicht begründet zurückweisen, ist seine Software voraussichtlich für den Fehler verantwortlich. Anderenfalls sieht sich der OEM zur Identifikation der realen Ursache und der womöglich damit verbundenen Kontaktaufnahme zu einem anderen Lieferanten veranlasst.

Sobald feststeht, dass die Software des involvierten Lieferanten für den Fehler verantwortlich zu sein scheint, wird unter Umständen eine Kategorisierung und Priorisierung der Fehlerdaten nötig (vgl. Abb. 7.5.5). Dies ist insbesondere dann der Fall, wenn mehrere verschiedene Beschwerden zu der gleichen Software vorliegen. Die Forderung nach diesem neuen Prozessschritt wurde in Abschnitt 4.5 deutlich. An dieser Stelle wäre mit der Verwendung von User Stories und Tasks zudem erstmals eine Anwendung agiler Elemente möglich.

Im Anschluss an die womöglich nötige Kategorisierung und Priorisierung wird die eigentliche Fehlerbehebung vorgenommen. Wie bereits mehrfach angeführt, erscheint eine entsprechend freie Gestaltung derselben durch den Lieferanten als sinnvoll. Hierzu wird ein erhöhtes Vertrauen des OEM in seinen Zulieferer benötigt. Unabhängig davon, soll die vorliegende Arbeit auf den nachfolgenden Seiten eine mögliche Option der Umsetzung mittels der bereits benannten 4D+1-Methode aufzeigen.

### 7.5.2 Teilprozess 4D+1-Methode

In diesem Abschnitt wird der Entwurf eines neuen Problemlösungswerkzeuges präsentiert. Dieses beruht auf den Strukturen der bewährten 8D-Methode, die hinsichtlich der in Abschnitt 4.5 offenbarten Verbesserungspotenziale angepasst werden. Das Hauptaugenmerk liegt dabei auf einem beschleunigten Ablauf. Dem Lieferanten sollten bei der Problemlösung möglichst viele Freiheiten gewährt werden.

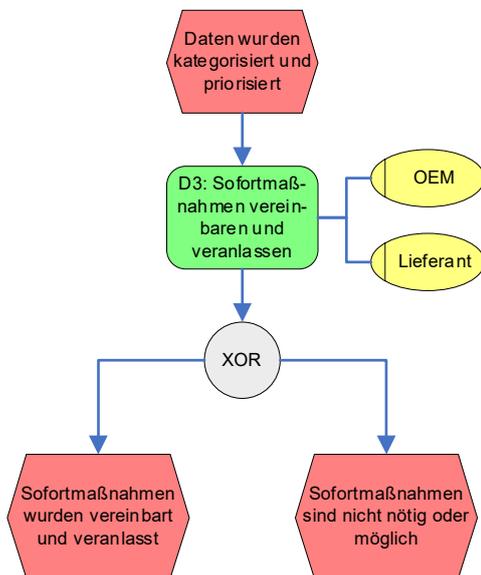


Abb. 7.8: D3: Sofortmaßnahmen

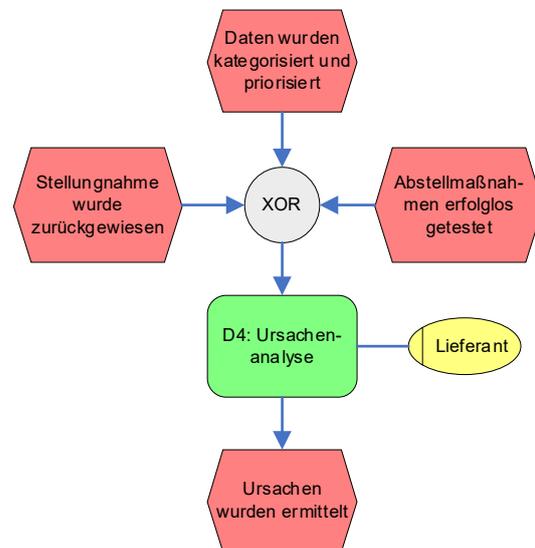


Abb. 7.9: D4: Ursachenanalyse

Zur Realisierung dieser Grundidee ist das bereits angeführte Umdenken der OEM als Voraussetzung aller weiteren Erfolgskriterien nötig. Als Hersteller der Software ist der Zulieferer am besten mit dieser und den möglichen Problemen vertraut. Der beschriebene Sachverhalt muss erkannt und berücksichtigt werden. Die nachfolgenden Schritte sollen demnach lediglich einen möglichen Ablauf der Problemlösung wiedergeben. Das vollständige Prozessschaubild ist dem Anhang 21 beigefügt.

Wie bereits in Abschnitt 4.5 erwähnt, finden sowohl die Disziplin eins (Problemlösungsteam) als auch die Disziplin zwei (Problembeschreibung) keine Anwendung in der angepassten Ablaufbeschreibung. Auf diese Weise wird eine erste Beschleunigung der Methodik erreicht. Der Prozess beginnt mit der Umsetzung von eventuell notwendigen Sofortmaßnahmen (D3). Die Entscheidung, ob und in welcher Weise diese getroffen werden sollen, liegt wiederum beim OEM und Lieferanten gemeinsam (S. Abb. 7.8). Sofortmaßnahmen werden dabei in der Regel durch das Aufspielen eines Backups oder einer vorherigen Softwareversion realisiert. Der

Entschluss zu diesem Schritt ist in hohem Maße von der Ausprägung des Problems und dem Aufwand für die Gegenmaßnahmen abhängig. OTA-Updates können letzteren wiederum entscheidend verringern.

Die Vorgehensweise während der gleichzeitig startenden Ursachenanalyse (D4) ist dem Lieferanten freigestellt (S. Abb. 7.9). Ihm werden keine Methoden, wie etwa das Ishikawa-Diagramm, vorgeschrieben. Für diesen und auch alle folgenden Schritte ist ausschließlich der Zulieferer zuständig. Die zugewiesene alleinige Verantwortung ist aus der bestehenden Ablaufbeschreibung bekannt. Der Start der Ursachenanalyse erfolgt entweder durch eine neue Anwendung der 4D+1-Methode, die Zurückweisung einer vorherigen Stellungnahme oder infolge fehlgeschlagener Tests.

Im Anschluss an die Ursachenermittlung erfolgt eine erneute Prüfung der Beanstandung (S. Abb. 7.10). Dieses Vorgehen wurde unverändert aus der bestehenden 8D-Methode übernommen, da es dem Lieferanten nach eingehender Problemuntersuchung die Möglichkeit zu einer erneuten Stellungnahme bietet.

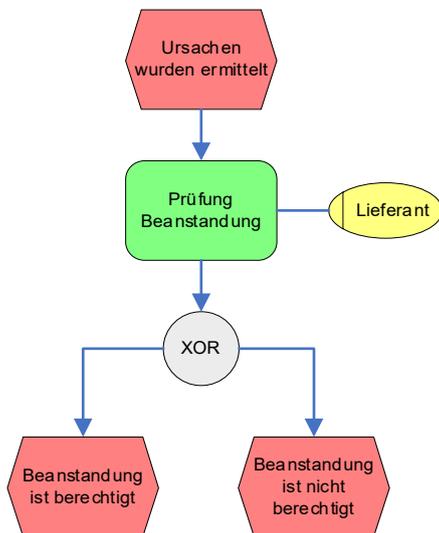


Abb. 7.10: Prüfung der Beanstandung

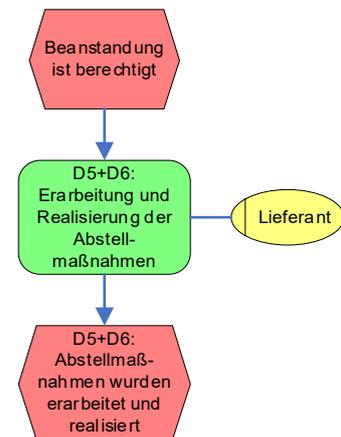


Abb. 7.11: D5+D6: Erarbeitung und Realisierung von Abstellmaßnahmen

Das Resultat der erneuten Prüfung besteht letztlich in einer Bestätigung oder Zurückweisung der Beanstandung. Im zweiten Fall wird nachfolgend auf den Teilprozess „Beanstandung ablehnen“ verwiesen (s. Abschnitt 7.5.4).

Im Falle einer berechtigten Beanstandung beginnt der Lieferant mit der Erarbeitung und Realisierung geeigneter Abstellmaßnahmen (D5+D6). Das entsprechende Vorgehen ist in diesem Fall nicht in einzelne Schritte wie bei der 8D-Methodik heruntergebrochen (S. Abb. 7.11). Eine derartig strikte Trennung erscheint bei Problemen der Software (digitale Dienste) nicht zielführend, wie durch das Industrie-Interview bestätigt wurde (vgl. Abschnitt 4.5.3). Das Pair Programming als Element der Agilität könnte innerhalb dieses Prozessschrittes zur Absicherung der Programmierung herangezogen werden, um eine Fehlerwiederholung weitestgehend auszuschließen.

Sobald die Maßnahmen in die Software integriert wurden, werden sie unverzüglich durch automatisierte Tests verifiziert (S. Abb. 7.12). Grundsätzlich ist diese Handlung bereits in der aktuellen 8D-Methode vorgesehen und dort innerhalb der sechsten Disziplin angesiedelt (Validierung der Abstellmaßnahmen). Durch die Ausprägung als separater Schritt wird die während des Experteninterviews hervorgehobene Bedeutung lediglich veranschaulicht. Zusätzlich ist auf diese Weise eine Darstellung der beiden potenziellen Ausgänge möglich. Entweder wird die Wirksamkeit bestätigt oder die Tests verlaufen erfolglos. Bei letzterem Sachverhalt muss die Beschwerde erneut der Ursachenanalyse (D4) zugeführt werden.

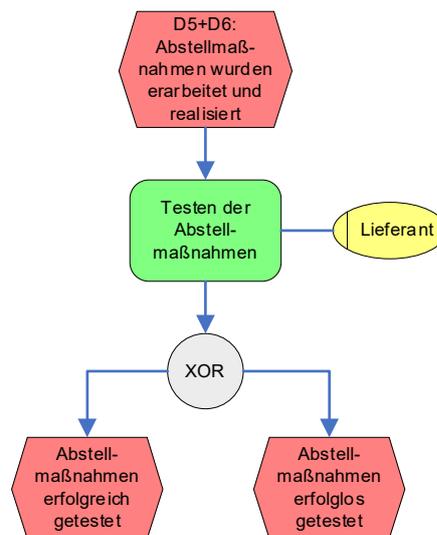


Abb. 7.12: Testen der Abstellmaßnahmen

Als direkte Folge eines positiven Testdurchlaufs gilt die 4D+1-Methode als erfolgreich abgeschlossen und der OEM kann mit der Bearbeitung der Reklamation fortfahren. Gesonderte Schritte zur Verhinderung einer Fehlerwiederholung (D7) oder zum Abschluss und Würdigung des Teamerfolgs (D8) werden eingeleitet. Dies hängt mit dem nicht vorhandenen Wert für die vorliegende Beschwerde zusammen. Der Fokus liegt stattdessen klar auf einer Beschleunigung des Prozesses. Die fraglos wichtigen Inhalte der Schritte werden jedoch nicht vernachlässigt, sondern an späterer Stelle ohnehin vom Entwicklerteam des Lieferanten durchlaufen.

### 7.5.3 Teilprozess Verifikation/Abschluss

Sobald die eigentliche Problemlösung abgeschlossen wurde, kommt es zur Prüfung der Stellungnahme des Zulieferers durch den OEM (S. Abb.). In diesem Zusammenhang soll dem OEM jedoch kein allumfassender 8D-Report zur Bewertung der zurückliegenden Maßnahmen überreicht werden. Vielmehr wird auf diese Weise ein (verbaler) Austausch zwischen dem OEM und Lieferanten angestrebt, der zum beiderseitigen Verständnis beiträgt. Die diesbezügliche Voraussetzung ist, dass der OEM die entsprechenden Kompetenzen zur Bewertung besitzt.

Falls der OEM mit der Stellungnahme des Zulieferers nicht zufrieden ist, wird diese zurückgewiesen und die Durchführung der Ursachenanalyse innerhalb der 4D+1 Methode beginnt erneut (vgl. Abb. 7.14 und Abb. 7.15). In dieser Vorgehensweise bestehen entscheidende Unterschiede zum bestehenden Prozess. Zum einen setzt die erforderliche Nacharbeit der erarbeiteten Lösung sehr viel später im Prozess ein als bisher (Ursachenanalyse anstatt Vorprüfung der Beanstandung). Zum anderen wird erforderliche Nacharbeit der erarbeiteten Lösung sehr viel später im Prozess ein als bisher (Ursachenanalyse anstatt Vorprüfung der Beanstandung). Zum anderen wird deutlich, dass die Wirksamkeitsprüfung im Anschluss an den Problemlösungsprozess auf ein notwendiges Minimum reduziert wurde. Auf diese Weise kann dem Ziel eines beschleunigten Gesamtprozesses entsprochen werden.

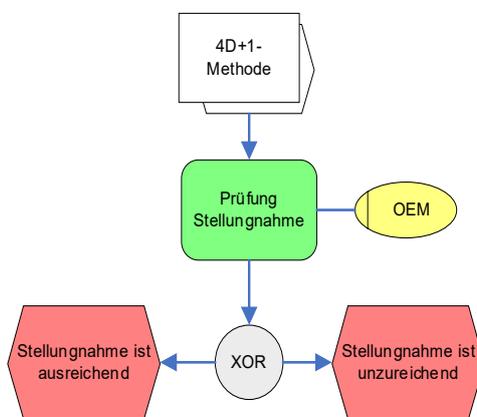


Abb. 7.13: Prüfung der Stellungnahme

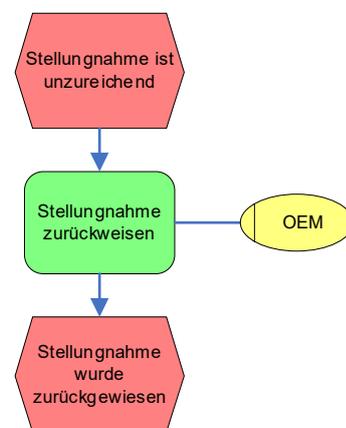


Abb. 7.14: Zurückweisung der Stellungnahme

Sofern die Stellungnahme des Lieferanten hingegen als ausreichend empfunden wird, sind zwei weitere Voraussetzungen zu einem Abschluss der Reklamation nötig. Die Beanstandung muss noch aktuell und damit nicht bereits storniert sein. Weiterhin ging und geht von ihr keine unmittelbare Beeinträchtigung der Sicherheit aus. Der OEM kann die Beanstandung in der Folge abschließen und den Lieferanten von seinen Pflichten entbinden. Er erkennt damit die für ihn zufriedenstellende Lösung der Problemstellung an. Der Zulieferer übergibt dem OEM alle benötigten Arbeitsergebnisse. Diese bestehen zum Beispiel in Form einer weiterentwickelten Software.

Im Anschluss an die erfolgreiche Reklamationsbearbeitung obliegt die Entscheidung über das weitere Vorgehen dem OEM. Idealerweise findet eine sofortige Aktualisierung der betreffenden Fahrzeugsoftware mittels eines OTA-Updates statt. Die dafür notwendigen technischen sowie rechtlichen Voraussetzungen sollten in den nächsten Jahren geschaffen werden. Nur auf diese Weise ist eine ganzheitliche, schnelle Reklamationsbearbeitung vom Aufkommen bis zur Behebung eines Fehlers realisierbar.

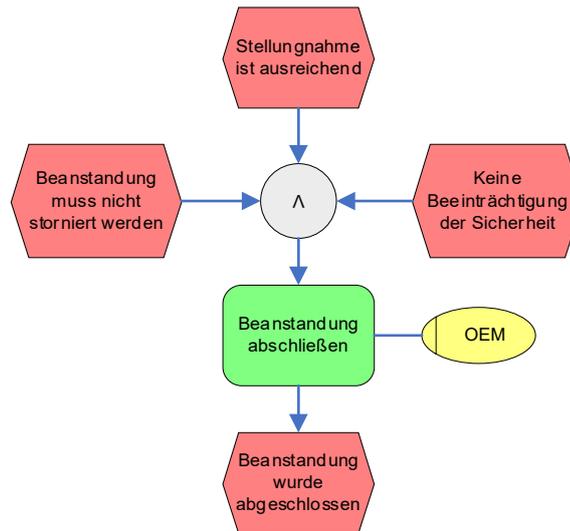


Abb. 7.15: Abschluss der Beanstandung

Während der regelmäßig stattfindenden Retrospektive des Software-Lieferanten werden im Anschluss an die Reklamationsbearbeitung zudem die Inhalte der 8D-Schritte sieben und acht berücksichtigt. Die Mitarbeiter tauschen sich über die vergangenen Aufgaben aus und projizieren gewonnene Erkenntnisse auf bestehende Prozesse. Dem Gedanken der kontinuierlichen Verbesserung wird somit entsprochen.

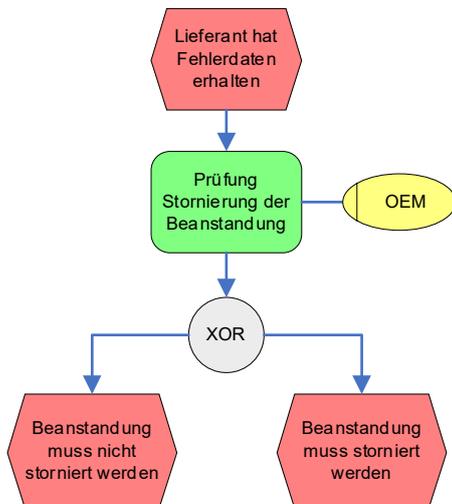


Abb. 7.16: Prüfung auf Stornierung der Beanstandung

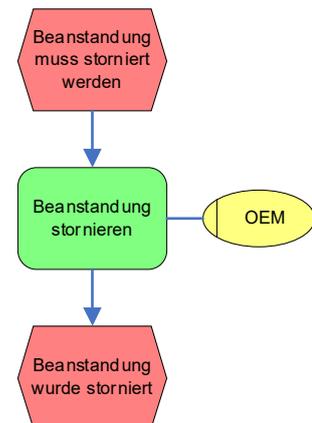


Abb. 7.17: Stornierung der Beanstandung

Ein weiterer möglicher Abschluss der Beanstandung besteht in der vorzeitigen Stornierung durch den OEM. Dieses Vorgehen hat sich gegenüber dem bestehenden Prozess nicht geändert (S. Abb. 7.16 und Abb. 7.17).

Der OEM ist in diesem Zusammenhang zu einer dauerhaften Prüfung einer möglichen Stornierung angehalten. Diese könnte beispielsweise in einer fehlerhaften Datenübermittlung durch den Kunden begründet sein. Als direkte Folge müsste der OEM den Lieferanten, unabhängig vom derzeitigen Arbeitsfortschritt, informieren und ihn mit sofortiger Wirkung von seinen Pflichten entbinden.

#### **7.5.4 Teilprozess Beanstandung ablehnen**

Dieser Teilprozess besitzt eine uneingeschränkte Relevanz für den weiterentwickelten Reklamationsprozess. Da die entsprechenden Schritte nahezu unverändert übernommen werden, wird aus Platzgründen auf eine Darstellung verzichtet. Die einzige Veränderung besteht in der Reduzierung der möglichen Prozesseingänge. Da die Vorprüfung der Fehlerdaten im weiterentwickelten Reklamationsprozess von OEM und Lieferant gemeinsam vorgenommen wird, ist die begründete Ablehnung der Beanstandung durch den Zulieferer nicht möglich bzw. nötig. Sollte sich bei der gemeinsamen Voruntersuchung der Software der anfängliche Verdacht also nicht bestätigen, findet eine direkte Stornierung der Beanstandung durch den OEM statt (s. Anhang 11).

#### **7.5.5 Teilprozess Produktrückruf**

Während der Erarbeitung des weiterentwickelten Reklamationsprozesses erschien die Ergänzung des Gesamtablaufs um den Teilprozess Produktrückruf sinnvoll. Dieser begründet sich in der zumeist erst späten Fehlerentdeckung beim Kunden. Damit ist die fehlerhafte Anwendung bereits im Feld aktiv. Dies erfordert besondere Konsequenzen, die in der Automobilindustrie bei dem Vorliegen eines Gefährdungspotenzials mittels einer Rückrufaktion umgesetzt werden. Die diesbezüglichen Schritte sollen nachfolgend nicht weiter skizziert und thematisiert werden, da viele Faktoren einen Einfluss besitzen und die Inhalte mühelos eine separate Forschungsarbeit füllen würden. Im Falle einer direkten Beeinträchtigung der Sicherheit der Fahrzeuginsassen oder umstehender Personen muss der Fehler jedoch schnellstmöglich behoben werden, um die Sicherheit wiederherzustellen. Das aktuelle Vorgehen sieht das Aufspielen einer neuen Software in den Werkstätten vor. Dieser Prozess ist jedoch sehr langwierig und mit einem hohen Aufwand für alle Beteiligten verbunden. Eine effektivere Lösung stellen OTA-Updates dar. Dafür müssen zunächst jedoch die nötigen, vor allem gesetzlichen, Bedingungen geschaffen werden. Die technischen Voraussetzungen sind durch die digitalen Dienste in der Regel bereits gegeben.

Das zurückliegende Kapitel beinhaltete die Vorstellung des weiterentwickelten Reklamationsprozesses. In diesem Zusammenhang wurden zunächst die diesbezüglichen Ziele aufgegriffen. Im Anschluss erfolgten die Darstellung und Beschreibung der einzelnen Prozessschritte unter Berücksichtigung der zuvor formulierten Erfolgskriterien. Die Prozesseinteilung wurde nach den bekannten Grundsätzen von VDA vorgenommen. Der Produktrückruf wurde als sinnvoller fünfter Teilprozess integriert. Dieses Vorgehen erschien im Hinblick auf die zumeist späte Fehlerentdeckung im Feld als nötig. Im vorangegangenen Abschnitt erfolgte ferner eine Zusammenfassung der identifizierten Herausforderungen und Risiken, die bei der Einführung und Umsetzung des angestrebten Reklamationsprozesses auftreten können.

Abschließend bleibt die Frage zu beantworten, ob und inwieweit die am Beginn des Kapitels formulierten Ziele durch den Prozess erreicht werden können. Durch das Entfernen bzw. Minimieren einiger bisheriger Prozessschritte (z. B.

Wirksamkeitsnachweis, Elemente der 8D-Methodik) ist zweifellos eine Beschleunigung des Gesamtprozesses zu erwarten. Gleichzeitig ist mit der Kategorisierung sowie Priorisierung der Fehlerdaten lediglich ein Prozessschritt hinzugekommen. Der dafür nötige zusätzliche Zeitaufwand wird sehr wahrscheinlich jedoch nicht stark negativ ins Gewicht fallen. Ferner wurde eine nachhaltige Fehlerbeseitigung als ein zentrales Ziel formuliert. Dieses kann vor allem durch das Pair Programming als ein Element der Agilität im Rahmen der Maßnahmenumsetzung erreicht werden. Weiterhin wurde eine verstärkte Berücksichtigung des indirekten Beschwerdemanagements angestrebt. Die 8D-Methode bildet dieses im aktuellen Prozess bereits sehr gut ab. In diesem Zusammenhang erscheint die nicht vorhandene Berücksichtigung der Disziplinen sieben und acht in der 4D+1-Methodik zunächst als sehr fragwürdig. Es wurde jedoch verdeutlicht, dass die Schritte an anderer Stelle und ohne Zwang des Reklamationsmanagements ohnehin durchlaufen werden. Die Inhalte bleiben folglich erhalten.

Die zu Beginn der Arbeit formulierten Anforderungen der digitalen Dienste finden im weiterentwickelten Fehleranalyseprozess ebenfalls eine umfassende Berücksichtigung. Der stetig zunehmenden Anzahl von Softwarefehlern wird durch eine entsprechende Kategorisierung und Priorisierung begegnet. Der späten Fehlerentdeckung beim Kunden wird durch die angestrebte automatische Datensendung und dem zusätzlichen Teilprozess des Produktrückrufs entsprochen. Die letztlich geforderte Weiterentwicklung des Prozesses hinsichtlich angepasster Abläufe unter Berücksichtigung der technischen Möglichkeiten ist durch die gesamte Ausprägung des Prozesses gegeben. Auf diese Weise kann abschließend von einer Wiederherstellung der zuvor gesunkenen Kundenzufriedenheit ausgegangen werden.

### **7.5.6 Ergebnisse und Zusammenfassung**

Das Ziel des Kapitels 7 bestand in der Weiterentwicklung des bestehenden Reklamationsprozesses in der Automobilindustrie hinsichtlich der durch die digitalen Dienste aufkommenden Herausforderungen. Diese bestehen in stetig wachsenden Softwareumfängen und -komplexitäten, der zumeist späten Fehlerentdeckung im Feld sowie in der Forderung nach technologisch angepassten Abläufen.

Zur Erreichung dieser Zielstellung erfolgte zunächst die Präsentation der notwendigen theoretischen Grundlagen. In diesem Zusammenhang wurden die digitalen Dienste, das Lieferantenmanagement, das allgemeine Beschwerdemanagement sowie der standardisierte Reklamationsprozess inklusive der 8D-Methode thematisiert. Dabei fiel auf, dass die bestehende Literatur zum Beschwerdemanagement sehr zahlreich, jedoch nahezu ausschließlich auf B2C-Beziehungen ausgerichtet ist. Das im Sinne des Reklamationsprozesses in der Automobilindustrie vorliegende B2B-Verhältnis findet nahezu keine Beachtung. Dieser Umstand resultierte in einer maßgeblich erschwerten Quellenrecherche, die sich bis in die diesbezüglichen Abläufe innerhalb der Software-Industrie zog. Aus diesem Grund war es lediglich mittels eines Fachinterviews möglich, einen kurzen Einblick in den Umgang mit Reklamationen in dieser Branche zu erhalten.

Ein neu entwickelter Use-Case führte die bisher getrennt betrachteten Elemente der digitalen Dienste und des standardisierten Reklamationsprozesses in der Automotive-Branche zusammen. Ein möglicher Fehler innerhalb der automatischen Parkplatznavigation wurde vollumfänglich auf die bestehende Ablaufbeschreibung angewandt. In diesem Zusammenhang offenbarten sich eine Reihe möglicher Konfliktpotenziale. Der anschließende Teil griff diese auf und stellte ihnen gleichzeitig mögliche Lösungsansätze zur Seite. Insgesamt konnten auf diese Weise acht zentrale Optimierungsmöglichkeiten des bestehenden Prozesses identifiziert werden. Diese reichten von zusätzlich benötigten Prozessschritten (Einführung einer Beschwerdestimulierung), über das Entfernen bestehender Elemente (Abschaffung der doppelten Wirksamkeitsprüfung) bis hin zur unter Umständen zielführenden Abschaffung der 8D-Methode. Im Zuge dieser Ausarbeitungen wurden nicht nur die Herausforderungen, sondern insbesondere die Möglichkeiten der Connectivity deutlich. Sie bestehen vor allem in der Nutzung von OTA-Updates und der automatischen Datensendung.

Im letzten Kapitel 7.5 erfolgte die abschließende Darstellung des weiterentwickelten Reklamationsprozesses. Dieser orientiert sich in seinen Grundzügen nach wie vor an den Teilprozessen des VDA. Zudem verwendet er aus Gründen der Vergleichbarkeit die identische Modellierungssprache. Die in Bezug auf die vorherigen Abschnitte umgesetzten Veränderungen sind dennoch offensichtlich. So wurde beispielsweise die 8D-Methodik um vier Prozessschritte verkürzt, um eine Beschleunigung des Gesamtablaufs zu erreichen. In diesem Zusammenhang wurde jedoch darauf geachtet, dass wichtige Elemente nicht einfach entfernt, sondern an anderer Stelle erneut integriert wurden oder womöglich bereits integriert waren. Dies war bei den Disziplinen 7 und 8 der Fall. Als Ergebnis konnte auf diese Weise letztlich ein auf das Wesentliche reduzierter Reklamationsprozess präsentiert werden, der an entscheidenden Stellen die Möglichkeiten der digitalen Dienste und Agilität zu seinem Vorteil nutzt.

Zum Abschluss wurden zusätzlich die Herausforderungen und Risiken der weiterentwickelten Ablaufbeschreibung diskutiert. Zentrale Punkte bestanden dabei in der nötigen Öffnung der OEM, den Gefahren der OTA-Updates sowie der erhöhten Datenflut. Der vorgestellte weiterentwickelte Reklamationsprozess lässt folglich keine sofortige Anwendung zu. Er bietet jedoch Ansätze, welche seine Möglichkeiten in einem hohen Maße erweitern. Diese sollten zwingend genutzt werden, weshalb eine zeitnahe Schaffung der nötigen Rahmenbedingungen als unausweichlich erscheint.

## **7.6 Herausforderungen und Risiken**

Die in den vorherigen Ausführungen präsentierte, weiterentwickelte Fehlerklassifizierung, das Fehlermanagement sowie der Reklamationsprozess in der Automobilindustrie geht zweifellos mit einer Reihe von Herausforderungen und Risiken einher. Die besonderen Voraussetzungen und möglichen Fehlentwicklungen wurden im Laufe der Arbeit bereits einige Male hervorgehoben und sollen an dieser Stelle lediglich zusammengefasst wiedergegeben werden.

Die aktuell im Automobilbau vorherrschende Kultur ist über Jahrzehnte gewachsen und fand ihre Bestätigung im stets anhaltenden wirtschaftlichen Erfolg der Branche. Sie ist jedoch von veralteten Wertesystemen und einer klar strukturierten Rangfolge geprägt.<sup>[146]</sup> Innerhalb dieser Umgebungsbedingungen ist eine Implementierung des weiterentwickelten Fehleranalyseprozesses nicht bzw. nur schwer möglich. Aus diesem Grund bildet das Umdenken der OEM die notwendige Grundvoraussetzung. Die nachfolgend angestrebte Kultur ist „[...] von Neugierde, Änderungsbereitschaft und flachen Hierarchien [...]“ geprägt. In ihr stehen „Geschwindigkeit und Agilität [...] über formalen, trägen Prozessabläufen.“<sup>[146]</sup> Der weiterentwickelte Reklamationsprozess stellt in dieser Umgebung nur eine von vielen möglichen und gleichzeitig nötigen Änderungen dar.

Eine weitere Herausforderung besteht in den stetig zunehmenden Verzweigungen innerhalb der gesamten Lieferkette, die auf immer komplexer werdenden Fahrzeugkomponenten beruhen. Die Funktionalitäten verteilen sich auf immer mehr Steuergeräte und Softwarebestandteile, wodurch die Fehlersuche um ein Vielfältiges erschwert wird.<sup>[147]</sup> Die Ursachenfindung für eine Problemstellung und die darauf fußende, eindeutige Zuordnung zu einem Lieferanten wird in der Folge immer aufwändiger. Diese starke Abhängigkeit der Teilsysteme voneinander sowie die daraus resultierenden Folgen wurden bereits in der Einleitung thematisiert und während der weiteren Ausarbeitungen durch ein Gespräch mit einem Experten aus der Automobilindustrie bestätigt.

Neben den benannten Herausforderungen resultiert der angestrebte Prozess in möglichen Risiken. Zum einen bestehen diese in der Voraussetzung technischer Gegebenheiten, welche nicht jedes Fahrzeug bieten kann. Beispielsweise beruhen viele Prozessschritte auf dem Vorhandensein einer stetigen Internetverbindung des Fahrzeugs (automatische Sendung der Fehlerdaten). Diese Möglichkeit ist zumeist nur bei moderneren Automobilen gegeben. Ist der Prozess folglich nur auf eine begrenzte Anzahl von Fahrzeugen anwendbar? In diesem Zusammenhang muss die folgende Überlegung berücksichtigt werden: Bei Fahrzeugen mit einer stetigen Internetverbindung handelt es sich um eben jene Automobile, die den weiterentwickelten Fehleranalyse- bzw. Reklamationsprozess aufgrund der installierten digitalen Dienste erfordern. Ältere und mit weniger Funktionen ausgestattete Fahrzeuge benötigen dies und folglich auch die Internetverbindung nicht.

Die damit in enger Verbindung stehenden, angestrebten OTA-Updates bergen weitere Risiken. Diese beziehen sich vor allem auf die erfolgreiche Installation sowie Integration in das sensible Gesamtsystem. Eine genaue Vorhersage des diesbezüglichen Verhaltens ist nur schwer möglich, dennoch muss die Verbreitung dieser Technologie alternativlos gefördert werden. Zu diesem Zweck ist insbesondere die Schaffung rechtlicher Grundlagen anzustreben, denn technisch sind die meisten Funktionen bereits jetzt oder in naher Zukunft abbildbar. Wie zuvor beschrieben bieten OTA-Updates sowohl für das Reklamationsmanagement als auch für Rückrufaktionen enorme Vorteile. Letztere sind bei einer vollumfänglichen Anwendung genau genommen nicht länger

nötig, da die Fahrzeuge zum Aufspielen eines Updates nicht länger in die Werkstätten zitiert werden müssten.

Viele der angeschnittenen Themen münden nicht zuletzt in Aspekten des Datenschutzes und der IT-Sicherheit. Die große Zahl der durch digitale Dienste produzierten Daten führt dabei zu einem erhöhten Risiko und der Fahrzeughersteller muss für den Schutz der Daten vor nicht autorisierten Zugriffen Sorge tragen. Mögliche Gefährdungspotenziale wurden in dieser Arbeit angeschnitten. Als notwendige Voraussetzung für die angestrebte automatische Datensendung muss der Fahrzeuginhaber dem Vorgehen zustimmen. In der Unterlassung dieser Einwilligung besteht ein hohes Risiko für den Erfolg des weiterentwickelten Reklamationsprozesses, welches mit der zunehmenden Sensibilisierung der Bevölkerung für den Wert ihrer Daten wächst. Die Lösung sollte in einer transparenten und beiderseitig lohnenden Vertragsgestaltung liegen, welche zur automatischen Weitergabe der Informationen an den OEM animiert. Ein großer Einfluss liegt zudem beim Staat, da er zur Schaffung der notwendigen Rahmenbedingungen durch eine geeignete Gesetzgebung angehalten ist.

## **7.7 Kritische Reflexion**

Allgemein lässt sich festhalten, dass die Ablaufbeschreibung und die damit verbundenen Gedanken zumeist auf theoretischer Basis entwickelt wurden. Entsprechende Versuche einer Validierung fanden folglich nicht statt. Zudem ließen sich viele Expertenmeinungen aus den Industrien zu entsprechenden Sachverhalten gewinnen und in den Prozess integrieren. Innerhalb dieses Prozessschrittes wird nach wie vor ein großer Handlungs- und Forschungsbedarf gesehen, der eventuell von einer nachfolgenden Forschungsarbeit vertiefend aufgegriffen werden könnte.

Einige Neuerungen lassen zudem auf einen erhöhten Personalbedarf schließen und mit diesem ist bei der Umsetzung des weiterentwickelten Prozesses fraglos zu rechnen. Durch die frühzeitig angestrebte und vertiefte Zusammenarbeit von Lieferanten und OEM sehen sich die Unternehmen mit einer erhöhten Personalbereitstellung konfrontiert. Letztlich ist darin jedoch eine notwendige Entwicklung zu sehen, die die wichtige Rolle des Fehleranalyseprozesses unterstreicht.

## 8 Zusammenfassung und Zukunftsausblick

Wegen der Digitalisierung ist eine zentrale Herausforderung der Automobilindustrie das intelligente Verknüpfen von Aspekten aus traditionellen und digitalen Wertschöpfungsstufen. Aspekte sind die möglichen Veränderungspotentiale durch die fortschreitende Digitalisierung. Die klassischen Automobilhersteller fokussieren sich heute neben den produktbezogenen Kernkompetenzen eines Automobils auf digitale Dienstleistungen.

Die überwiegend statischen Konzernstrukturen können die Automobilhersteller an einer erfolgreichen Bewältigung digitaler Herausforderungen hindern. Das Automobil als traditionelles Kernprodukt dieser Industrie weist mit seinen komplexen Produktmerkmalen eine Asynchronität im Entwicklungszyklus zu den digitalen Innovationen auf. Ein agiler Entwicklungsprozess, das Einbeziehen von Kundenerwartungen in die Entwicklungsphase und die Umsetzung hoher Qualitätsanforderungen für eine fehlerfreie Nutzungsphase zählen mitunter zu den künftigen Erfolgsfaktoren.<sup>[12]</sup> Die digitalen Innovationen bestehen aus für die digitalen Geschäftsmodelle relevanten Fahrzeugbauteilen und einer mobilen Softwareanwendung. Der Produktlebenszyklus teilt sich allgemein in eine Entwicklungs- und Kundennutzungsphase. Die Phasen der Fahrzeugbauteile unterscheiden sich sowohl inhaltlich als auch zeitlich von denen der Softwareanwendung.

### Zentrale Erfolgskriterien

Ausgehend von den bisherigen Erkenntnissen existieren drei zentrale Erfolgskriterien, welche über das Erreichen oder Verfehlen des Zieles entscheiden (s. Abb. 8.1).

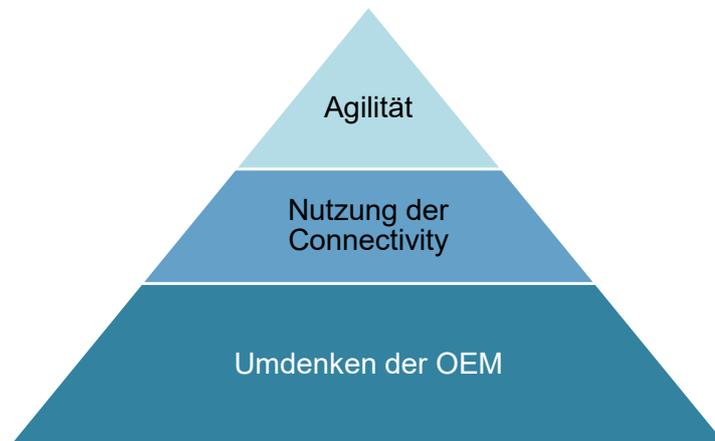


Abbildung 8.1: Erfolgskriterien des weiterentwickelten Reklamationsprozesses (eigene Darstellung)

#### 1. Umdenken der OEM

Die diesbezügliche Grundvoraussetzung besteht in einem Umdenken innerhalb der Organisation. Die OEM müssen eine Öffnung für Ideen, welche ihren Ursprung außerhalb der Automobilindustrie haben, zulassen. Auf diese Weise soll das bereits aufgegriffene Not-Invented-Here-Syndrom überwunden werden. Die Erfolgs-

wahrscheinlichkeit der Zielerreichung ist in hohem Maße von dieser Entwicklung abhängig. Ferner darf sich der Gedanke der kontinuierlichen Verbesserung in den Unternehmen nicht allein auf die Produkte beschränken. Er muss auch auf das Fehleranalyseverfahren und andere Prozesse übertragen werden, um nicht zuletzt den Forderungen der Normung zu entsprechen.

Zusätzlich ist eine Vertiefung der Zusammenarbeit zwischen OEM, Lieferanten und neuer Player anzustreben. Das Ziel sollte in einer gestärkten Kommunikation und aufeinander abgestimmten Strukturen bestehen. Ein geeignetes Management der gemeinsamen Arbeit kann in diesem Zusammenhang einzig und allein durch eine leistungsstarke IT-Lösung erfolgen. Ein händisches Management durch das Versenden von E-Mails und Excel-Dokumenten wird den zunehmenden Anforderungen nicht länger gerecht. Die genannten Gründe führen dazu, dass die Organisation die notwendige Basis für die zwei weiteren Erfolgskriterien bildet.

## 2. Nutzung der Connectivity

Das nächste Hauptaugenmerk liegt auf einer Nutzung der Chancen, die mit den digitalen Diensten einhergehen. Folglich bietet diese nicht ausschließlich Herausforderungen, sondern gleichzeitig konkrete Entwicklungspotenziale für bestehende als auch neue Prozesse. Die Vielzahl der bereits generierten bzw. zu generierenden Daten soll in diesem Zusammenhang aktiv für den weiterentwickelten Reklamationsprozess genutzt werden.

## 3. Agilität

Die dargestellte Pyramide der zentralen Erfolgskriterien resultiert schließlich in einer angestrebten Umsetzung von Elementen der Agilität, die in dieser Arbeit schon vorgestellt wurden. Dieser Schritt erscheint langfristig sinnvoll, da die Entwicklung der digitalen Dienste nach eben jenen Prinzipien vollzogen wird. Eine Isolation der Fehleranalysebearbeitung von dieser für den Lieferanten gewohnten Umgebung erscheint unter keinen Umständen zielführend. Zusätzlich kann auf diese Weise einer Vielzahl an offenbarten Problemstellungen begegnet werden. Nicht zuletzt wurden mit den Elementen der Agilität in der Vergangenheit ähnliche Herausforderungen bewältigt wie sie heute in der Automobilindustrie auftreten.

## **Zukunftsausblick**

Aktuelle gesamtwirtschaftliche Entwicklungen wie die zunehmende Globalisierung, der wachsende Kostendruck, die stetig steigenden Risiken sowie die immer kürzer werdenden Produktlebenszyklen werden zukünftig eine weitere Verstärkung erfahren. Von diesen Herausforderungen ist auch die Automobilindustrie nicht ausgenommen. Wie wird sie darauf reagieren und wie wird sie sich anpassen müssen? Als Antwort auf diese Frage soll abschließend ein Ausblick aufkommende Entwicklungen und Forschungsbedarfe gegeben werden.

In diesem Zusammenhang werden zunächst die unmittelbar notwendigen Veränderungen und Entwicklungen verdeutlicht, ehe ein spekulativer Einblick in die Automotive-Branche bis hinein in die 2030er Jahre gegeben wird.

Die Inhalte der Arbeit haben deutlich gezeigt, dass eine Kulturveränderung innerhalb der OEM unabdingbar sein wird. Die Automobilindustrie muss sich für Neuerungen aus anderen Branchen öffnen und passende Ideen adaptieren. Dieser Umstand ist insbesondere bei dem Zusammenspiel mit Software-Lieferanten sowie Startups unerlässlich. In einer repräsentativen Umfrage bestätigten die Experten diese Notwendigkeit. Zusätzlich besteht in diesem Zusammenhang unter Umständen der Bedarf einer erhöhten Fehlertoleranz, in deren Folge Irrtümer als inhärenter Teil des Lernprozesses verstanden und entsprechend nicht verurteilt werden.

Die Herausforderung wird darin liegen, eine geeignete „[...] Balance zwischen Freiheit, Kreativität und Fehlertoleranz auf der einen und Qualitäts- und Sicherheitsanforderungen auf der anderen Seite herzustellen.“ Durch eine erhöhte Fehlertoleranz würde zudem eine kürzere Time-to-Market realisiert werden können, da Entwicklungen nicht bis ins letzte Detail erprobt und somit dem Endverbraucher früher zur Verfügung gestellt werden. Die Umfrage zeigt jedoch auch, dass dies aktuell noch den Grundsätzen dieser auf höchste Qualität bedachten Industrie widerstrebt. Knapp zwei Drittel der Befragten gehen nicht von einer entsprechenden Entwicklung aus.

Bezogen auf die Entwicklung digitaler Dienste wird zudem die Anwendung eines Open-Source-Modells in Erwägung gezogen. In diesem finden Wettkampf und Kooperation parallel statt. Die Begründung besteht in den zahlreichen Herausforderungen bei der Entwicklung, denen kein Marktteilnehmer allein begegnen kann. Womöglich wird sich die Ausprägung dieses Geschäftsmodells zwangsläufig vollziehen, damit eine Behauptung gegen die zahlreichen Konkurrenten auf dem Markt der digitalen Dienste möglich ist. Konzerne wie Apple oder Google sind nicht nur mit dem nötigen Wissen, sondern auch mit dem nötigen Kapital ausgestattet, um den OEM ihre vorherrschende Rolle auf dem Automobilmarkt nachhaltig streitig zu machen. Eine Anwendung des Open-Source-Gedankens auf den gesamten Fehleranalyseprozess erscheint gleichermaßen denkbar. Auf diese Weise könnten Hersteller von den Fehlern anderer OEM oder Lieferanten lernen und somit einen Vorteil gegenüber den New Playern erzielen.

## 9 Literatur

- [1] Daimler AG (2017): [www.daimler.com](http://www.daimler.com) (Zugriff: 10.04.2018)
- [2] <https://www.bitkom.org/Presse/Presseinformation/Automobilbranche-Jeder-Zweite-macht-einen-Bogen-um-Start-ups.html>. 17.06.2018
- [3] Bennett, T. L.; Wennberg, P. W. (2005): Eliminating embedded software defects prior to integration test, in: Crosstalk, Journal of Defence Software Engineering, S. 13-18
- [4] <https://www.zukunftsinstitut.de/dossier/megatrend-konnektivitaet/> (Zuletzt: 05.05.2017)
- [5] BMW AG (2016): Geschäftsbericht 2016 - Eine neue Ära, Online: [https://www.bmwgroup.com/content/dam/bmw-group-websites/bmwgroup\\_com/ir/downloads/de/2017/GB/2016-BMW-Group-Geschaeftsbericht.pdf](https://www.bmwgroup.com/content/dam/bmw-group-websites/bmwgroup_com/ir/downloads/de/2017/GB/2016-BMW-Group-Geschaeftsbericht.pdf) (Zugriff: 18.07.2019)
- [6] Dust, Robert (2018a): Einführung in die Automobilindustrie, Unveröffentlichtes Skript zur Vorlesung im Wintersemester 18/19
- [7] Daimler AG 2017: Daimler Geschäftsbericht 2017, Online: <https://www.daimler.com/dokumente/investoren/berichte/geschaeftsberichte/daimler/daimler-ir-geschaeftsbericht-2017.pdf> (Zugriff: 18.07.2019)
- [8] Volkswagen, <https://www.volkswagenag.com/en/news/stories/2019/06/volkswagen-is-developing-more-of-its-own-software.html>; 01.07.2019
- [9] Daimler AG (2016): Geschäftsbericht 2016, Online: <https://www.daimler.com/dokumente/investoren/berichte/geschaeftsberichte/daimler/daimler-ir-geschaeftsbericht-2016.pdf> (Zugriff: 18.07.2019)
- [10] Umwelt Bundesamt 2017: Gesundheitsrisiken durch Feinstaub, Online: <https://www.umweltbundesamt.de/daten/umwelt-gesundheit/gesundheitsrisiken-durch-feinstaub#textpart-1> (Zuletzt: 18.07.2019)
- [11] Umwelt Bundesamt 2015: Emissionsstandards, Online: <https://www.umweltbundesamt.de/themen/verkehr-laerm/emissionsstandards> (Zuletzt: 18.07.2019)
- [12] Strelow, M., Wussmann, 2016: Digitalisierung in der Automobilindustrie. Ismaning: Iskander Business Partner, Online erhältlich unter: <http://i-b-partner.com/wpcontent/uploads/2016/08/2016-09-06-Iskander-RZ-Whitepaper-Digitalisierung-in-der-Automobilindustrie-DIGITAL.pdf> (Zugriff: 24.05.2018)
- [13] <https://www.next-mobility.news/connected-car-features-ungenutzt-unbekannt-und-zu-teuer-a-685393/>, 17.06.2018
- [14] Strelow, M., Wussmann, 2016: Digitalisierung in der Automobilindustrie. Ismaning
- [15] Automobilwoche Beilage, 2015 (Zugriff: 04.05.2018)
- [16] Cars online 2017, Beyond the car, Capgemini Consulting
- [17] [http://www.handelsblatt.com/unternehmen/management/digitaletransformation /neue-kooperation-alibaba-soll-autos-von-daimler-und-audi-intelligenter-](http://www.handelsblatt.com/unternehmen/management/digitaletransformation/neue-kooperation-alibaba-soll-autos-von-daimler-und-audi-intelligenter-)

- machen/21202890.html?ticket=ST-1696308-QFr0MEzJThS0dBchXZJM-ap2  
(06.05.2018)
- [18] <https://entwickler.de/online/development/gartner-trends-it-2018-579815076.html>,  
06.05.2018
- [19] DIN EN ISO 9000 Qualitätsmanagementsysteme (2015)
- [20] Dust, R., Lu, Y., Dietze, B. (2018): Entwicklung eines agilen Fehlermanagementprozesses für Apps in der Automobilindustrie, Forschungsstudie Fachgebiet Qualitätsstrategie und Qualitätskompetenz, TU Berlin.
- [21] Sommerhoff, B., Brecht, A., & Fiegler, M. (2018): DGQ-Deutsche Gesellschaft für Qualität. Abgerufen am 9. Dezember 2018 von [https://www.dgq.de/z/schnupper/WP\\_QS.pdf](https://www.dgq.de/z/schnupper/WP_QS.pdf)
- [22] Brüggemann, H.; Bremer, P. (2015): Grundlagen Qualitätsmanagement - Von den Werkzeugen über Methoden zum TQM, 2. Aufl., Wiesbaden: Springer Vieweg
- [23] Hohler, B. (2014): Qualitätsmanagement bei der Software-Entwicklung, in: Pfeifer, T.; Schmitt, R.; Masing, W. (Hg.), Masing Handbuch Qualitätsmanagement, 6. überarb. Aufl.: Carl-Hanser Verlag, S. 829-919
- [24] ISO/IEC 25000:2014-03, Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE)
- [25] Schatten, A., Biffel, S., Demolsky, M., Gostischa-Franta, E., Östreicher, T. & Winkler, D. (2010): Best Practice Software-Engineering. Eine praxiserprobte Zusammenstellung von komponentenorientierten Konzepten, Methoden und Werkzeugen. Heidelberg, Neckar: Spektrum
- [26] Brandes, C. & Heller, M. (2016): Qualitätsmanagement in agilen IT-Projekten – quo vadis? In: HMD Praxis der Wirtschaftsinformatik, 53. Jg., Nr. 2, S. 169–184
- [27] Broy, M.; Kuhrmann, M. (2013): Projektorganisation und Management im Software Engineering, Berlin Heidelberg: Springer Vieweg
- [28] Hoffmann, W. D. (2013): Software-Qualität. Karlsruhe: Springer Vieweg
- [29] Morenzin, J., & Vanamali, B. (06 2009): Angekommen in der Entwicklung - Automotive Spice durchdringt Steuergeräteentwicklung. QZ Qualität und Zuverlässigkeit, S. 18–23
- [30] Freidank, Maximilian Philip (2017): Bewertung von erfolgversprechenden Ansätzen intelligenter, vernetzter Mobilität, Studiengang Wirtschaftsingenieurwesen an der Fakultät Betriebswirtschaft der Hochschule Esslingen, 11.08.2017
- [31] Schmidt, T. & Paetzold, K. (2016): Design for X - Beiträge zum 27. DfX-Symposium. Hamburg: Tutech
- [32] Wieczorrek, H. & Mertens, P. (2008): Management von IT-Projekten. Berlin, Heidelberg: Springer
- [33] ISO 25010, Norm für Qualitätskriterien von Software, IT-Systemen und Software-Engineering
- [34] Vigenschow, U. (2010): *Testen von Software und Embedded Systems*. Professionelles Vorgehen mit modellbasierten und objektorientierten Ansätzen, Heidelberg: dpunkt Verlag

- [35] Hoffmann, D. (2008): Software-Qualität. Berlin, Heidelberg: Springer
- [36] Becker, S., Plasil, F. & Reussner, R. (2008): Quality of software architectures. Berlin, Heidelberg: Springer, 2008
- [37] Johanning, Volker; Mildner, Roman (2015): Car IT Kompact. Das Auto der Zukunft - Vernetzt und autonom fahren. Wiesbaden, Springer Vieweg, ISBN 978-3-658-09967-1
- [38] Liggesmeyer, P. (2009): Software-Qualität. Testen, Analysieren und Verifizieren von Software, 2. Auflage. Heidelberg: Spektrum
- [39] Levi, M., Allouche, Y. & Kontorovich, A. (2018): Advanced Analytics for Connected Car Cybersecurity. In: 2018 IEEE 87th Vehicular Technology Conference (VTC Spring). Porto, Portugal, 3-6 June 2018: proceedings, Piscataway, NJ, S. 1–7
- [40] Schmidt, Reinhard: Qualitätsgerechte Entwicklung softwareintensiver technischer Systeme, Shaker Verlag, ISBN 13: 9783832259396, März 2007
- [41] ISO/IEC 25010-2011, 3f.: ISO/IEC 25010-2011-03, Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models
- [42] Bevan, N. (1999): Quality in use: Meeting user needs for quality, in: *Journal of systems and software* (vol. 49), S. 89-96
- [43] Johanning, Volker; Mildner, Roman (2015): Car IT Kompact. Das Auto der Zukunft - Vernetzt und autonom fahren. Wiesbaden, ISBN 978-3-658-09967-1
- [44] Reuter, Florian (2015); Die Integration von Mobilien Diensten ins Fahrzeug – Nachhaltige Wettbewerbsvorteile durch digitale Komplementärgüter und überlegenen Customer Value. Berlin
- [45] Reuter, Florian (2015): Die Integration von Mobilien Diensten ins Fahrzeug – Nachhaltige Wettbewerbsvorteile durch digitale Komplementärgüter und überlegenen Customer Value. Berlin
- [46] <https://www.lindbaum.de/glossar/app>, 11.07.2018
- [47] Knott, Daniel (2016); Mobile App Testing: Praxisleitfaden für Softwaretester und Entwickler mobiler Anwendungen, dpunkt.verlag
- [48] Seiberth, G. (2018): Wie verändern digitale Plattformen die Automobilwirtschaft? Taunus: Accenture
- [49] Deloitte (2018): Automobilbranche 2025. London: Deloitte
- [50] Lemmer, K. (Hrsg.): Neue auto Mobilität. Automatisierter Straßenverkehr der Zukunft (acatech STUDIE), München: Herbert Utz Verlag 2016
- [51] Langmack, H. (2001): Fehlermanagement. In: Managementsysteme in der Lebensmittelwirtschaft. Hamburg: Behr's Verlag, S. 1– 15
- [52] Wolf, G., Leszak, M. Dr. (2007): Fehlerklassifikation für Software. Berlin: BITKOM Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V., Online [erhältlich unter:](https://www.bitkom.org/noindex/Publikationen/2008/Leitfaden/BITKOM-)

- LeitfadenFehlerklassifikation-fuer-Software/080118-Fehlerklassifikation-fuer-Softwarehaftung.pdf (Zugriff: 24.05.2018)
- [53] Rupp, C. (2001): Requirements-Engineering und -Management - Professionelle, iterative Anforderungsanalyse für die Praxis, München, Hanser Verlag
- [54] Leffingwell, D., & Widrig, D. (2000): Managing Software Requirements - A Unified Approach. Boston, San Francisco, New York, Toronto, Montreal, London, Munich, Paris, Madrid, Capetown, Sidney, Tokyo, Singapore, Mexico City: Addison-Wesley
- [55] Grande, M. (2011): 100 Minuten für Anforderungsmanagement. Wiesbaden: Vieweg +Teubner Verlag
- [56] Ries, Eric (2014): The lean startup. How today´s entrepreneurs use continuous innovation to create radically successful businesses. USA, Currency
- [57] Sharp, H. & Hall, T. (2016): Agile Processes, in Software Engineering, and Extreme Programming. Cham, Springer
- [58] Kollmann et al. (2016): Deutscher Startup Monitor 2016: der perfekte Start. Zug: Kpmg
- [59] Sutton, S. (2000): The role of process in software start-up. In: IEEE Software, 17. Jg., Nr. 4, S. 33–39
- [60] Kohl, H. (2016): Grundlagen des Benchmarkings. Berlin: TU Technische Universität Berlin
- [61] Linß, G.: Qualitätsmanagement für Ingenieure, 3. Auflage, Leipzig 2011
- [62] J. D. Power (2017): Vehicle Dependability Study. USA, Online erhältlich unter: <http://www.jdpower.com/ratings/study/Vehicle-Dependability-Study-%28VDS%29-byMake/1881ENG> (Zugriff: 24.05.2018)
- [63] Appbot Pty Ltd (2018): Startseite. Australien, Online erhältlich unter: <https://appbot.co> (Zugriff: 24.05.2018)
- [64] Adjust GmbH (2018): Startseite. San Francisco, Online erhältlich unter: <https://www.adjust.com> (Zugriff: 24.05.2018)
- [65] Göhlich, D. (2016): Methodische Produktentwicklung. Berlin, TU Technische Universität Berlin
- [66] Masing, W. (2014): Das Unternehmen im Wettbewerb, in: Pfeifer, T.; Schmitt, R.; Masing, W. (Hg.), Masing Handbuch Qualitätsmanagement, 6. überarb. Aufl.: Carl-Hanser Verlag
- [67] Strelow, M., Wussmann, M. (2016): Digitalisierung in der Automobilindustrie. Ismaning: Iskander Business Partner, Online erhältlich unter: <http://i-b-partner.com/wpcontent/uploads/2016/08/2016-09-06-Iskander-RZ-Whitepaper-Digitalisierung-in-derAutomobilindustrie-DIGITAL.pdf> (Zugriff: 24.05.2018)
- [68] <https://www.next-mobility.news/connected-car-features-ungenutzt-unbekannt-und-zu-teuer-a-685393/>, 17.06.2018
- [69] [https://www.ford.de/service/ihre-mobilitaet/fordpass?searchid=ppc:DEU-FordEvents-FordPass-Exact:FordPass-Exact:fordpass:e:c:g:GOOGLE&gclid=EAlaIqobChMIssyb6vPd2wIVBFXTCh0OUABJEAAAYASAAEgl-hfD\\_BwE](https://www.ford.de/service/ihre-mobilitaet/fordpass?searchid=ppc:DEU-FordEvents-FordPass-Exact:FordPass-Exact:fordpass:e:c:g:GOOGLE&gclid=EAlaIqobChMIssyb6vPd2wIVBFXTCh0OUABJEAAAYASAAEgl-hfD_BwE)
- [70] <https://my.opel.de/>, 01.06.2018

- [71] <http://www.skoda-auto.de/service/connectivity>, 01.06.2018
- [72] [https://www.toyota.de/service\\_und\\_zubehoer/service/index.json](https://www.toyota.de/service_und_zubehoer/service/index.json), 01.06.2018
- [73] <https://www.renault.de/>, 02.06.2018
- [74] [https://www.volvocars.com/de/zubehoer-und-services/apps-und-bedienung/volvo-on-call?gclid=EAlaIQobChMInd3X7\\_Td2wIVl-FRCh1HHw1SEAAAYASAAEgltUfD\\_BwE](https://www.volvocars.com/de/zubehoer-und-services/apps-und-bedienung/volvo-on-call?gclid=EAlaIQobChMInd3X7_Td2wIVl-FRCh1HHw1SEAAAYASAAEgltUfD_BwE), 02.06.2018
- [75] <https://www.renault.de/>, 01.06.2018
- [76] Meyer, A., Kantsperger, R. und Schaffer, M. (2013): Die Kundenbeziehung als ein zentraler Unternehmenswert – Kundenorientierung als Werttreiber der Kundenbeziehung. Betriebswirtschaftlicher Verlag Dr. Th. Gabler. Wiesbaden
- [77] Jacobsen, J. (2013): Customer Journey, User Experience, Maps und der umfassende Blick auf den Nutzer. <https://www.usabilityblog.de/customer-journey-user-experience-maps-und-der-umfassende-blick-auf-den-nutzer/>, 25.08.2019
- [78] Schadteilanalyseprozess, VDA: Das gemeinsame Qualitätsmanagement in der Lieferkette. Vermarktung und Kundenbetreuung, Schadteilanalyse Feld, ISSN 0943-9412, 2011
- [79] Schmitt, R. und Pfeifer, T. (2015): Qualitätsmanagement – Strategien – Methoden – Techniken, 5. Auflage, München und Wien
- [80] Kristina Zolotarev: Datenqualität im Fehlerabstellprozess – Eine Untersuchung bei Mercedes-Benz Cars, Betriebswirtschaftlicher Institut der Universität Stuttgart, Lehrstuhl für allgemeine Betriebswirtschaftslehre und Wirtschaftsinformatik I, 2017
- [81] ISO/IEC 15504 (2012) Information technology -- Process assessment
- [82] Wimmer, Frank (1985): Beschwerdepolitik als Marketinginstrument. In: Ursula Hansen (Hg.): Verbraucherabteilungen in privaten und öffentlichen Unternehmen. Frankfurt/Main: Campus-Verl. (Campus Forschung, 461), S. 225–247
- [83] Stauss, Bernd; Seidel, Wolfgang (2014): Beschwerdemanagement. Unzufriedene Kunden als profitable Zielgruppe. 5., vollständig überarbeitete Auflage. München: Carl Hanser Verlag
- [84] Kroonder, Michael (2014): Qualitätssicherungsvereinbarungen. In: Tilo Pfeifer und Robert Schmitt (Hg.): Masing Handbuch Qualitätsmanagement. 6. Aufl. München, Wien: Carl Hanser Fachbuchverlag, S. 580–599
- [85] Verband der Automobilindustrie e. V. (2018): Qualitätsmanagement in der Automobilindustrie. 8D Methode Problemlösung in 8 Disziplinen
- [86] Jung, Berndt; Schweißler, Stefan; Wappis, Johann (2017): 8D - systematisch Probleme lösen. 3. Auflage. München: Hanser
- [87] Brückner, Claudia (2011): Qualitätsmanagement. Das Praxishandbuch für die Automobilindustrie. München: Hanser Verlag
- [88] Riesenberger, Carlos; Sousa, Sergio Dinis (2010): Application of the Six Sigma methodology in customer complaints management. In: 21st Annual Conference. 21st Annual Conference. Vancouver, 07.-10.05.2010. Production and Operations Management Society

- [89] Wilde, Ingo (2007): Internetbasiertes Reklamationsmanagement entlang der Wertschöpfungskette mit der erweiterten 8D-Methode. Zugl.: Hannover, Univ., Diss., 2007. Garbsen: PZH Produktionstechn. Zentrum (Berichte aus dem IPH, 2007,2)
- [90] Marchenko, Maxim; Ullmann, Georg; Behrens, Bernd-Arno; Overmeyer, Ludger (2010): Exzellentes Reklamationsmanagement. Text Mining basierte Qualitätsbewertung von 8D-Berichten. In: *ZWF* 105 (12), S. 1102–1107
- [91] Fröhlich, Klaus (2015): Umbruch der Automobilindustrie. In: *ATZ - Automobiltechnische Zeitschrift* 117 (9), S. 98. Online verfügbar unter <https://bit.ly/2yri7X2>, zuletzt geprüft am 08.10.2018
- [92] Pressemitteilung BMVI Forschungsprojekte: online erhältlich unter: <http://www.bmvi.de/SharedDocs/DE/Pressemitteilungen/2017/102-dobrindt-22-millionen-euro-fuer-forschungsprojekte-zum-automatisierten-fahren.html?nn=14462>, zuletzt abgerufen am 09.11.2018
- [93] iTunes Store und Google Play, 01.03.2018
- [94] DIN 55350 Teil 31 (2008), Begriffe der Qualitätssicherung und Statistik
- [95] Google Nutzungsbedingungen – Gewährleistung und Haftungsausschluss: online erhältlich un-ter <https://www.google.de/policies/terms/regional.html>, zuletzt aufgerufen am 09.11.2017
- [96] <https://www.scrum.de/was-macht-product-owner/>
- [97] <https://landing.google.com/sre/interview/ben-treynor.html>
- [98] Schmidt, C. (2016): *Agile Software Development Teams. The Impact of Agile Development on Team Performance*, Heidelberg New York Dordrecht London: Springer
- [99] Cockburn, A. (2002): *Agile Software Development: Software through people*, Boston: Addison-Wesley Longman
- [100] Fazal-Baqaie, M., Güldali, B. & Grieger, M. (2016): Ganzheitliches Qualitätsmanagement in agilen Groß- Projekten. In: Engstler, M., Fazal-Baqaie, M., Hanser, E., Linssen, O., Mikusz, M. & Volland, A. (Hrsg.), Bonn: Gesellschaft für Informatik e.V., S. 109-120
- [101] Bruhn, M. (2016): *Servicetransformation. Entwicklung vom Produktanbieter zum Dienstleistungsunternehmen*. Forum Dienstleistungsmanagement. Wiesbaden: Springer
- [102] Birk, A., Heller, G. (2011): Qualitätsmanagement in der agilen Entwicklung – QM muss sich neu positionieren. In: *OBJEKTspektrum - Ausgabe Agility/2011*, S. 1–5
- [103] Beck, K. (1999): Embracing change with extreme programming, in: *IEEE Computer* (vol. 32), 70-77
- [104] Cockburn, A. (2002): *Agile Software Development: Software through people*, Boston: Addison-Wesley Longman
- [105] Abrahamsson, P.; Salo, O.; Ronkainen, J.; Warsta, J. (2002): Agile software development methods: Review and analysis, in: *VTT publication 478*, Espoo, S. 107-162
- [106] Foegen, M.; Battenfeld, J.; Croome, D.; Dorn, M.; Gansser, C.; Kröll, A. K.; Meyser, A.; Porro, S.; Raak C. (2015): *Der Ultimative Scrum Guide 2.0*, 3. korr. Aufl., Darmstadt

- [107] Schwaber, K.; Sutherland, J. (2017): The Scrum Guide. Der gültige Leitfaden für Scrum: Die Spielregeln, Online: <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-German.pdf> (Zugriff: 16.07.2019)
- [108] Schmidt, C. (2016): Agile Software Development Teams. The Impact of Agile Development on Team Performance, Heidelberg New York Dordrecht London: Springer
- [109] Disterer, G. (2011). ITIL-basierte Inbetriebnahme neuer Anwendungen. HMD Praxis der Wirtschaftsinformatik, 48-57
- [110] Alt, R., Auth, G., & Kögler, C. (2017). Innovationsorientiertes IT-Management mit DevOps - IT im Zeitalter von Digitalisierung und Software-defined Business. Wiesbaden: Springer Gabler
- [111] Humble, J., & Molesky, J. (August 2011): Why Enterprises Must Adopt Devops to Enable Continuous Delivery. The Journal of Information Technology Management, Vol. 24, No. 8 - Devops: A Software Revolution in the Making, S. 6–12
- [112] Mao, K., Harman, M., & Jia, Y. (2016). Sapienz: Multi-objective Automated Testing for Android Applications. Conference: the 25th International Symposium on Software Testing and Analysis (S. 94-105). New York: Association for Computing Machinery
- [113] Jia, Y., Mao, K., & Harman, M. (13. September 2018). Facebook Code. Abgerufen am 10. Dezember 2018 von <https://code.fb.com/developer-tools/finding-and-fixing-software-bugs-automatically-with-sapfix-and-sapienz/>
- [114] Spinellis, D. (Juli 2012): Don't Install Software by Hand. IEEE Software, 29(4), S. 86-87
- [115] Berlin, C. (3. Dezember 2018): Entwicklung des Taycan Cross Turismo Porsche setzt auf virtuelle Prototypen. Abgerufen am 10. Dezember 2018 von <https://www.automotiveit.eu/porsche-setzt-auf-virtuelle-prototypen/entwicklung/id-0064347>
- [116] Sloss, B. T. (2016): O’Rielly Media. Abgerufen am 10. Dezember 2018 von <https://learning.oreilly.com/library/view/Site+Reliability+Engineering/9781491929117/ch01.html#idm140509011490064>
- [117] Brown, A., Kersten, N., Forsgren, N., Kim, G., u. Humble, J. (2016): State of DevOps Report. Puppet + DORA
- [118] Ringbauer, A. (2017): Qualitätsmanagement versus Agilität in IT-Unternehmen, Wiesbaden: Springer Gabler
- [119] Dombrowski, U. (2015): Lean Development - Aktueller Stand und zukünftige Entwicklungen, Berlin Heidelberg: Springer Vieweg
- [120] Böhm, J. (2019): Erfolgsfaktor Agilität - Warum Scrum und Kanban zu zufriedenen Mitarbeitern und erfolgreichen Kunden führt, Wiesbaden: Springer Vieweg
- [121] Bertagnolli, F. (2018): Lean Management - Einführung und Vertiefung in die japanische Management-Philosophie, Wiesbaden: Springer Gabler
- [122] Marrold, L. (2018): Mit Holacracy auf dem Weg zur agilen Organisation, in: Fortmann, H. R.; Kolocek, B. (Hg.), Arbeitswelt der Zukunft. Mit Holacracy auf dem Weg zur agilen Organisation - Trends – Arbeitsraum – Menschen - Kompetenzen, Wiesbaden: Springer Gabler, S. 83–96

- [123] Gorecki, P.; Pautsch, P. (2014): Praxishandbuch Lean Management. Der Weg zur operativen Excellence, 2.Aufl., Regensburg: Carl Hanser Verlag
- [124] Hofert, S.; Thonet, C. (2019): Der agile Kulturwandel. 33 Lösungen für Veränderungen in Organisationen, Wiesbaden: Springer Gabler
- [125] Jeromin, J. (2018): Theorie und Praxis von Leadership-Konzepten mit reduzierten Hierarchien, in: Jeromin, J.; Jourdan, G.; von Nell, F. (Hg.), Leadership in Organisationen mit reduzierten Hierarchien - Praxiswissen für die Führungsaufgabe, Wiesbaden: Springer Gabler
- [126] Maximini, D. (2018): Scrum - Einführung in der Unternehmenspraxis. Von starren Strukturen zu agilen Kulturen, 2. überarb. und erw. Aufl., Berlin Heidelberg: Springer Gabler
- [127] Tödtmann, C. (2018): Fünf Irrtümer, mit denen die Idee von Agilität in Planlosigkeit oder Chaos endet. Gastbeitrag von Internet-Pionier Willms Buhse, Online: <https://blog.wiwo.de/management/2018/08/15/fuenf-irrtuemer-mit-denen-die-idee-von-agilitaet-in-planlosigkeit-oder-chaos-endet-gastbeitrag-von-internet-pionier-willms-buhse/> (Zugriff: 04.07.2019)
- [128] Appelfeller, W.; Buchholz, W. (2011): Supplier Relationship Management. Strategie, Organisation und IT des modernen Beschaffungsmanagements, Wiesbaden: Springer Gabler
- [129] Sánchez-Gordón u. Rory (2016): Understanding the gap between software process practices and actual practice in very small companies. In: Journal Software Quality Journal, Volume 24 Issue 3, S. 549–570
- [130] Nascimento, L. u. Travassos, G. (2017): Software Knowledge Registration Practices at Software Innovation Startups. In: Luciana Maria Azevedo Nascimento, Guilherme Horta Travassos (Hrsg.): Software Knowledge Registration Practices at Software Innovation Startups, New York, NY, S. 234–243
- [131] Hof, J. (2019): Requirements Practices in Software Startups. In: Morris Undergraduate Journal: Vol. 6: Iss. 1, Article 3, S. 1–6
- [132] Gralha, C., Damian, D., Wasserman, A., Goulão, M. u. Araújo, J. (2018): The evolution of requirements practices in software startups, In: Crnkovic, Ivica u. a. (Hrsg.): 2018 ACM/IEEE 40th International Conference on Software Engineering. ICSE 2018: 27 May–3 June 2018, Gothenburg, Sweden: proceedings, Piscataway, NJ, S. 823–833
- [133] Melegati, J. & Goldman, A. (2016): Requirements Engineering in Software Startups: A Grounded Theory approach. In: Conference: Proceedings 22nd ICE/IEEE International Technology Management Conference 2016, S. 57–65
- [134] Giardino, C., Unterkalmsteiner, M., Paternoster, N., Gorschek, T. u. Abrahamsson, P. (2014): What Do We Know about Software Development in Startups? In: IEEE Software, 31. Jg., Nr. 5, S. 28–32
- [135] VersionOne (2017): VersionOne 11th Annual State of Agile Report. Online erhältlich unter: <http://www.agile247.pl/wp-content/uploads/2017/04/versionone-11th-annualstate-of-agile-report.pdf> (Zugriff: 12. Januar 2019)
- [136] Komus, A. & Kuberg, M. (2015): Status Quo Agile. Nürnberg: GPM Deutsche Gesellschaft für Projektmanagement e. V.

- [137] Kuß, A. (2007): Marktforschung. Grundlagen der Datenerhebung und Datenanalyse, 2. Auflage. Wiesbaden: Springer
- [138] Dalle, Besten & Menon (2017): Using Crunchbase for economic and managerial research. Online erhältlich unter: <https://www.oecd-ilibrary.org/docserver/6c418d60en.pdf?expires=1558548106&id=id&acname=guest&checksum=826AB2D9D14B01275E11FA3DA91A94A6>. (Zugriff: 22. März 2019)
- [139] Bogner, A., Littig, B. & Menz, W. (2002): Das Experteninterview. Theorie, Methode, Anwendung. Wiesbaden: VS Verlag für Sozialwissenschaften
- [140] Weddehage, J. (2016): Warum Usability-Tests in der Softwareentwicklung so wichtig sind. Frankfurt: Software & Support Media GmbH, Online erhältlich unter: <https://entwickler.de/online/ux/usability-tests-wichtig-in-softwareentwicklung197870.html> (Zugriff: 24.05.2018)
- [141] Rempel, A., Friedrich, K., Gross, A. (2013): Optimierung eines Workflows für Softwareentwickler bei der Bearbeitung von Arbeitspaketen. Fachstudie Nr. 181, Stuttgart: Institut für Softwaretechnologie, Online erhältlich unter: [ftp://ftp.informatik.unistuttgart.de/pub/library/medoc.ustuttgart\\_fi/FACH-0181/FACH-0181.pdf](ftp://ftp.informatik.unistuttgart.de/pub/library/medoc.ustuttgart_fi/FACH-0181/FACH-0181.pdf) (Zugriff: 24.05.2018)
- [142] Reissland, J. (2013): Konzepte und Methoden zur Prüfung der Gebrauchstauglichkeit (usability) von Software und Produkten. Berlin: TU Technische Universität Berlin
- [143] Dust, R. Prof. Dr.-Ing. (2016): Fachgebiet Qualitätsstrategie und -kompetenz, Technische Universität Berlin, Berlin
- [144] GitHub (2018): Startseite. San Francisco, Online erhältlich unter: <https://github.com> (Zugriff: 24.05.2018)
- [145] agosense. (kein Datum). agosense. Abgerufen am 3. Dezember 2018 von <https://agosense.com/de/mediathek/case-studies-whitepaper/>
- [146] Winkelhake, Uwe (2017): Die digitale Transformation der Automobilindustrie. Berlin, Heidelberg: Springer Berlin Heidelberg. S.200
- [147] Jegl, Stephan (Ausgabe 01.11.2015): Gewährleistungsdaten. Potenziale nutzen statt Kosten verwalten. In: ATZechnik 10 (6), S. 70–73
- [148] Frank Witte (2018): Testmanagement und Softwaretest - Theoretische Grundlagen und praktische Umsetzung, Springer Verlag, ISBN 3658250879

## 10 Anhang

### Anhang 1: Fragenkatalog, der ein Fehlermanagementtool verwalten können

1. Welche Probleme sind aufgetreten?
2. Welcher Art sind die Probleme?
3. Welche Priorität hat die Behebung des Fehlers?
4. Welcher Tester hat das Problem festgestellt?
5. Welcher Testfall ist von dem Fehler betroffen?
6. Welcher Entwickler ist für das Problem zuständig?
7. In welcher Programmversion wurde der Fehler beobachtet?
8. Sind evtl. weitere Versionen vom Fehler betroffen?
9. Wie schwer wiegt der Fehler?
10. Wie ist die betriebliche Auswirkung des Fehlers bzw. die Auswirkung für andere Systeme und Organisationen?
11. Wie hoch ist die Priorität für die Fehlerbehebung?
12. Hat der Fehler Konsequenzen auf andere Funktionalitäten, die dadurch nicht getestet werden können oder fehlerhaftes Verhalten implizieren?
13. Was wurde unternommen, um das Problem zu beheben (z. B. Programmänderung)?
14. Ist der Fehler reproduzierbar?
15. Gibt es Möglichkeiten, das Problem zu umgehen, wenn ja, wie?
16. Ist das Problem wirklich behoben, also nachgetestet?
17. Ist ein einmal behobenes Problem wiederaufgetaucht? Wenn ja, wieso?
18. In welcher Programmversion wird das Problem voraussichtlich behoben sein? Ist die Behebung dann früh genug oder muss sie vorgezogen werden
19. Wann wird das Problem voraussichtlich behoben sein?
20. Wie hoch ist der geschätzte Aufwand zur Lösung des Problems?
21. Wie hoch war der tatsächliche Aufwand, um den Fehler zu beheben?
22. Wie ist der Status des Fehlers zu welchem Zeitpunkt: Wann wurde der Fehler festgestellt, wann wurde die Software geändert, wann fand der erneute Test statt? Wurde der Fehler erneut eröffnet?
23. In welchen Formaten erfolgt die Speicherung? Proprietärer Formate erschweren die Verwendung auf unterschiedlichen Plattformen bzw. machen sie unmöglich
24. Lassen sich Dokumente an die Fehlerbeschreibung anhängen (Screenshots, Logfiles usw.) damit ein Entwickler die notwendigen Informationen und Traces exakt nachverfolgen kann?
25. Sind die Problemdokumente von allen Beteiligten zugreifbar (Dokumentfreigaben, Web-Zugriffe usw.)?
26. Lasst sich im System nach Stichworten suchen? Ist die Suche so komfortabel, dass man mit bestimmten Strings auch bei Hunderten oder Tausenden von Fehlern in kurzer Zeit die Beschreibung und den Status eines bestimmten gesuchten Fehlers finden kann?
27. Lassen sich Fehler zusammenfassen?

(Quelle: Frank Witte, Testmanagement und Softwaretest - Theoretische Grundlagen und praktische Umsetzung)

## Anhang 2: Übersicht Automobilhersteller und Mobilitätsdienstleister

Filter	beliebte Automobil- hersteller*	Nutzfahr- zeughersteller	Automobil- hersteller Deutschland**	Innovative Automobil- hersteller***	Kundenzu- friedenheits- Reports****	Mobilitäts- dienstleister
	<b>Mercedes-Benz</b>	DAF	Audi	Tesla	Lexus	<b>car2go</b>
	Volkswagen	Iveco	BMW	Google-Car	Porsche	<b>Moovel</b>
	BMW	MAN	Fiat	X-Peng	Toyota	DriveNow
	Audi	<b>Mercedes-Benz</b>	Ford	Byton	Buick	Flinkster
	Ford	Renault	<b>Mercedes-Benz</b>	...	<b>Mercedes-Benz</b>	Cambio
	Opel	Scania	Opel		Hyundai	Drivy
	Skoda	Volvo	Renault		BMW	Free2Move
	Toyota	Setra	Skoda		Chevrolet	myScooty
	Renault	Neoplan	Toyota		Honda	Coup eScooter
	Volvo		Volkswagen		Jaguar	eCooltra

\* mobile.de „beliebteste Marken“ \*\*Online-Bestellte Ersatzteile ATP-Autoteile & kfzteile24 \*\*\*eigene Auswahl \*\*\*\*Ranking J.D. Power Report 2017

(Eigene Darstellung)

**Anhang 3: Watchliste (1/3)**

Stand (Datum): 03.03.2018

**Watchlist**

beliebteste Automobilhersteller												
Sterne	Mercedes me	Car-Net	BMW Connected	Audi MMI connect	FordPass	myOpel	Skoda Connect	Toyota Service	MY Renault	Volvo On Call		
5	103	37	193	110	218	14	15	1	3	7		
4	36	17	92	93	70	10	6	0	0	6		
3	30	14	62	61	54	10	5	1	4	5		
2	66	12	46	39	21	9	3	2	1	4		
1	362	114	117	53	49	20	26	7	12	6		
Summe Anzahl Bewertungen	597	194	510	356	412	63	55	11	20	28		
Durchschnitt Bewertung	2.1	2.2	3.4	3.5	3.9	2.8	2.7	1.7	2.1	3.1		

iOS: iTunes

beliebteste Automobilhersteller												
Sterne	Mercedes me	Car-Net	BMW Connected	Audi MMI connect	FordPass	myOpel	Skoda Connect	Toyota Service	MY Renault	Volvo On Call		
5	647	278	354	615	662	835	137	42	160	1.370		
4	359	114	172	296	336	407	58	25	111	809		
3	228	139	206	271	191	280	47	17	66	378		
2	492	279	306	324	115	278	55	25	59	316		
1	2.332	1.882	996	891	374	891	225	74	132	458		
Summe Anzahl Bewertungen	4.058	2.692	2.034	2.397	1.678	2.691	522	183	528	3.331		
Durchschnitt Bewertung	2.1	1.7	2.3	2.8	3.5	3.0	2.7	2.7	3.2	3.7		

Android: Google Play

V.....

### Anhang 3: Wachliste (2/3)

Nutzfahrzeughersteller									Neue Player (OEM)
MY DAF	Iveco Non Stop	DAILY Business UP	MAN mobile24	MAN Telematics	Remote Truck	Scania Fleet	Fiat Ducato Camper Mobile	Ciao Fiat mobile	Tesla
									18
									7
									3
									0
									1
									29
									<b>4.4</b>

MY DAF	Iveco Non Stop	DAILY Business UP	MAN mobile24	MAN Telematics	Remote Truck	Scania Fleet	Fiat Ducato Camper Mobile	Ciao Fiat mobile	Tesla
18	32	8	50	11	199	76	1,994		
0	4	0	5	1	53	91	622		
1	1	1	4	2	27	63	247		
1	2	0	5	2	13	40	128		
2	9	5	4	1	33	92	124		
22	48	14	68	17	325	362	3,115		
<b>4.4</b>	<b>4.0</b>	<b>3.4</b>	<b>4.4</b>	<b>4.1</b>	<b>4.1</b>	<b>3.1</b>	<b>4.4</b>		

Kundenzufriedenheits-Reports						
Lexus Enform Remote	Porsche Car Connect	myBuick	Hyundai Service Guide	myChevrolet	My Honda	Jaguar InControl Remote
33	3	16	2	113	5	7
21	1	13	0	75	1	0
27	1	8	1	94	3	1
39	3	8	4	94	0	2
179	15	19	11	256	19	16
299	23	64	18	632	28	26
<b>2.0</b>	<b>1.9</b>	<b>3.0</b>	<b>1.8</b>	<b>2.5</b>	<b>2.0</b>	<b>2.2</b>

Lexus Enform Remote	Porsche Car Connect	myBuick	Hyundai Service Guide	myChevrolet	My Honda	Jaguar InControl Remote
197	99	293	208	2.794	37	129
66	23	85	52	821	22	68
64	16	79	42	588	19	70
80	12	54	68	549	33	113
346	22	156	392	1.377	121	273
753	172	667	762	6.129	232	653
<b>2.6</b>	<b>4.0</b>	<b>3.5</b>	<b>2.5</b>	<b>3.5</b>	<b>2.2</b>	<b>2.5</b>

Mobilitätsdienstleister									
car2go	Moovel	DriveNow Carsharing	Flinkster	Cambio	Drivy	Free2Move	myScotty	Coup eScooter Sharing	eCooltra
4.813	26	46	470	18	241	31	12	41	3
784	7	13	74	8	58	3	6	9	0
144	5	15	17	2	17	3	0	9	1
24	12	27	6	4	3	5	0	8	0
48	5	90	11	5	17	7	2	49	2
5.813	55	191	578	37	336	49	20	116	6
<b>4.8</b>	<b>3.7</b>	<b>2.5</b>	<b>4.7</b>	<b>3.8</b>	<b>4.5</b>	<b>3.9</b>	<b>4.3</b>	<b>2.9</b>	<b>3.3</b>

car2go	Moovel	DriveNow Carsharing	Flinkster	Cambio	Drivy	Free2Move	myScotty	Coup eScooter Sharing	eCooltra
14.079	1.102	2.860	223	260	2.611	944	34	478	1.009
7.698	338	1.781	140	133	1.141	299	8	132	417
2.241	219	679	98	49	277	141	4	54	135
957	147	499	103	16	149	83	0	74	74
2.094	452	1.208	297	33	331	132	8	468	136
27.069	2.258	7.027	861	491	4.509	1.599	54	1.206	1.771
<b>4.1</b>	<b>3.7</b>	<b>3.7</b>	<b>2.9</b>	<b>4.2</b>	<b>4.2</b>	<b>4.2</b>	<b>4.1</b>	<b>3.1</b>	<b>4.2</b>

**Anhang 3: Wachliste (3/3)**

weitere Unternehmen										
Amazon	ebay	Apple Store	Instagram	Spotify	Sonos	Bosch Smart Home	Philips Hue	McDonalds	Huawei HiLink	Samsung Smart View
984	224.281	327	180.160	47.712	31.536	36	80	133	6	62
451	39.746	88	29.276	5.680	7.446	20	66	44	4	28
451	11.356	69	9.008	1.704	2.190	22	48	44	8	28
574	2.839	50	2.252	568	876	32	58	44	1	17
1.640	5.678	100	4.504	1.136	1.752	92	117	368	7	210
4.100	283.900	634	225.200	56.800	43.800	202	369	633	26	345
<b>2.7</b>	<b>4.7</b>	<b>3.8</b>	<b>4.7</b>	<b>4.7</b>	<b>4.5</b>	<b>2.4</b>	<b>2.8</b>	<b>2.3</b>	<b>3.0</b>	<b>2.2</b>

Amazon	ebay	Apple Store	Instagram	Spotify	Sonos	Bosch Smart Home	Philips Hue	McDonalds	Huawei HiLink	Samsung Smart View
552.424	1.858.287		43.532.238	8.060.826	32.257	60	2.304	4.634	61.025	33.386
146.519	439.538		9.971.367	1.559.351	11.420	34	1.495	691	10.791	8.475
57.242	147.184		3.342.901	472.989	4.030	25	1.062	509	5.960	5.868
24.411	68.949		1.125.662	183.817	3.241	10	1.185	471	2.903	4.378
66.514	188.129		2.320.425	425.666	7.342	10	2.276	3.989	7.790	21.929
847.110	2.702.087		60.292.593	10.702.649	58.290	139	8.322	10.294	88.469	74.036
<b>4.3</b>	<b>4.4</b>		<b>4.5</b>	<b>4.6</b>	<b>4.0</b>	<b>3.9</b>	<b>3.0</b>	<b>3.1</b>	<b>4.3</b>	<b>3.4</b>

## Anhang 4: Ranking Benchmarking

Automobilhersteller								
Hersteller App	IOS: itunes		Android: Google Play		Durchschnitt		Standardabweichung	
	Bewertung	Anzahl Bewertungen	Bewertung	Anzahl Bewertungen	einfach	mit Gewichtung* iT	mit Gewichtung*	
Tesla	4.4	29	4.4	3.115	4.4	4.4	1,1	
Porsche Car Connect	1.9	23	4.0	172	3.0	3,8	1,4	
Volvo On Call	3.1	28	3.7	3.331	3.4	3,7	1,4	
myBuick	3.0	64	3.5	667	3.3	3,5	1,6	
myChevrolet	2.5	632	3.5	6.129	3.0	3,4	1,6	
myOpel	2.8	63	3.0	2.691	2.9	3,0	1,7	
Audi MMI connect	3.5	356	2.8	2.397	3.2	2,9	1,6	
Skoda Connect	2.7	55	2.7	522	2.7	2,7	1,7	
BMW Connected	3.4	510	2.3	2.034	2.9	2,5	1,6	
Jaguar InControl Remote	2.2	26	2.5	653	2.4	2,5	1,6	
Lexus Enform Remote	2.0	299	2.6	753	2.3	2,4	1,6	
Mercedes me	2.1	597	2.1	4.058	2.1	2,1	1,6	
Car-Net	2.2	194	1.7	2.692	2.0	1,7	1,4	
*Berücksichtigung der Anzahl der Bewertungen					∅	2.9	3.0	1.5

Mobilitätsdienstleister								
Hersteller App	IOS: itunes		Android: Google Play		Durchschnitt		Standardabweichung	
	Bewertung	Anzahl Bewertungen	Bewertung	Anzahl Bewertungen	einfach	mit Gewichtung*	mit Gewichtung*	
car2go	4.8	5.813	4.1	27.069	4.5	4,2	1,1	
Drivy	4.5	336	4.2	4.509	4.4	4,2	1,2	
eCooltra	3.3	6	4.2	1.771	3.8	4,2	1,2	
Free2Move	3.9	49	4.2	1.599	4.1	4,2	1,3	
Cambio	3.8	37	4.2	491	4.0	4,2	1,2	
myScotty	4.3	20	4.1	54	4.2	4,2	1,4	
Maovel	3.7	55	3.7	2.258	3.7	3,7	1,6	
DriveNow Carsharing	2.5	191	3.7	7.027	3.1	3,7	1,5	
Flinkster	4.7	578	2.9	861	3.8	3,6	1,3	
Coup eScooter Sharing	2.9	116	3.1	1.206	3.0	3,1	1,8	
*Berücksichtigung der Anzahl der Bewertungen					∅	3.8	3.9	1.4

weitere Unternehmen								
Hersteller App	IOS: itunes		Android: Google Play		Durchschnitt		Standardabweichung	
	Bewertung	Anzahl Bewertungen	Bewertung	Anzahl Bewertungen	einfach	mit Gewichtung*	mit Gewichtung*	
Apple Store	3.8	634	n/a	n/a	n/a	n/a	1,5	
Spotify	4.7	56.800	4.6	10.702.649	4.7	4,6	1,0	
Instagram	4.7	225.200	4.5	60.292.593	4.6	4,5	1,0	
ebay	4.7	283.900	4.4	2.702.087	4.6	4,4	1,1	
Huawei HiLink	3.0	26	4.3	88.469	3.7	4,3	1,3	
Amazon	2.7	4.100	4.3	847.110	3.5	4,3	1,2	
Sonos	4.5	43.800	4.0	58.290	4.3	4,2	1,2	
Samsung Smart View	2.2	345	3.4	74.036	2.8	3,4	1,7	
McDonalds	2.3	633	3.1	10.294	2.7	3,1	1,8	
Bosch Smart Home	2.4	202	3.9	139	3.2	3,0	1,4	
Philips Hue	2.8	369	3.0	8.322	2.9	3,0	1,6	
*Berücksichtigung der Anzahl der Bewertungen					∅	3.7	3.9	1.3

(Quelle: Forschungsstudie QSK der TU Berlin)

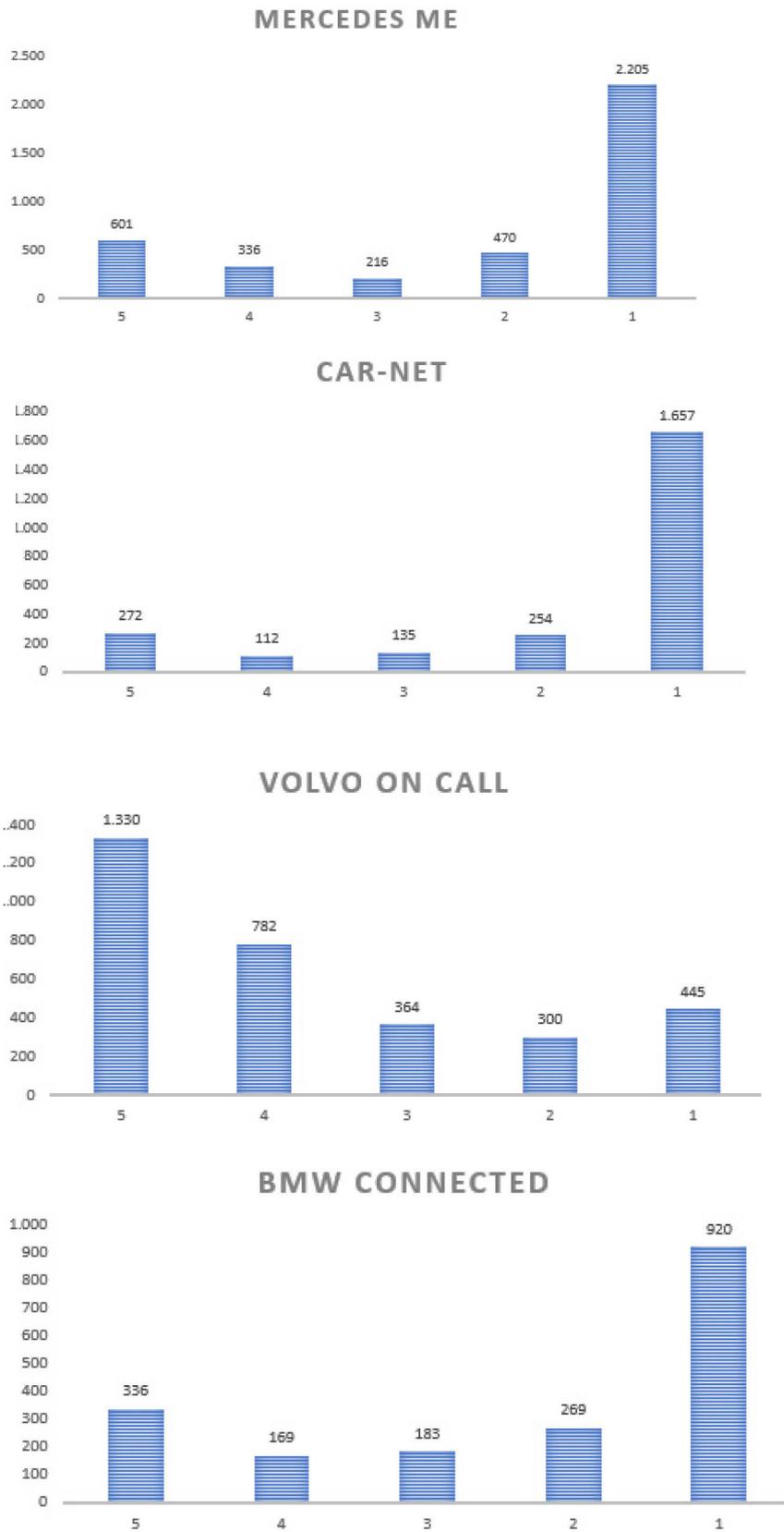
## Anhang 5: Gewichteter Durchschnittswert für die weiteren Unternehmen

Hersteller App	Durchschnitt mit Gewichtung*
Spotify	4,6
Instagram	4,5
ebay	4,4
Huawei HiLink	4,3
Amazon	4,3
Sonos	4,2
∅	4,4

Hersteller App	Durchschnitt mit Gewichtung*
McDonalds	3,1
Bosch SH	3,0
Philips Hue	3,0
∅	3,0

(Eigene Darstellung)

## Anhang 6: Beispiel Bewertung Kundenbewertung



(Eigene Darstellung)

## Anhang 7: 8D-Roadmap

Schritt	Hauptaufgaben	Werkzeuge	Ergebnisse
Schritt 1: Problemlösungsteam	Problemlösungsteam (inkl. Teamleiter) festlegen	8D-Report	Problemlösungsteam ist definiert
Schritt 2: Problembeschreibung	Problem erfassen, vollständig beschreiben und abgrenzen	Fehlersammelkarte, Pareto-Diagramm	Problem ist klar beschrieben und abgegrenzt
Schritt 3: Sofortmaßnahmen	Fehlerhafte Teile aus dem gesamten Umlauf entfernen Maßnahmen treffen, die die Lieferfähigkeit sicherstellen	Interimistischer Arbeitsplan, interimistischer Prüfplan / Control Plan	Kunde (intern/extern) ist mit dem Problem nicht mehr konfrontiert
Schritt 4: Ursachenanalyse	Mögliche Probleursachen ermitteln Ursache-Wirkungs-Zusammenhänge ermitteln und darstellen	Ishikawa-Diagramm, Verlaufsdiagramm, Korrelationsdiagramm	Kernursachen des Problems sind identifiziert
Schritt 5: Auswahl und Verifizierung der Abstellmaßnahmen	Mögliche Korrekturmaßnahmen entwickeln, bewerten und auswählen Ausgewählte Korrekturmaßnahmen erproben und Wirksamkeit nachweisen	FMEA, Erprobungsplan, Prozessfähigkeitsuntersuchung	Wirksamkeit der Korrekturmaßnahmen ist nachgewiesen
Schritt 6: Realisierung und Validierung der Abstellmaßnahmen	Korrekturmaßnahmen organisatorisch verankern Sofortmaßnahmen aufheben	Arbeitsplan, Prüfplan / Plan, Schulungsplan	Korrekturmaßnahmen sind nachhaltig in der Organisation verankert
Schritt 7: Fehlerwiederholung verhindern	Gewonnene Erkenntnisse für andere bestehende Produkte/Prozesse verfügbar machen Gewonnene Erkenntnisse für zukünftige Produkte/Prozesse verfügbar machen	Konstruktionsrichtlinie, Audit-Checkliste, FMEA-Software / FMEA Datenbank	Gewonnene Erkenntnisse werden auch für andere Produkte/ Prozesse genutzt
Schritt 8: Abschluss	Erfolgreiche Umsetzung der vereinbarten Maßnahmen überprüfen und Problemlösungsprozess abschließen	8D-Report	Problemlösungsprozess ist formal abgeschlossen

(Quelle: Verband der Automobilindustrie e. V. (2018): Qualitätsmanagement in der Automobilindustrie. 8D Methode Problemlösung in 8 Disziplinen)

## **Anhang 8: Experteninterview zum Umgang mit Reklamationen in der Software-Branche**

### **Interviewer:**

Ich beschäftige mich im Rahmen meiner Arbeit mit dem Reklamationsmanagement in der Automobilindustrie. Da der Software in Fahrzeugen eine immer größere Bedeutung zuteilwird, bin ich an dem diesbezüglichen Vorgehen in der Software-Branche interessiert. Zunächst möchte ich Sie allerdings bitten, sich und Ihren Fachhintergrund kurz vorzustellen.

### **Fachexperte:**

Ich entwickle vorrangig Server-Software. Genauer gesagt beschäftigt sich meine Firma mit der Schnittstellenentwicklung für verteilte Systeme, die vorrangig in der Werbeindustrie genutzt werden. Folglich bewegen wir uns in einem B2B-Geschäftsfeld. Als Vorgehensmodell nutzen wir Scrum. Ich selbst besitze in diesem Bereich eine Facherfahrung von sechs Jahren und bin aktuell als Teamlead im Backend beschäftigt.

### **Interviewer:**

Vielen Dank für die kurze Vorstellung.

Gehen wir nun von folgendem Szenario aus: Ein Kunde wendet sich mit einer Beschwerde an Ihr Unternehmen. Welcher Prozess wird in der Folge angestoßen? Gibt es diesbezügliche Normen oder Richtlinien innerhalb der Branche?

### **Fachexperte:**

Spezifische Normen oder Richtlinien sind mir nicht bekannt. In der Regel gibt es den folgenden Ablauf: Ein Kunde hat ein Problem (z. B. Anwendung bzw. Feature funktioniert nicht). Dieser Umstand wird zumeist an einen Kundenbetreuer (bei großen Kunden) kommuniziert oder durch eine Beschwerde-Funktion (E-Mail, Formular o. ä.) an eine zentrale Schnittstelle geleitet. In der Folge wird die Beschwerde dem Produktmanagement übermittelt. Dieses sichtet das Problem und setzt sich im Zweifel nochmal mit dem Betreuer oder sogar dem Kunden in Verbindung. Auf diese Weise wird versucht das Problem einzugrenzen bzw. genauer zu beschreiben. Im Anschluss wird die Beschwerde als Task formuliert und auf technisch eindeutige Akzeptanzkriterien heruntergebrochen. Dieser wird an den Product Owner (Produktmanager für ein spezifisches Produkt z. B. User Login oder Statistiken) des Entwicklungsteams weitergegeben. Er priorisiert das Thema und taktet es in den Arbeitsablauf des Entwicklungsteams ein. Die Abarbeitung geschieht folglich parallel zum normalen Tagesgeschäft.

### **Interviewer:**

Existiert während der sich anschließenden Problemlösung ein vergleichbarer Ablauf wie im dargestellten 8D-Prozessschaubild und damit ein definiertes Vorgehen?

### **Fachexperte:**

Ein direkt definiertes Vorgehen besitzen wird nicht. Grundsätzlich ist der Ablauf allerdings ähnlich. Die erste Disziplin (Problemlösungsteam) ist durch das bestehende Entwicklungsteam gegeben. Die Problembeschreibung (D2) existiert bereits. Sie ist das Ergebnis des Prozesses, welchen ich zuvor beschrieben habe. Je nach Dringlichkeit können auch Sofortmaßnahmen (z. B. Backup aufspielen oder Traffic umleiten) ergriffen werden. Das wird aber nur gemacht, wenn sich der Aufwand lohnt bzw. das Problem groß genug ist. Die Disziplinen vier bis sechs bestehen bei uns in nur einem Schritt. Das ist

dann die Umsetzung des zuvor angesprochenen Tasks. Zu diesem Zweck besitzen wir eine Definition of Done, welche definiert wann ein Problem gelöst bzw. eine Aufgabe abgeschlossen ist. Die genauen Inhalte sind teamspezifisch festgelegt. Bei uns ist zum Beispiel vorgesehen, dass alle Akzeptanzkriterien erfüllt und alle Änderungen und Test abgedeckt sind. Automatisierte Tests spielen in der Entwicklung und Problemlösung ohnehin eine sehr wichtige Rolle. Die Disziplinen sieben und acht sehe ich eher durch unsere sogenannten Retrospectives repräsentiert. In diesen besprechen wir die aufgetretenen Probleme erneut und versuchen auf diese Weise nachhaltig aus ihnen zu lernen.

Grundsätzlich sind die Probleme eher kleinerer Natur, weshalb die Schritte nicht getrennt sind. Auf die Analyse der Ursachen folgt daher direkt deren Behebung. Alles andere würde den Prozess unnötig aufblähen und damit verlangsamen.

**Interviewer:**

Wie viel Zeit vergeht in etwa vom Aufkommen bis zur Behebung eines Problems?

**Fachexperte:**

Das kommt ganz auf die Dringlichkeit an. Wenn etwas sehr kritisch ist (z. B. Zahlungsanbieter ist nicht zu erreichen oder Server sind offline) dauert die Behebung oft nur wenige Stunden. Aber der "PDF-Download" kann z. B. durchaus Mal ein paar Tage warten.

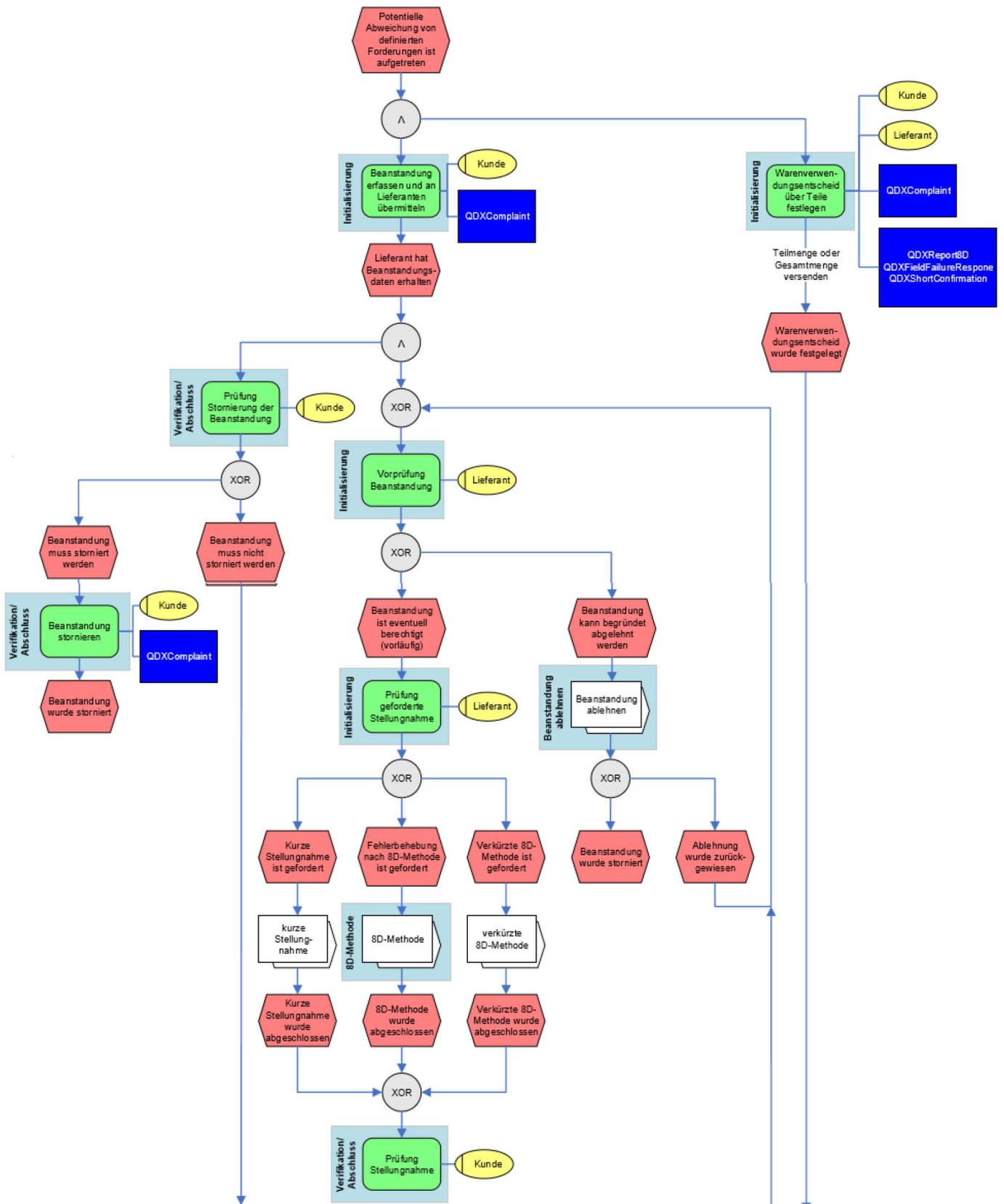
**Interviewer:**

Eine letzte Frage noch: Besteht der Kunde auf einer Dokumentation der Ursachenfindung und Problemlösung, oder ist er nur an einer schnellen Problemlösung interessiert?

**Fachexperte:**

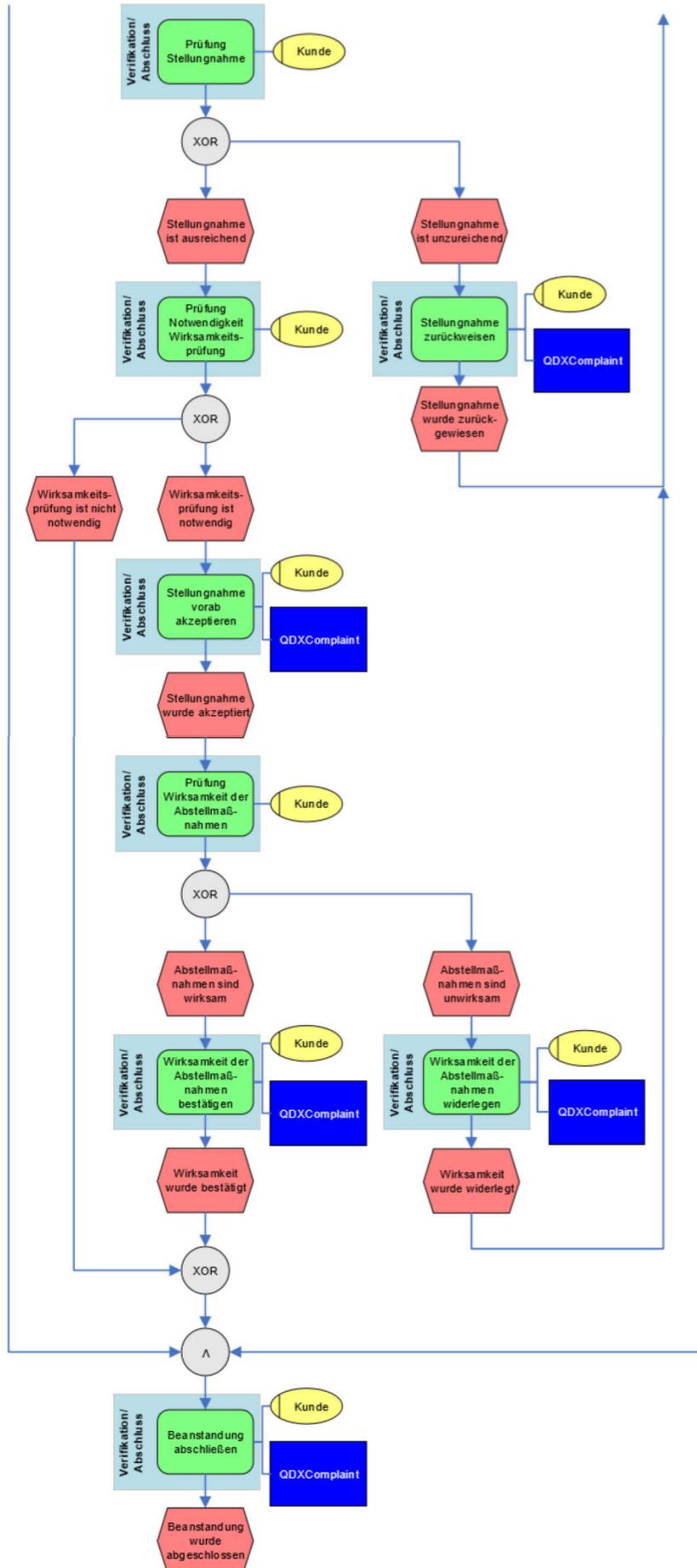
Das ist sehr unterschiedlich und kommt letztlich auf den Kunden an. In der Regel geben wir aber aus unternehmenstaktischen Gründen nur selten detaillierte Informationen heraus. Eigentlich nur wenn es sich um einen internen Kunden handelt. Aus diesem Grund erhalten Kunden i. d. R. lediglich eine Bestätigung über das Auftreten und die Bearbeitung der Beschwerde.

### Anhang 9: Standardisierter Reklamationsprozess Teil 1/2



(Eigene Darstellung)

Anhang 9: Standardisierter Reklamationsprozess Teil 2/2



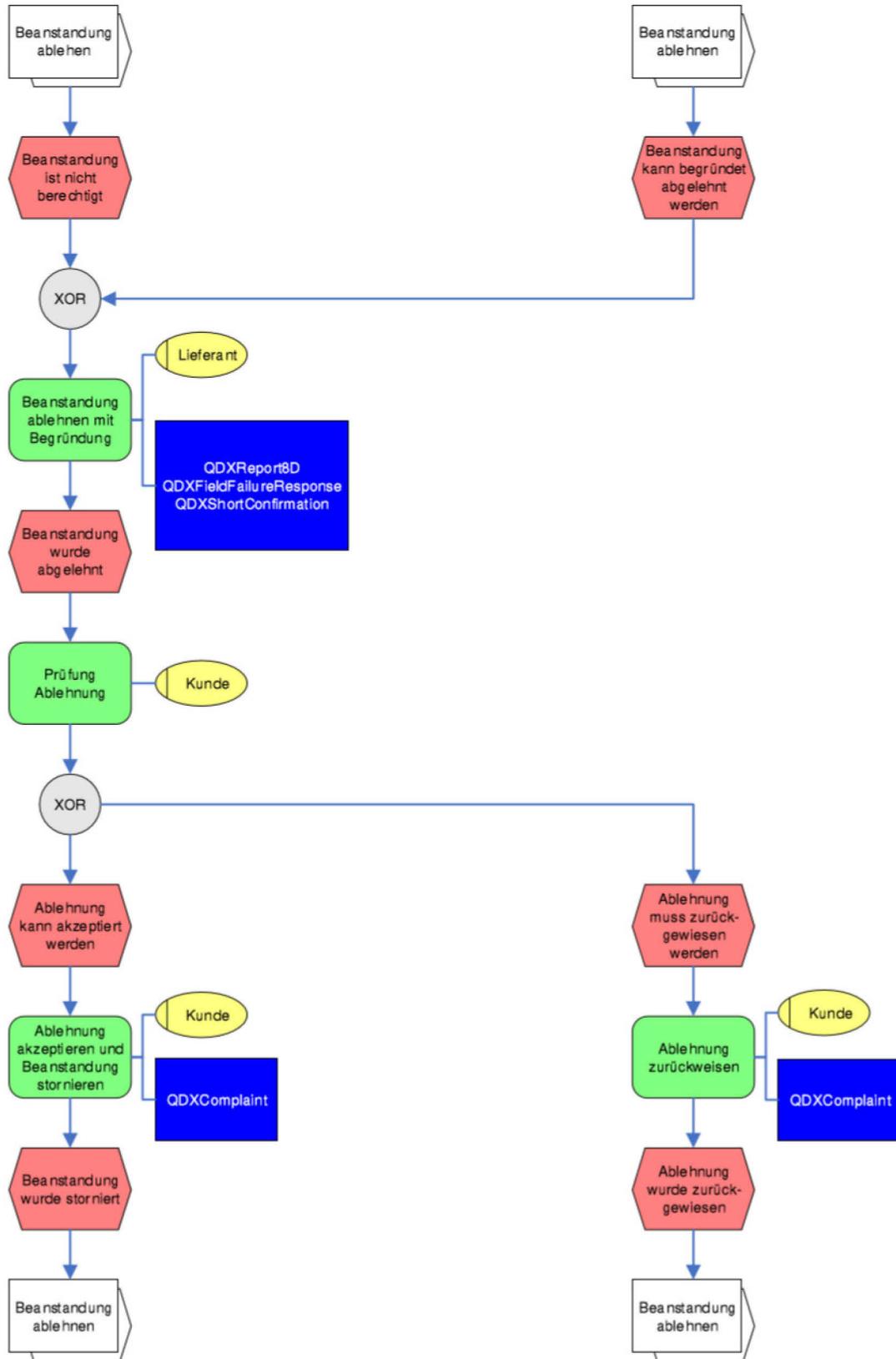
(Eigene Darstellung)

**Anhang 10: Beschreibung des standardisierten Reklamationsprozesses**

Name: ID: Beschreibung:	
Vorbedingungen: Auslöser: Ablauf: Ergebnis: Varianten: Ausnahmen:	
Durchführungsverantwortlicher: Entscheidungsverantwortlicher: Verpflichtung zur Mitarbeit: zu informieren: QDX-Dokument: Input: Output: Anmerkungen:	

(Eigene Darstellung nach VDA)

### Anhang 11: Teilprozess Beanstandung ablehnen



(Eigene Darstellung)

## Anhang 12: Morphologischer Kasten Geschäftsprozess & Verantwortung Fehleranalyse Digitale Dienste

Geschäftspartner	SW	HW	Infrastruktur- anbieter	EE-Teile Lieferanten	Startups	IT Unternehmen	Service- anbieter	Partner Unternehmen	Mobilfunk- anbieter	Backend	...
Merkmale											
Fehlerart/-ort											
Workflow											
Verantwortlicher											
.....											

Verantwortung	Abt. 1	Abt. 2	Abt. 3	Abt. 4	Lft. 1	Lft. 2	....
Fehlerarten							
Fehlerart 1							
Fehlerart 2							
Fehlerart 3							
Fehlerart 4							
.....							

(Eigene Darstellung)

## **Anhang 13: Fragenkatalog Kooperationen zwischen Automobilherstellern und Software-Start-ups**

Sehr geehrte Damen & Herren,

im Rahmen meiner Forschungsarbeit an der Technischen Universität Berlin untersuche ich die Tendenz zu neuen Kooperationen zwischen Automobilherstellern und Software-Start-ups aus dem Bereich Connectivity. Als Ziel soll ein Leitfaden für die konkrete Gestaltung der Zusammenarbeit erstellt werden. Ein Schwerpunkt liegt hierbei auf der Qualitätsebene. Hierfür ist ein Verständnis der Arbeitsweisen und Prozesse beider Seiten notwendig. Mit dem folgenden Fragebogen werden die grundlegenden Vorgehensweisen in der Softwareerstellung erfasst. Der Fragebogen wurde mit der Software LamaPoll durchgeführt. Die erhobenen Daten werden selbstverständlich vertraulich und anonym verarbeitet, informieren sie sich hierzu gerne unter: <https://www.lamapoll.de/Support/Sicherheit>. In den folgenden Punkten sind die Rahmendaten und Links zur Umfrage zu finden:

- Wofür: Ausschließlich für den Forschungszweck der Masterarbeit.
- Umfang der Umfrage: 18 Fragen.
- Dauer der Umfrage: ca. 10-15 min.
- Zeitraum: Die Teilnahme ist bis zum 15.01.2019 geöffnet.
- Link zur Umfrage (Deutsch): <https://lamapoll.de/QM-Software-Start-ups>
- Link zur Umfrage (Englisch): [https://lamapoll.de/QM-Software-Start-ups\\_ENG](https://lamapoll.de/QM-Software-Start-ups_ENG)

Wenn sie darüber hinaus Interesse an der Durchführung von Interviews zu dem beschriebenen Thema haben, dann melden Sie sich gerne jederzeit bei mir. Auf den folgenden Seiten sind die entsprechenden Fragen der Umfrage kurz für Sie zusammengefasst.

An dieser Stelle möchte ich mich im Voraus herzlich für Ihre Unterstützung bedanken.

Mit freundlichen Grüßen

### **Fragenkatalog:**

1. Wie lange gibt es Ihr Unternehmen bereits?
2. Bietet Ihr Unternehmen Software-Produkte für Endkunden oder Unternehmen an?
3. Hat Ihr Unternehmen bereits kooperativ mit einem Automobilhersteller zusammengearbeitet?
4. Wurde Ihr Unternehmen bezüglich eines Qualitätsmanagementsystems zertifiziert?
5. Sind in Ihrem Unternehmen feste Stellen für den Bereich Qualitätsmanagement?
6. Sind in Ihrem Unternehmen dezidierte Verantwortlichkeiten bezüglich der Softwarequalität definiert?
7. Welche der folgenden Normen oder Referenzmodelle werden in Ihrem Unternehmen angewandt.
8. Welche der folgenden Entwicklungsmodelle nutzt Ihr Unternehmen und welche Relevanz haben die genutzten Entwicklungsmodelle?

9. Bitte tragen Sie die von Ihrem Unternehmen genutzten Entwicklungsmethoden in das Freitextfeld unter dem Spaltenkopf "Entwicklungsmethode" ein und bewerten Sie deren Relevanz.

10. Ein hybrides Entwicklungsmodell kombiniert mehrere Ansätze aus verschiedenen Entwicklungsmodellen, nutzt Ihr Unternehmen ein hybrides Entwicklungsmodell?

11. Wenn Ihr Unternehmen kein hybrides Entwicklungsmodell nutzt, dann kreuzen Sie bitte diese Frage an.

12. Wie regelmäßig evaluiert Ihr Unternehmen die in Projekten angewandten Software Engineering Methoden?

13. Würden Sie sagen, dass der Herstellungsprozess von Software-Produkten in Ihrem Unternehmen systematisch abläuft?

14. Würden Sie sagen, dass Ihr Unternehmen mit einem systematischen Anforderungsmanagement arbeitet?

15. Würden Sie sagen, dass Ihre entwickelte Software fortlaufend und strukturiert dokumentiert wird?

16. Würden Sie sagen, dass Ihr Unternehmen mit einem systematischen Änderungsmanagement arbeitet?

17. Wie ist das Vertragsmanagement in Ihrem Unternehmen gestaltet?

18. Über weiteres Feedback und Hinweise Ihrerseits würde ich mich sehr freuen. Nutzen Sie dafür bitte das folgende Freitextfeld.

## **Anhang 14: Interviewfragen – Leitfaden Fehlermanagement**

1. Für welche Produkte bzw. digitale Anwendungen sind Sie in der Abteilung zuständig? Vielleicht können Sie ein paar Beispiele geben.
2. Welche Rolle spielen Softwarezulieferer bei der Entwicklung dieser digitalen Produkte Anwendungen?
3. Wieviel unterschiedliche Softwarelieferanten gibt es?
4. Mich interessiert vor allem wie das Fehlermanagement für die digitale Produkte, die in der Nutzungsphase aussieht, und vor allem ob die Softwarelieferanten auch in dieser Phase eine Rolle spielen? Wenn ja welche?
5. Wie wird im Betrieb eine effiziente und schnelle Fehlerallokation und Zuweisung zu einem bestimmten Softwarelieferanten (der für den Fehler verantwortlich ist) sichergestellt? (gemeinsame Systeme, Backup)
6. Welche Zugriffe auf erste Fehleranalysen/Produktivsysteme werden Softwarelieferanten ermöglicht?
7. Gibt es Vorgaben an die Entwicklungsabteilung des Lieferanten, um ein anschließendes effektives Fehlermanagement während des Betriebes zu gewährleisten?
8. Entstehen die Produkte hauptsächlich durch agile Softwareentwicklung? Hat das Auswirkungen auf die Nutzungsphase, wenn ja welche?
9. Welche Rolle spielen während der Entwicklung Tests und Testautomatisierung, um anschließend ein effektives Fehlermanagement zu gewährleisten?
10. Gibt es Kunden die als Betatester unfertige Produkte testen (gestaffelte Produktivsetzung)?

## **Anhang 15: Standardisierter Fehlerkatalog**

### **Download (D)**

- (D1) Fehler Vertriebskanal / Verfügbarkeit
- (D2) Fehler Unterstützung Betriebssystem
- (D3) Fehler Datei Herunterladen
- (D4) Fehler Installation / Deinstallation

### **Verifikation (V)**

- (V1) Fehler Registrierung / Erstellung Nutzerkonto
- (V2) Fehler Anmeldung / Login
- (V3) Fehler Aktivierung / Freischaltung
- (V4) Prozessrichtlinien / Prozessgestaltung
- (V5) Zustimmung Nutzungsbedingungen / AGB / Zugriffsrechte

### **Support (S)**

- (S1) Störungs- Fehlerbeseitigung
- (S2) Unterstützung bei Anwendungsproblemen
- (S3) Informationsauskunft / Fachauskunft
- (S4) Kontaktaufnahme / Erreichbarkeit / Warteschlange
- (S5) Bearbeitungsstatus / Feedback / Empfangsbestätigung
- (S6) Bearbeitungsdauer

### **Störung im Betrieb (B)**

- (B1) Fehler Funktion „Remote-Zugriff“
- (B2) Fehler Funktion „Statusabfrage“
- (B3) Fehler Funktion „Unterstützung“
- (B4) Fehler Funktion „Weitere Funktionen“
- (B5) Fehler Benachrichtigung / Falschmeldung
- (B6) Transparenz Fehlermeldung
- (B7) Datenverlust / personalisierte Einstellungen
- (B8) Programmabstürze / Programmverklemmung

## Anhang 16: Funktionsübersicht

### 1) Remote-Zugriff

- Remote Parking: Parken auf Knopfdruck, AUTOPILOT
- Remote Control: Ver- und Entriegelung der Türen, Schiebedach, Motorhaube, Fenster, Dach und Kofferraumklappe, Zugriff Rückfahrkamera, Außenspiegel anklappen, Motor Fernstart und -stop
- Remote Search: Signalhorn, Blinkleuchten
- Remote Temperature: De- und Aktivierung der Standheizung und -belüftung
- Remote Settings: Änderung von Einstellungen, Personalisierung: Farben, Spiegeleinstellungen, Radiosender
- Remote Monitoring: Gebiets- Geschwindigkeitsbenachrichtigung: Definition Grenzwerte
- Digital Key: Smartphone als Fahrzeugschlüssel mit Sharing-Funktion

#### Nutzfahrzeuge

- Remote Control: Aktive Steuerung Kippfunktion-Ladefläche, Niveauregulierung, Lampenkontrolle, Arbeitslicht, Innenlicht, Ver- Entriegelung Ladebordwand

#### Elektromobilität

- Remote Control: Ladeeinstellungen Ladevorgang starten und stoppen, Aufladeverhalten definieren, Ladeprofile festlegen, Ladetimer
- Remote Temperature: Vorklimatisierung

### 2) Statusabfragen

- Information: Kilometerstand, Tageskilometerzähler Kraftstoff- Batterieladezustand, verbleibende Restreichweite, elektrische Restreichweite (Hybrid), Reifendrucke, Temperaturen, Garantieinformationen
- Check: ver- entriegelte Türen, aus- eingeschaltetes Licht, Schiebedach, Fenster, Verdeck, Kofferraum
- Geographic Monitoring: Diebstahlwarnung, letzte Parkposition
- Maintenance: Warnmeldungen und überfällige Wartungsereignisse, Verschleißgrad der Bremsanlage, Hinweis zur nächsten Inspektion, Werkstatt hat digitalen Zugriff, Öl- Restlebensdauer, Beifahrerairbag, Batterie- Motorcheck
- Liquid Level: Flüssigkeits-Füllstände Wischwasserstand, Bremsflüssigkeit, Kraftstofffüllstand, Ölfüllstand, AdBlue-Füllstand

#### Nutzfahrzeug: Vorratsdrücke, Achslasten

#### Elektromobilität: Ladefortschritt, Powerwall Anteile Solarenergie, Batterieladestatus, Ladevorgang-Abschlussmeldung

### 3) Unterstützung

- Trip: Erleichterte Reisplanung, Senden der Zieladresse an das Navigationssystem
- Reisplanung und Zeitmanager: Benachrichtigung optimale Startzeit, OnlineVerkehrsinformation (Live-Traffic), Gefahrenbenachrichtigung, Verkehrszeicheninformation, Erweiterte 3D Darstellung, Parkplatz- Tankstellen- Ladestellendienst
- Emergency: Notfall- Pannendienstfunktion, SOS-Notruf RettungskräfteUnfallmeldung, Service-Terminplanung
- Evaluation: Auswertung Fahrprofile, Geschwindigkeiten, zurückgelegte Strecken, Durchschnittsverbrauch und Fahrdauer

- Manual: Digitales Handbuch

Nutzfahrzeuge: Einfahrbeschränkungen aufgrund Fahrzeugdröße und zulässigem Gesamtgewicht, FleetWork-Management, Ladung-Trackingsystem, Lenk- Ruhezeiten, Reports, Bewertung Fahrverhalten

#### 4) Weitere Funktionen

- Online-Services Neuwagenbestellung: Status Produktion
- Integration von Mobilitätsdienstleistern: moovel, car2go, mytaxi
- Concierge Service: Restaurant- Hotelempfehlung, Eventbuchung, Blumenbestellung, Auskunft über Aktienkurse, Kinoprogramm, Wettervorhersage
- In-Car Office: Telefonkonferenzen, autom. Zielübertragung zu Ortsangaben aus dem Terminkalender, Anzeige Konferenzdetails, Anzeige der To Dos
- App Connect: Mirrow-Link, Android Auto, Apple CarPlay
- Autom. Benachrichtigung bei Zielerreichung: SMS Versand an festgelegte Personengruppen
- Online Kartenaktualisierung
- Last Mile Route, Wegbeschreibung
- Routinefahrten: Benachrichtigung bei außergewöhnlichen Ereignissen
- WLAN-Hotspot
- Informationsdienste: Wetter, Nachrichten, Bus- Flug- Bahninformation, Treibstoffpreise, City-Events, Landesinformationen
- Video-Anleitungen
- Digitales Fahrtenbuch: Reisekostenabrechnung
- Valet Modus: Werkstattbesuch, ausblenden persönlicher Daten, Geschwindigkeits- Leistungsbegrenzung, Verriegelung Handschuhfach, Frontkofferraum
- Voice-Pilot: Sprachsteuerung, Vorlesefunktion
- Smart Home: Garage öffnen, Rollläden hochfahren, Temperatursteuerung
- Demo Testversion: virtuelles Fahrzeug

## Anhang 17: Tool zur Fehlererfassung

Support-Ticket  E-Mail, Chat Service  Auswertung Kundenrezension  Feedback App

Klassifizierungsmodell						
Klassifizierungsvarianten						
1. Fehlerschwere	2. betroffene Nutzer	3. Fehlerfrequenz	Fehler-Index			
Datum [Jahr.Monat.Tag]	Fehler-ID [Buchstabe/Zahl]	zusätzliche Fehlerbeschreibung [Notiz]	Fehlerklassifizierungs-ID [1-4]	Fehlerklassifizierungs-ID [1-3]	Fehlerklassifizierungs-ID [1-3]	Priorität [3=hoch - 10=gering]
2018-03-03	(D) Download	Notiz verfassen	(1) Nutzungsausfall	(1) Gesamtheit Nutzer	(1) dauerhaft	(1) + (1) + (1) = 3
2018-03-03	Bitte auswählen	Notiz verfassen	Bitte auswählen	Bitte auswählen	Bitte auswählen	
	Bitte auswählen					
	(D) Download					
	D1					
	D2					
	D3					
	D4					
	D5					
	(V) Verifikation					
	V1					
	V2					
	V3					

Support-Ticket  E-Mail, Chat Service  Auswertung Kundenrezension  Feedback App

Klassifizierungsmodell						
Klassifizierungsvarianten						
1. Fehlerschwere	2. betroffene Nutzer	3. Fehlerfrequenz	Fehler-Index			
Datum [Jahr.Monat.Tag]	Fehler-ID [Buchstabe/Zahl]	zusätzliche Fehlerbeschreibung [Notiz]	Fehlerklassifizierungs-ID [1-4]	Fehlerklassifizierungs-ID [1-3]	Fehlerklassifizierungs-ID [1-3]	Priorität [3=hoch - 10=gering]
2018-03-03	(D) Download	Notiz verfassen	(1) Nutzungsausfall	(1) Gesamtheit Nutzer	(1) dauerhaft	(1) + (1) + (1) = 3
2018-03-03	Bitte auswählen	Notiz verfassen	Bitte auswählen	Bitte auswählen	Bitte auswählen	
			Bitte auswählen			
			(1) Nutzungsausfall			
			(2) funktionale Beeinträchtigung			
			(3) Bedienerfehler / Usability			
			(4) Verbesserungsvorschlag			

Support-Ticket  E-Mail, Chat Service  Auswertung Kundenrezension  Feedback App

Klassifizierungsmodell						
Klassifizierungsvarianten						
1. Fehlerschwere	2. betroffene Nutzer	3. Fehlerfrequenz	Fehler-Index			
Datum [Jahr.Monat.Tag]	Fehler-ID [Buchstabe/Zahl]	zusätzliche Fehlerbeschreibung [Notiz]	Fehlerklassifizierungs-ID [1-4]	Fehlerklassifizierungs-ID [1-3]	Fehlerklassifizierungs-ID [1-3]	Priorität [3=hoch - 10=gering]
2018-03-03	(D) Download	Notiz verfassen	(1) Nutzungsausfall	(1) Gesamtheit Nutzer	(1) dauerhaft	(1) + (1) + (1) = 3
2018-03-03	Bitte auswählen	Notiz verfassen	Bitte auswählen	Bitte auswählen	Bitte auswählen	
				Bitte auswählen		
				(1) Gesamtheit Nutzer		
				(2) Nutzergruppen		
				(3) einzelne Nutzer		



**Anhang 18: Fehlerklassifizierung nach Fehleranalyse (Car-Net)**

Support-Ticket  E-Mail, Chat Service  Auswertung Kundenrezension  Feedback App

		Klassifizierungsmodell					Fehler-Index
		Klassifizierungsvarianten			3. Fehlerfrequenz	Fehler-Index	
		1. Fehlerschwere	2. betroffene Nutzer	3. Fehlerfrequenz	Fehler-Index	Priorität	
Datum [Jahr. Monat. Tag]	Fehler-ID [Buchstabe/Zahl]	zusätzliche Fehlerbeschreibung [Notiz]	Fehlerklassifizierungs-ID [1-4]	Fehlerklassifizierungs-ID [1-3]	Fehlerklassifizierungs-ID [1-3]	[3=hoch - 10=gering]	
2018-03-03	B1	Steuerung Standheizung	(2) funktionale Beeinträchtigung	(1) Gesamtheit Nutzer	(1) dauerhaft	(2) + (1) + (1) = 4	
2018-03-03	V2	Notiz verfassen	(1) Nutzungsausfall	(3) einzelne Nutzer	(1) dauerhaft	(1) + (3) + (1) = 5	
2018-03-03	B3	Live-Traffic	(2) funktionale Beeinträchtigung	(2) Nutzergruppen	(1) dauerhaft	(2) + (2) + (1) = 5	
2018-03-03	N3	Notiz verfassen	(3) Bedienerfehler / Usability	(1) Gesamtheit Nutzer	(1) dauerhaft	(3) + (1) + (1) = 5	
2018-03-03	S1	Notiz verfassen	(2) funktionale Beeinträchtigung	(3) einzelne Nutzer	(1) dauerhaft	(2) + (3) + (1) = 6	
2018-03-03	N1	Notiz verfassen	(2) funktionale Beeinträchtigung	(2) Nutzergruppen	(2) zyklisch	(2) + (2) + (2) = 6	
2018-03-03	S2	Notiz verfassen	(3) Bedienerfehler / Usability	(3) einzelne Nutzer	(1) dauerhaft	(3) + (3) + (1) = 7	
2018-03-03	B2	Datenaktualisierung	(2) funktionale Beeinträchtigung	(2) Nutzergruppen	(3) sporadisch	(2) + (2) + (3) = 7	
2018-03-03	B5	Fehlmeldung: Status (Türen, Heckklappe)	(2) funktionale Beeinträchtigung	(2) Nutzergruppen	(3) sporadisch	(2) + (2) + (3) = 7	
2018-03-03	S5	Notiz verfassen	(4) Verbesserungsvorschlag	(3) einzelne Nutzer	(1) dauerhaft	(4) + (3) + (1) = 8	
2018-03-03	VB1	Notiz verfassen	(4) Verbesserungsvorschlag	(2) Nutzergruppen	(2) zyklisch	(4) + (2) + (2) = 8	
2018-03-03	U5	Notiz verfassen	(4) Verbesserungsvorschlag	(3) einzelne Nutzer	(2) zyklisch	(4) + (3) + (2) = 9	

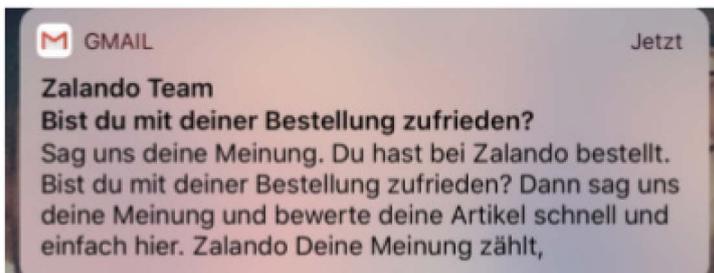
## Anhang 19: Kundenfeedback Kanäle

### 1. Internet-Browser

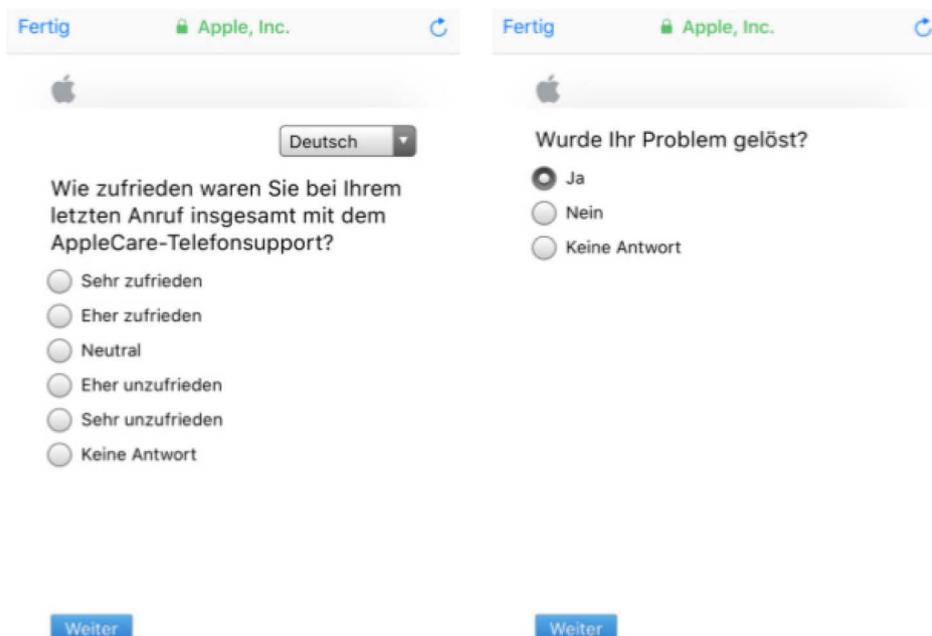


Quelle: <https://translate.google.de/?hl=de&tab=wT>

### 2. E-Mail-Nachricht (persönlich)



### 3. Per App



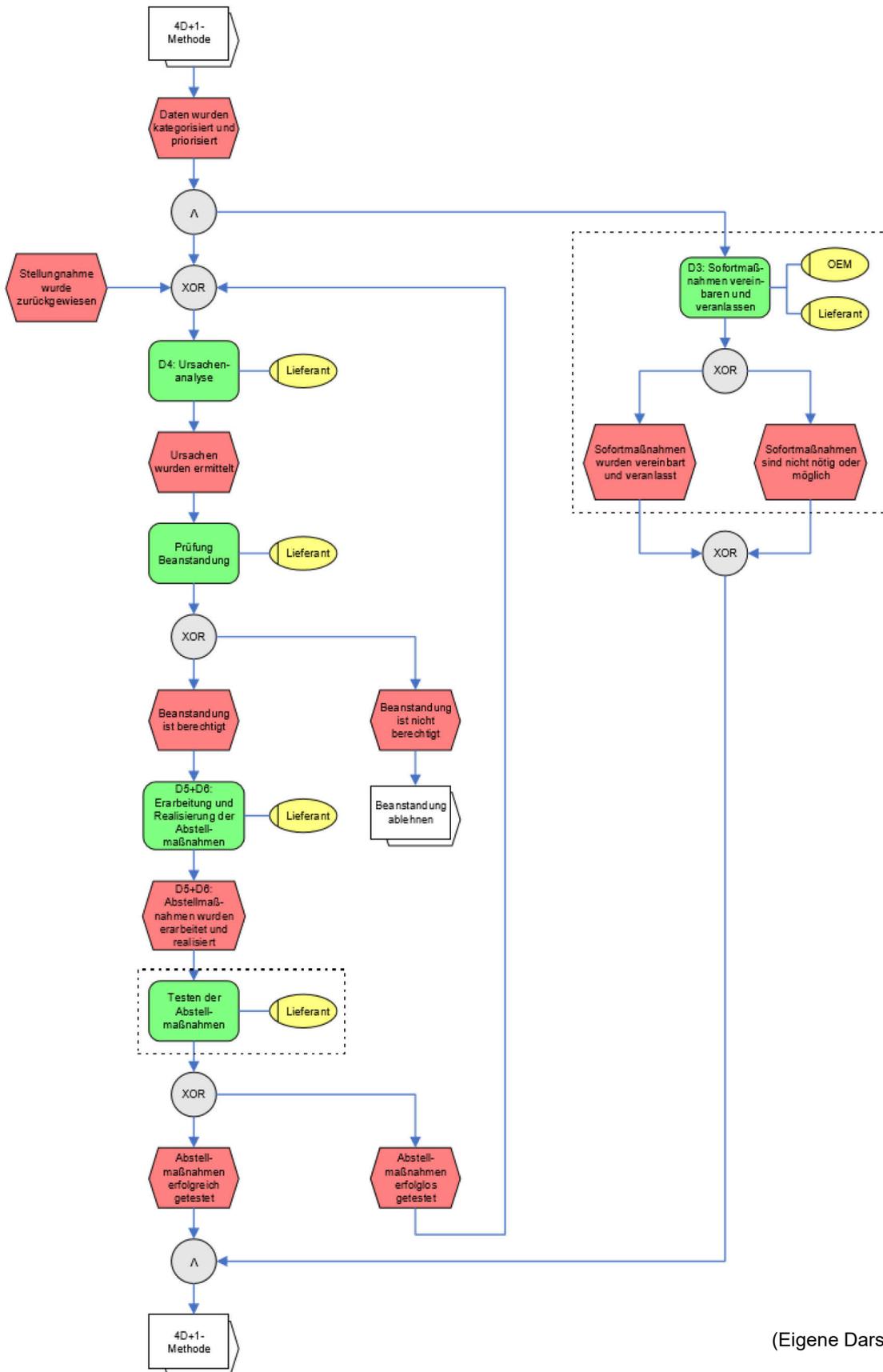
Anhang 20: Fehlervergleich (1/2)

	Automobilhersteller Anzahl: 17	Häufigkeit in %	Mobilitätsdienstleister Anzahl: 10	Häufigkeit in %	IT-Branche Anzahl: 6	Häufigkeit in %	weitere Unternehmen Anzahl: 5	Häufigkeit in %
Download (D)	0	0	0	0	0	0	0	0
(D1) Fehler Vertriebskanal / Verfügbarkeit	0	0,00	0	0,00	0	0,00	0	0,00
(D2) Fehler Unterstützung Betriebssystem	0	0,00	0	0,00	0	0,00	0	0,00
(D3) Fehler Datei Herunterladen	0	0,00	0	0,00	0	0,00	0	0,00
(D4) Fehler Installation / Deinstallation	0	0,00	0	0,00	0	0,00	0	0,00
Verifikation (V)	21		9		5		10	
(V1) Fehler Registrierung / Erstellung Nutzerkonto	1	5,88	1	10	1	16,67	0	0
(V2) Fehler Anmeldung / Login	7	41,18	3	30	2	33,33	1	20
(V3) Fehler Aktivierung / Freischaltung	6	35,29	1	10	1	16,67	4	80
(V4) Prozessrichtlinien / Prozessgestaltung	6	35,29	3	30	1	16,67	3	60
(V5) Zustimmung Nutzungsbedingungen / AGB / Zugriffrechte	1	5,88	1	10	0	0,00	2	40
Support (S)	21		13		10		7	
(S1) Störungen- Fehlerbeseitigung	7	41,18	4	40	3	50,00	4	80
(S2) Unterstützung bei Anwendungsproblemen	3	17,65	3	30	3	50,00	2	40
(S3) Informationsaustausch / Fachauskunft	6	35,29	3	30	3	50,00	0	0
(S4) Kontaktaufnahme / Erreichbarkeit / Warteschlange	1	5,88	3	30	1	16,67	1	20
(S5) Bearbeitungsstatus / Feedback / Empfangsbestätigung	3	17,65	0	0	0	0,00	0	0
(S6) Bearbeitungsdauer	1	5,88	0	0	0	0,00	0	0
Störung im Betrieb (B)	42		16		11		15	
(B1) Fehler Funktion „Remote-Zugriff“	7	41,18	0	0	1	16,67	1	20
(B2) Fehler Funktion „Statusabfrage“	8	47,06	0	0	0	0,00	0	0
(B3) Fehler Funktion „Unterstützung“	3	17,65	0	0	0	0,00	0	0
(B4) Fehler Funktion „Weitere Funktionen“	5	29,41	5	50	2	33,33	4	80
(B5) Fehler Benachrichtigung / Fälschmeldung	5	29,41	0	0	0	0,00	0	0
(B6) Transparenz Fehlermeldung	6	35,29	3	30	2	33,33	3	60
(B7) Datenverlust / personalisierte Einstellungen	2	11,76	2	20	2	33,33	2	40
(B8) Programmabstürze / Programmverkleinerung	5	29,41	5	50	3	50,00	4	80
(B9) Tastatur / Buttons	1	5,88	1	10	1	16,67	1	20
Update (U)	9		6		8		5	
(U1) Fehler Vertriebskanal / Verfügbarkeit	0	0,00	0	0	1	16,67	0	0
(U2) Fehler Unterstützung Betriebssystem	0	0,00	0	0	0	0,00	0	0
(U3) Fehler Datei Herunterladen	0	0,00	0	0	0	0,00	1	20
(U4) Fehler Installation	1	5,88	0	0	0	0,00	1	20
(U5) Häufige Zwangsaktualisierungen	4	23,53	5	50	4	66,67	3	60
(U6) Änderung der Basisfunktionen / Funktionen	3	17,65	1	10	3	50,00	0	0
(U7) Störung im Betrieb in Verbindung mit Update	1	5,88	0	0	0	0,00	0	0
Nutzung (N)	41		14		16		7	
(N1) Zuverlässigkeit	12	70,59	8	80	5	83,33	5	100
(N2) Usability	5	29,41	0	0	3	50,00	1	20
(N3) Wiederholende Eingabe von Login-Daten	11	64,71	1	10	0	0,00	1	20
(N4) Erhöhter Batterieverbrauch	1	5,88	0	0	1	16,67	0	0
(N5) Erhöhter mobile Datenverbrauch	1	5,88	0	0	3	50,00	0	0
(N6) Kostenpflichtiges Abonnement	4	23,53	0	0	1	16,67	0	0
(N7) Funktionale Erweiterungsansprüche	7	41,18	3	30	0	0,00	0	0
(N8) Anforderung / Leistungsanforderung mobiles Endgerät	0	0,00	1	10	2	33,33	0	0
(N9) Auswahl mobiles Endgerät	0	0,00	0	0	0	0,00	0	0
(N10) Werbung / Reklame	0	0,00	1	10	0	0,00	0	0
(N11) Sicherheit / Schutz	0	0,00	0	0	1	16,67	0	0
Verbindung (VB)	32		10		7		9	
(VB1) Zeit Verbindungsaufbau	16	94,12	5	50	5	83,33	5	100
(VB2) Verbindungsaufbau / Datenaustausch / Kopplung	16	94,12	5	50	2	33,33	4	80

Anhang 20 Fehlervergleich (2/2)

	Automobilhersteller Anzahl: 17	Häufigkeit in %	Mobilitätsdienstleister Anzahl: 10	Häufigkeit in %	IT-Branche Anzahl: 6	Häufigkeit in %	weitere Unternehmen Anzahl: 5	Häufigkeit in %
(VB1) Zeit Verbindungsaufbau	16	94,12	5	50	5	83,33	5	100
(VB2) Verbindungsaufbau / Datenaustausch / Kopplung	16	94,12	5	50	2	33,33	4	80
(M1) Zuverlässigkeit	12	70,59	8	80	5	83,33	5	100
(N3) Wiederholte Eingabe von Login-Daten	11	64,71	1	10	0	0,00	1	20
(B2) Fehler Funktion „Statusabfrage“	8	47,06	0	0	0	0,00	0	0
(V2) Fehler Anmeldung / Login	7	41,18	3	30	2	33,33	1	20
(S1) Störungs- Fehlerbeseitigung	7	41,18	4	40	3	50,00	4	80
(B1) Fehler Funktion „Remote-Zugriff“	7	41,18	0	0	1	16,67	1	20
(N7) Funktionale Erweiterungswünsche	7	41,18	3	30	0	0,00	0	0
(V3) Fehler Aktivierung / Freischaltung	6	35,29	1	10	1	16,67	4	80
(V4) Prozessrichtlinien / Prozessgestaltung	6	35,29	3	30	1	16,67	3	60
(S3) Informationsauskunft / Fachauskunft	6	35,29	3	30	3	50,00	0	0
(B6) Transparenz Fehlermeldung	6	35,29	3	30	2	33,33	3	60
(B4) Fehler Funktion „Weitere Funktionen“	5	29,41	5	50	2	33,33	4	80
(B5) Fehler Benachrichtigung / Falschmeldung	5	29,41	0	0	0	0,00	0	0
(B8) Programmabstürze / Programmverklemmung	5	29,41	5	50	3	50,00	4	80
(N2) Usability	5	29,41	0	0	3	50,00	1	20
(U5) Häufige Zwangsaktualisierungen	4	23,53	5	50	4	66,67	3	60
(N6) Kostspflichtiges Abonnement	4	23,53	0	0	1	16,67	0	0
(S2) Unterstützung bei Anwendungsproblemen	4	23,53	3	30	3	50,00	2	40
(S5) Bearbeitungsstatus / Feedback / Empfangsbestätigung	3	17,65	0	0	0	0,00	0	0
(B3) Fehler Funktion „Unterstützung“	3	17,65	0	0	0	0,00	0	0
(U6) Änderung der Basisfunktionen / Funktionen	3	17,65	1	10	3	50,00	0	0
(B7) Datenverlust / personalisierte Einstellungen	2	11,76	2	20	2	33,33	2	40
(V1) Fehler Registrierung / Erstellung Nutzerkonto	1	5,88	1	10	1	16,67	0	0
(V5) Zustimmung Nutzungsbedingungen / AGB / Zugriffrechte	1	5,88	1	10	0	0,00	2	40
(S4) Kontaktaufnahme / Erreichbarkeit / Warteschlange	1	5,88	3	30	1	16,67	1	20
(S6) Bearbeitungsdauer	1	5,88	0	0	0	0,00	0	0
(B9) Tastatur / Buttons	1	5,88	1	10	1	16,67	1	20
(U4) Fehler Installation	1	5,88	0	0	0	0,00	1	20
(U7) Störung im Betrieb in Verbindung mit Update	1	5,88	0	0	0	0,00	0	0
(N4) Erhöhter Batterieverbrauch	1	5,88	0	0	1	16,67	0	0
(N5) Erhöhter mobile Datenverbrauch	1	5,88	0	0	3	50,00	0	0
(D1) Fehler Vertriebskanal / Verfügbarkeit	0	0,00	0	0	0	0,00	0	0
(D2) Fehler Unterstützung Betriebssystem	0	0,00	0	0	0	0,00	0	0
(D3) Fehler Datei Herunterladen	0	0,00	0	0	0	0,00	0	0
(D4) Fehler Installation / Deinstallation	0	0,00	0	0	0	0,00	0	0
(U1) Fehler Vertriebskanal / Verfügbarkeit	0	0,00	0	0	1	16,67	0	0
(U2) Fehler Unterstützung Betriebssystem	0	0,00	0	0	0	0,00	0	0
(U3) Fehler Datei Herunterladen	0	0,00	0	0	0	0,00	1	20
(N8) Anforderung / Leistungsanforderung mobiles Endgerät	0	0,00	1	10	2	33,33	0	0
(N9) Auswahl mobiles Endgerät	0	0,00	0	0	0	0,00	0	0
(N10) Werbung / Reklame	0	0,00	1	10	0	0,00	0	0
(N11) Sicherheit / Schutz	0	0,00	0	0	1	16,67	0	0

Anhang 21: 4D+1-Methode



(Eigene Darstellung)

## Anhang 22: Interviews Anpassung der Rolle QM im Bereich Connectivity

### Interview 1

Interviewpartner: Testmanager in der Entwicklung im Bereich Connected Car mit dem Schwerpunkt [REDACTED] Testing (Abkürzung: IP - Interviews Partner, I - Ich)

*Frage 1: In welchem Bereich sind sie tätig und wie lange gibt es diesen Bereich bereits im Unternehmen?*

IP: Ich bin im Bereich der [REDACTED] Forschung und Entwicklung im Gebiet Connected Car im End-to-End Testing. [REDACTED]

*Frage 2: Welche Stellung besetzen sie in diesem Bereich?*

IP: Eine davon ist Testmanager bzw. Defect Manager für den Managementprozess.

*Frage 3.1: Wie lange sind sie bereits in der Entwicklung für Connectivity Produkte tätig?*

IP: Seit zwei Jahren.

*Frage 3.2: Haben sie bereits Vorerfahrung in der Entwicklung von digitalen Produkten oder von Softwareprodukten gemacht?*

IP: Ja ich war 20 Jahre bei der IBM auch im Testmanagement oder im Test Bereich als Tester, Business-Tester, technischer Tester, Test-Architekt, Test-Consultant und Testmanagement. War 20 Jahre tätig in kundenspezifischen Projekten mit kleinen Projekten mit 30 Mann bis zu 250 Mann Projekten.

*Frage 4: Welche digitalen Produkte im konkreten begleiten Sie denn?*

IP: Aktuell im Bereich Connectivity begleiten wir hauptsächlich alle Themen, die mit der Connected Car [REDACTED] Frontends zu tun haben. Wir testen nur stichprobenartig. Portal-Testen dann als Schwerpunkt die User Experience im Auto. Dann gibt es auch noch ein Parallel-Team im anderen Bereich, die testen quasi hauptsächlich im Auto und die Dienste quasi nur stichprobenmäßig. Das heißt die sind mehr Fahrzeug-zentriert und wir sind mehr Frontend-zentriert also Apps oder Programme.

*Frage 5: Wie sind denn die Bereiche Entwicklung und Qualitätssicherung für Connectivity Produkte im Unternehmen eingegliedert? Also sind das in der Organisation des Unternehmens eigenständige Bereiche oder ist das Qualitätsmanagement auch in die Entwicklung integriert?*

IP: Jain, ja und nein. Generell ist in der Entwicklung immer ein Test Bereich in die Entwicklung eingegliedert. Unsere Abteilung als eigenständige Test-Abteilung wurde vor knapp drei Jahren gegründet, weil festgestellt wurde, dass jedes System seine eigene Qualitätssicherung macht und wenn die Systeme dann zusammengeschaltet werden, das Gesamtsystem nicht funktioniert. Und es wurde dann festgestellt es funktioniert gar nicht, was da von den einzelnen Software Entwicklungsbereichen geliefert wird. Es sollte also ein eigenständiges Testing geben, das Ende zu Ende testet. Sprich über alle beteiligten Systeme hinweg. Also der aktuelle Standard für die Qualitätssicherung: Erst entwickelt und testet jeder für sich selbst und dann kommt ein unabhängiges Team, also halbwegs unabhängiges Team und testet dann nochmal End-to-End mit eigenen Test-Managern und eigenen Test-Ressourcen. Also mit eigenen Autos und auch eigenen Prozessen. Jedes Entwicklungs-Team hat für sich seine Entwicklung und eigene Tests, manche machen es mehr und manche machen es weniger. Theoretisch sollten die sich untereinander absprechen und die ersten Integrationstests machen zwischen verschiedenen Modulen, bevor es zu uns geht als erste unabhängige Testorganisation. Das findet aber nicht immer statt, was heißt, wir sind die Ersten die zum ersten Mal von allen Modulen und Services zusammen ein neues Release zu sehen bekommen.

*Frage 6: Hinweis auf den Produktlebenszyklus bzw. Entwicklungszyklus angefangen von der Anforderungsfestlegung bis zur Außerbetriebnahme. An welchen Stellen entlang dieses Zyklus gibt es Schnittstellen zwischen Entwicklung und Qualitätsmanagement?*

IP: Meine Rolle ist der mittlere Teil, sprich Test- und Integration. Und da eher die rechte Seite von diesem Fall da wir als unabhängiges Test-Team agieren. Also ohne Weisungsbefugnis des Entwicklungsteams und die Entwicklungsteams machen auf der linken Seite die ersten Tests selbst. Wir haben bei 20 bis 25 verschiedene Themen und Module

IP: Der größte Unterschied, den wir haben, ist das bei Hardware-Komponenten die Vorlaufzeiten sehr viel höher sind, sodass die eine Fehlerbehebung nicht so einfach machen können, wie in der Entwicklung von reinen Softwareprodukten oder reinen Connectivity-Produkten, die nur Backend haben und wenig Frontend-

Anteile. Wie ich vorher schon gesagt habe, haben wir auch Hardware-Anforderungen oder oftmals Software im Fahrzeug, die wir benutzen müssen oder brauchen. Das heißt, da sind wir auch eingebunden. Es gibt ein Teil Richtung Backend und Kunden-Frontend, den wir erst sehr spät testen können, sehr zeitnah ein Fehler fixen können und teilweise auch noch in der Produktion Fehler finden können, wenn dort welche auftreten. Im Gegensatz zu Hardware-Anforderungen oder Software, die auf der Hardware im Auto läuft, die wir relativ frühzeitig schon testen müssen, um dortige Fehler zu entdecken. Dass die noch im Fahrzeugentwicklungs-Prozess rechtzeitig behoben werden, da benötigt man eine Vorlaufzeit von bis zu einem halben Jahr.

*Frage 7: Zum Verständnis, der Entwicklungsprozess der Hardware verläuft innerhalb anderer Teams bzw. Projekte als der Software-Erstellungsprozess für das Produkt, richtig?*

IP: Ja und nein. Es gibt quasi ein Generalunternehmer für uns, der in die Hardware-Entwicklung und zusätzlich auch in die Software-Entwicklung integriert ist. Die arbeiten hier sehr eng mit der [REDACTED] Entwicklungsabteilung zusammen, um eben neue Features, die eine Hardware-Änderung im Fahrzeug notwendig machen, auch rechtzeitig einfließen lassen zu können.

*Frage 8: Wer nimmt Kundenanforderungen im Prozess auf bzw. wie wird diese betreut? Ist das ein kontinuierlicher Prozess oder ist das eine einmalige Aufnahme?*

IP: Bei Hardware ist es eine einmalige Aufnahme. Alles andere wird schwieriger einzustellen da CRs immer teurer werden bzw. dann in die nächste Generation verlegt werden können. Im Falle des Fahrzeugmodel [REDACTED] wurde das schon vorher eingeplant indem man sogenannte Fresh-Ups eingeführt hat, das heißt man hat einen Stand SOP und im Nachhinein kommen Fresh-Ups. Das heißt da kommen dann neue Software-Stände, so kann man noch nachträglich neue Anforderungen eintüten. Das heißt man möchte das Fahrzeug immer noch mit neuen Features versorgen können. Zu diesem Zweck versucht man mit gewissen Plattform-Technologien zu arbeiten. Das heißt man hat gewisse Hardware und Plattform-Techniken und man kann durch neue Backend-Kausalitäten neue Features anbieten. Das Fahrzeug bietet gewisse Möglichkeiten, die praktisch schon da sind und daraus kann man dann neue Connect Services stricken. Es gibt Komponenten, die sie frühzeitig anfordern müssen. Das heißt vom Anforderungsmanagement der Hardware reinkommen müssen.

IP: Diese Änderungen kann man über Backend machen und da sind die Zyklen sehr viel kürzer. Also Hauptunterschied zwischen Fahrzeug-Hardware-Komponenten und Connectivity-Produkten sind das der Zyklus tatsächlich sehr viel kürzer ist und leichter erneut zu durchlaufen. Wenn Hardware im Fahrzeug benötigt wird, ist es deutlich schwieriger und wird meist in die nächste Fahrzeuggeneration verschoben. Man versucht das auch zu entkoppeln, das heißt man packt schon möglichst viel in die Fahrzeug-Anforderungen mit rein und benutzt diese als Art Plattform und speist über das Backend neue Funktionen ein. Dadurch ist es möglich den Hardware-Zyklus und den Software-Zyklus voneinander zu entkoppeln.

*Frage 8: Gibt es automatisierte Tests bzw. wer ist für das Testing verantwortlich innerhalb des Softwareprojekt?*

IP: Eigentlich jeder, da es auf allen Ebenen stattfindet. Die Testpyramide von Martin Fowler kennen sie vermutlich? Der untere Teil umfasst die billigsten, dafür in großer Breite, Unit Test oder Komponententests. Nach oben hin werden es dann weniger, da die teurer sind. Sie bauen alle aufeinander auf. Das heißt jedes Entwicklungsteam macht eigene Unit Test vor dem Build und Deploy machen sie ihre Unit-Tests und wenn dann etwas schief läuft und kein Bild aufgebaut wird, danach werden die ersten komplexeren, automatisierten Tests durchgeführt, bevor wir es auf die Testumgebung bekommen. Wir sind in dieser Test-Pyramide relativ weit oben, sprich Richtung Tests, die man auch manuell von Hand machen muss. Wir haben zwar auch automatisierte Tests und eben die Regressionstests, die zum Großteil abzudecken. Und wenn es Richtung Oberfläche-Testing geht, das heißt Backend und Server, die dort Automatisierung testen. Wir haben aber auch im Aufbau Tests, die runter gehen bis zum Fahrzeug. Also automatisierte Tests bis ins Fahrzeug, sprich sie legen User an, die verknüpfen sich mit dem Fahrzeug und die machen im Fahrzeug dann die Tür auf, sie fragen den Tankfüllstand ab oder es wird an den Fahrer zurückgemeldet, wenn sich das Auto bewegt. Das wird dann abgeprüft. Das wird nur für Regressionstest hauptsächlich gemacht und nicht um neue Funktionalitäten zu testen. Die automatisierten Regressionstests werden vor allem für alte Funktionalitäten angewendet, die schon existieren, da diese noch funktionieren sollten. Die gehen teilweise runter bis ins Auto, das heißt wir haben Testautos hier die immer online und immer Empfang haben. Da machen wir die Gut-Tests und die Schlecht-Tests machen wir manuell und die gut-Tests laufen automatisiert ab. Das ist deutlich leichter automatisiert abzufragen, ging alles gut? Wenn ja: Gut, wenn nein: Folgt ein manuelles Fehlermanagement. Das soll verhindern, dass man sich zu Tode klickt und dass die neuen Funktionalitäten dann manuell in der Tiefe getestet werden können, da die einfachen Tests automatisiert gemacht werden.

*Frage 9: Was sind die größten und bedeutendsten Unterschiede zwischen Hardware- und Softwarekomponenten im Bereich Connectivity? Gibt es Maßnahmen, die aus diesem Unterschied resultieren oder resultieren sollten?*

IP: Wir haben auch immer noch eine Kopplung mit der Fahrzeug-Software und Hardware, die im Fahrzeug verbaut ist. Das heißt, wenn wir bei alten NTGs bisher neue Features brauchten, da war die Software sehr starr. Wenn wir darauf neue Dienste entwickelt wollten, waren wir beschränkt auf die Möglichkeiten, die schon da waren. An Signalen, an Möglichkeiten dort einzugreifen im Auto. Und mit den neuen NTGs also Head Units oder Infotainment Systemen können wir ins Fahrzeug eingreifen, wie z.B. Fenster hoch und runterfahren oder Fahrzeug anschalten und ausschalten bzw. Türen aufschließen und zu schließen, da versuchen wir zu entkoppeln, das heißt, man packt schon sehr viel ins Auto rein und stellt eine Art Plattform her. Dann kann man sich relativ schnell neue Software oder neue Features quasi zusammenbauen, ohne dass sie eine Hardware Änderung oder neue Software benötigen.

*Frage: Das heißt man lässt einige Schnittstellen offen?*

IP: Z.B. die ersten iPhones waren noch sehr beschränkt und in späteren Versionen kam dann der App-Store dazu, wo sie in der Lage waren sich nachträglich neue Apps herunterzuladen und die dann auch zu benutzen oder zur Programmierung freizugeben. Also hat man weiterhin die Fahrzeug-Entwicklungszyklen, wo man relativ lange Vorlaufzeiten benötigt, mit sehr harten Prüfkriterien und mit sehr vielen Testkilometern, um alle Eventualitäten abprüfen zu können. Hitzebeständigkeit, Kältebeständigkeit, das sind einfache Sachen, die müssen Sie testen und mit in die Entwicklungsarbeit einstellen. Es werden also mit einem gewissen Puffer Hardware-Ausstattung und Software und mit gewissen Standard APIs, auf die man dann von Außen zugreifen kann und über das Backend anpassen kann, ausgestattet. Das heißt man verlagert mehr Funktionalität Richtung Backend, weil es von dort deutlich leichter ist einen Software-Stand aufzuspielen und zum Beispiel durch A-B-Tests zu prüfen. Das heißt in 10% der Autos arbeite ich mit diesem Stand und die Restlichen 90% belasse ich bei dem alten Stand. Dann beobachte ich was passiert. Also einmal, wo muss ich mich im Fahrzeug-Entwicklungsprozess anpassen, was die langen Laufzeiten angeht, einfach aus Qualitäts- und Sicherheitsgründen und wo kann ich das Fahrzeug schon so bauen bzw. Entwickeln lassen, dass ich praktisch eine bekomme, wo ich dann beliebig neue Dienste draufbauen kann. Das heißt Software in das Fahrzeug reinbringen kann, die für die Nutzer ein Gewinn bringt.

*Frage 10: Agile Arbeitsweisen haben wir bereits angesprochen. Für die Software-Branche ist das ja bereits gängige Praxis, wohingegen in der Automobilbranche erst seit ein paar Jahren. In welchem Umfang wird das in der Entwicklung von Connectivity Produkten angewendet?*

IP: Also bei den einzelnen Teams Richtung Frontend-Bereich, also Customer Experience Bereich, also Smartphone oder Desktop, da wird das schon sehr stark angewendet, weil wir dort relativ kurze Zyklen benötigen, um Fehler zu korrigieren und zu beheben. Bei der Fahrzeugentwicklung können sie aber nicht sagen, ich baue jetzt ein neues Steuergerät ein, weil mein alter Prozessor nicht ausreicht oder weil die Hardware die große Hitze in den arabischen Emiraten nicht aushält. Das bekommen sie nicht hin. Da kommen ganz schnell mal Kosten in mehrstelliger Millionenhöhe auf sie zu und auch die Reputation leidet. Agilität lebt davon, dass ich Fehler machen darf und diese schnell beheben kann. Wohingegen es bei der Fahrzeug-Hardware schwierig ist dort Fehler zu machen. Da müssen sie sehr wohl noch versuchen möglichst fehlerfrei zu arbeiten, bevor es zum Kunden geht oder in den Abnahmeprozess geht, weil es dann in die Produktion anders gebaut werden müsste. Während bei reiner Software sie noch Änderungen selbst in der Produktion schnell machen können. Momentan haben wir da einen Drei-Monats-Rhythmus für Standard-Releases, aber da müssen wir noch schneller werden. Bei Porsche sind sie dabei alle zwei bis vier Wochen neue Features rauszubringen. Das funktioniert für die Kunden-zentrischen Sachen, sobald sie aber ins Fahrzeug rein müssen, haben sie das Standardproblem, mit zu langen Laufzeiten und eben sehr hohen Qualitätsansprüchen.

*Frage 11: Wie heißen die Projektmanagement-Methode bzw. das Vorgehensmodell, das in der Entwicklung angewendet wird? Ich habe bereits einiges gelesen über Scrum und Extreme Programming, gibt es dafür auch einen Namen?*

IP: Ja und nein. Jedes Entwicklungsteam darf selbst entscheiden, ob sie Scrum oder Kanban nehmen. Das hängt vom Team ab oder den Anforderungen, die vom Management oder dem Kunden an das Produkt gestellt werden. Da geben wir keine Vorgaben. Unser Team ist im SW-Entwicklungsprozess relativ weit hinten. Das können sie sich so vorstellen, vorne erfolgt Anforderungsfestlegung, Grob- und Feinplanung, Implementierung dann der erste Schnitt zwischen Implementierung und Integration. Dieser Abschnitt läuft komplett agil ab und jedes Entwicklungsteam darf das für sich entscheiden, welche Entwicklungsmethode sie verwenden. Bei dem ersten Test und Integration, wo praktisch alle Systeme zusammenkommen, da fahren wir naja nicht Wasserfall, aber auf diesen Stand, da testen wir alle Funktionalitäten durch, wir machen die Regressionstest Ende-zu-Ende wirklich bis ins Auto rein. Wir haben echte Autos bei uns, die werden

gefahren, um eben Fehler im Echtbetrieb zu sehen. Wechselnde Mobilfunk-Verhältnisse oder Dopplereffekte sind praktisch Sachen, die sie nicht richtig simulieren können. Und um quasi diese Fehlerquellen auch abzufangen wie z.B. Doppel-Verarbeitung von Signalen, eine Rückmeldung kommt vom Auto nicht rechtzeitig an, unterschiedliche Latenzzeiten in den Märkten europaweit. All diese Sachen, die nicht im Standard-Entwicklungstest rauszubekommen sind. Das heißt wir sind dann eher in Richtung Test und Integration das heißt normale Qualitätssicherungsmaßnahmen unterwegs. Sprich wir machen die kompletten Regressionstests plus einen Test von allen neuen Features und melden das dann an die entsprechenden Entwicklungsteams zurück. Darauf aufbauend können die dann mit Scrum oder Kanban arbeiten. Also uns regelmäßig neue Versionen bringen. Bis zu den festgelegten Zeiten, wo man sagt, jetzt geht es in die Produktion. Davor wird der aktuelle Stand für drei bis vier Wochen eingefroren, um diesen dann im Gesamtkontext zu testen. Und bis dahin darf jede agile Methode angewendet werden, die einen wollen Scrum, die anderen Kanban, manche wollen Wasserfall. Das ist uns nach hinten hinaus egal, solange die sagen, ok hier ist die Software bitte schaut von extern noch mal drauf im Zusammenspiel mit allem anderen System inklusive Auto. So wie ein echter Kunde dann eben auch am Ende die Sicht auf das Gesamtsystem hat. Also ob die dann einen riesigen Entwicklungszyklus machen von maximal acht Wochen wo dann ein neues Release kommt oder ob die dann wöchentliche Sprints machen mit Scrum oder Kanban, ist uns relativ egal. Das sind deren eigenen Entwicklungszyklen. Was uns interessiert ist, dass die Software im versprochenen Umfang abgeliefert wird. Das heißt, dass die neuen Features nicht nur für ein neues System gebracht werden, sondern für vier, fünf, sechs verschiedene Systeme.

*Frage 12: Das heißt das Testing steht da außerhalb und ist nicht in diese Zyklen mit einbegriffen?*

IP: Genau sie haben vorne rum bis zur ersten Hälfte der Test und Integrations-Phase, da haben sie diese vielen Zyklen oder auch nur einen Zyklus von jedem Entwicklungs-Team. Das ist eben der Entwicklung selbst überlassen und dann kommen wir und testen einmal komplett über alle Systeme mit der Komplettübersicht. Also aus Endkunden-Sicht, sprich dass dann auch im Auto ankommt, dass beim Fahren dann auch Rückmeldung kommt. Über alle beteiligten Hardware-Komponenten und Software-Komponenten muss das dann ja auch funktionieren. Dann geht das in die Produktion für aktuell noch 3 Monate. In Zukunft werden die Zyklen dann kürzer werden. Das geht dann schon eher in Richtung Continuous Integration, Continuous Testing und Continuous Deployment. Einfach die Entwicklung zu verkürzen. Aber es wird immer eine Phase geben, wo praktisch alle Systeme eine neue Funktionalität gemeinsam abgeben müssen Wo es dann in den End-to-End reingeht für drei bis vier Wochen und dann geht es erst an den Kunden heraus. Damit stellen wir sicher, dass alle Systeme auch zusammen die neue Funktionalität liefern können und nicht jemand sagt, ja bei mir hat es aber funktioniert sie haben die AP falsch verstanden. Das kann ja passieren.

*Frage 13: In welcher Form werden denn Mitarbeiter in der Abteilung bzw. in dem Team gefördert und weiterentwickelt? Gerade hinsichtlich dieser Testing-Maßnahmen und Qualitätssicherungsmaßnahmen?*

IP: Im Team haben wir zwei Schwerpunkte. Also einmal diese Hardware-Technik, weil sie ja auch immer Fahrzeug mit testen müssen und oftmals eben auch im Fahrzeug das Problem ist. Das heißt z.B., dass die gesendeten Signale nicht die gewünschte Information richtig wiedergeben. Das heißt, Sie müssen ins Backend schauen und dort etwas anders machen, also versuchen, den Fahrzeug-Fehler über das Backend zu lösen. Dafür brauchen sie jemanden der sich Fahrzeug-technisch relativ gut auskennt. Also auch Richtung Kernebene und darunter. Da haben wir Leute, die sich dorthin weiterentwickeln. In Zukunft gibt es ja auch diesen Can over Ethernet. Das heißt, dass ein Teil des Teams das Fahrzeugtechnik Know-how aufbaut und der andere Teil, der sich Fachwissen in der Softwareentwicklung aufbaut. Das heißt verschiedene agile Testing Methoden die es da gibt z.B. die ISTQB-Schulung.

*Frage 14: Welche Methoden und Werkzeuge gibt es denn, welche sich an ständiger Verbesserung ausrichten im Team oder im Projekt bzw. Produktentwicklung? Was wird da denn gemacht?*

IP: Für uns ist es wichtig, ein starkes Release Management zu haben. Das heißt, was, wann kommt, welches Feature zu uns in welcher Baureihe, damit wir wissen gegen was testen wir eigentlich. Das widerspricht in gewisser Weise dem agilen Gedanken, bei dem ich erstmal ausprobieren, was gehen könnte. Wir müssen aus Endkunden-Sicht wissen, was kommt und welche Systeme müssen für dieses neue Feature geändert werden. Also wir bekommen keine Klassendiagramme oder Sequenzdiagramme, sondern die Use Case Beschreibung aus einer Kundensicht. Und das heißt, da brauchen wir ein Anforderungsmanagement, was uns sagt, wann kommt welches Feature und in welchem Markt soll es kommen. Da brauchen wir schon Vorlauf von einem halben bis dreiviertel Jahr, um sich darauf einzustellen. Designabstimmung kann dann auch immer noch während dem Test passieren. Also wenn die GUI nicht funktioniert oder nicht effizient ist dann kann man da auch noch kleine Änderungen machen. Aber prinzipiell bei einem Feature zu sagen, nein das soll jetzt komplett anders verlaufen, ich mache das jetzt in zwei Wochen anders. Das funktioniert bei uns auch nicht, speziell, wenn mehrere Systeme beteiligt sind und Änderungen an jedem System gemacht wurden. Dann kündigen wir ein Feature ab und sagen, kommt für das nächste Release in zwei Monaten wieder und macht euch Gedanken darüber, wie ihr das kundenfreundlicher machen könnt. Das haben wir

auch schon gehabt das Features abgelehnt wurden, da sie nicht kundenbedienbar waren. Das kam dann eben im nächsten Release noch mal deutlich verbessert.

*Frage 15: Werden die dann schon bei der Produkt-Anforderungsaufnahme mit dem Kunden diskutiert oder passiert das dann wirklich erst wenn es schon im Entwicklungsprozess ist?*

IP: Es kommt drauf an. Es gibt verschiedene Wege, wie neue Features reinkommen können, da gibt es keine richtigen Vorschriften. Es gibt ein eigenes Team, das nennt sich Ideas to Product, welches wirklich strategisch überlegt, wie wollen wir uns von den Mitbewerbern in Zukunft abheben. Da werden strategische Zukunftsfelder oder Themen auch definiert. Dann gibt es noch das Team für einzelne Customer Function, die dann verantwortlich sind für das neue Feature aus Kundensicht und dann auch eigne Entwicklungsteams bekommen. Die machen dann auch die Abstimmung mit den Verträgen und klären auch schon, was die rechtlichen Anforderungen sind und wo kann man es überall anbieten. In der EU ist das dann aktuell noch relativ einfach, weil dort ähnliche rechtliche Rahmenbedingungen herrschen, aber sobald sie dann in die USA gehen kommt eine strenge Produkthaftung ins Spiel. Das heißt, da müssen wir auch nochmal Rücksprache halten. Bevor das in die Produktion gehen kann, benötigen wir rechtlich abgesicherte Nutzungsbedingungen in den Sprachen, in denen wir es dann auch vertreiben wollen. Und um das Ganze zu koordinieren gibt es das Customer Function Owner Teams, dass diese Themen dann treibt. Eine andere Quelle für neue Features ist die Marktforschung, die sagt, okay wir brauchen etwas Neues. Das stammt dann aus neuen Kundenanforderung. Oder der Vorstand ■■■ hat auf der Messe irgendwas Cooles gesehen und kommt dann mit der Anforderung, dass er das auch haben möchte. D.h. es gibt verschiedene Wege zu neuen Anforderungen, aber es gibt ein zentrales Demand Management, welches diese Sachen aufnimmt und bewertet und dann auch entsprechend mit den Entwicklungsteams den Aufwand bewertet. Das Demand Management bringt das dann auch rein. Es ist also so eine Art "Pipeline". Aber wer das in die "Pipeline" reinstellt, dass ist egal.

*Frage 16: Wo sehen Sie die größten Verbesserungsmöglichkeiten im Produktlebenszyklus? An welchen Stellen?*

IP: Das ist jetzt aus meiner Sicht und Erfahrungen als Tester. Ich sage immer, die einfachste Verbesserungsmaßnahme ist, wenn man sich vorher mehr Gedanken darüber macht und eigenständig testet und nicht einfach nur runter programmiert. Das heißt praktisch den Blick über den Tellerrand zu werfen und das eigene Produkt in einem Gesamtkontext von Fahrzeug bzw. von Backend zu Frontend zu sehen. Das heißt, aus Sicht eines Endkunden, der eben nicht nur diese einzelne Funktionalität sieht/benutzt, sondern eben die Funktionalität eingebettet hat in einer Connected Cars App oder einem anderen Frontend. Das heißt es sollte sich ähnlich verhalten und benutzerfreundlich sein, nicht nur für Deutsche sondern auch Japaner, Koreaner, Amerikaner. Aber das ist oftmals das Problem, dass man sieht es wurde am Anfang nicht genügend durchdacht. So sehe ich das aus Tester-Sicht. Ich hätte es am liebsten so, dass ich eine Software bekomme, die sauber abgestimmt ist zwischen allen beteiligten Parteien und ab sofort benutzbar ist ohne Fehler. Aber das ist eine Wunschvorstellung.

I: Ja das ist der Idealfall.

IP: Genau und in der Realität weicht man eben davon ab. Also wie gesagt, ich war davor 20 Jahre bei der IBM bei allen großen Kunden hier in Europa, und auch Mittelständler und die kämpfen praktisch alle mit den gleichen Problemen. Sprich sehr ehrgeizige Zeitpläne etwas Neues zu bringen, damit man beim Kunden einen Unique Selling Point bekommt. Und dementsprechend wird dann möglichst schnell, wenn es dumm läuft, vorne das Anforderungsmanagement verkürzt dargestellt oder nicht gemacht. Die Entwicklung hat auch nicht so viel Zeit. Das heißt Unit-Tests fallen runter oder es werden nur sehr Einfache gemacht und dann fehlt noch eine Absprache zwischen verschiedenen Systemen. Letztendlich kommt es dann das erste Mal im End-to-End Test zusammen und dann knallt es. Das ist weniger ein Fahrzeug-spezifisches oder Software-spezifisches Problem, sondern ein allgemeines Problem in der Softwareentwicklung. Deshalb haben wir auch versucht das agil zu machen, das heißt, um schnellere Zyklen zu bekommen indem ich einfach Software schneller in den End-to-End Test gebe und früher Rückmeldungen bekomme. Und da ist das Problem, dass wenn ich viele verteilte Software-Entwicklungsteams habe, die zusammenarbeiten müssen. Irgendwann bringt man dann alles zusammen und dann knallt es meistens gewaltig, wenn vorher zu wenig abgestimmt wurde.

I: Verstehe. Das wird oftmals in der Literatur übersehen. Man geht immer davon aus, eine Software zu haben und ein Entwicklungsteam, das daran arbeitet und dabei dann möglichst auch zusammenarbeitet. Aber es ist eben meistens so, dass mehrere Projekte parallel laufen und dann auch zusammenfließen, da müssen auch die Abstimmung bzw. die Kommunikation stimmen.

IP: Und da braucht man eben am Anfang schon einen sauberen Schnitt, eine saubere Trennung. Das ist ein Ansatz den sehen sie auch bei Amazon, Apple und Android. Die haben eine Plattform, auf der bieten sie

eine große Fülle an Funktionalitäten, die dann App-Entwickler zusammenbauen bzw. darauf aufbauen können. Das heißt es findet eine Entkopplung statt zwischen verschiedenen Systemen und fachlichen Modulen. Dann hat man das Problem, dass man zwar agil arbeitet aber am Ende muss es eben zusammenpassen und wenn man dann Vorne keine Absprachen hat bzw. keine verbindlichen Absprachen gemacht wurden, was ja dem Agilen widerspricht, na ja nicht widerspricht, aber da liegt eben nicht der Schwerpunkt dann knallt es hinten. Das heißt, wenn bei der agilen Entwicklung dann später ein Produkt rauskommt und für den Kunden auch eine Kundenerfahrung entstehen soll, braucht es eine Trennung in eine Plattform, die die Basis-Funktionalitäten bereitstellt und darauf aufsetzenden fachlichen Modulen / Apps. Also quasi eine Plattform, worauf dann die einzelnen neuen Komponenten aufsetzen, die der Kunde dann benutzen kann. (5 zeilen)

I: Entschuldigung das letzte Wort, den letzten Satz habe ich gerade nicht verstanden.

IP: Das Ziel ist es die vielen Module oder die vielen Entwicklungssysteme als ein Kunden-Produkt abzuliefern. Das heißt sie sollten sich überall ähnlich verhalten oder ein gleiches Profil haben. Zum Beispiel bei der Connected Car App [REDACTED]. Da gibt es mehr als 15 verschiedene Entwicklungsteams, die da die einzelnen Komponenten rein liefern und die sollten sich alle ähnlich verhalten. Alles was unter einem gemeinsamen Einstieg an einer Oberfläche kommt, sollte sich ähnlich verhalten. Das heißt es muss gewisse Vorgaben geben, wie die Einbindung auszusehen hat. Also rein agil, d. h. wenn sie machen, was sie wollen, funktioniert das nur bedingt, wenn sie eine App haben und verschiedene Entwicklungsteams die verschiedene Funktionalitäten in dieser App liefern. Das heißt man benötigt gewisse Vorgaben, möglichst allgemein gehalten soweit es geht, aber eben auch so detailliert gehalten wie möglich.

*Frage 17: Ist es denn auch vorstellbar, wenn man das vergleicht mit der Smartphone Industrie, dass man eben sagt, man hat eine Plattform und eben mehrere Schnittstellen, wo diese einzelne Software wie eine Art App eingespielt werden und in dieses Gesamtsystem integriert wird?*

IP: Genau das ist dann das Ziel. Das ist praktisch die Entkopplung bzw. Abstrahierung vom Fahrzeug und von der Software, die darauf nachher laufen soll.

*Frage 18: Also dann praktisch das Fahrzeug ansehen wie ein Mobile Phone. Und da dann durch einen App Store praktisch die einzelne Software downloaden und anbringen durch vordefinierte Schnittstellen. Das wäre praktisch die Idee dahinter?*

IP: Genauso ist praktisch die Idee das man versucht eine Plattform herzustellen wie beim iPhone. Wobei auch da haben sie unterschiedliche Hardware Plattformen. Das heißt sie haben Apps, die laufen auf jeder Hardware und es gibt Apps, die nur laufen, wenn sie auch den neuesten Prozessor haben. Aber es läuft alles über die gleiche Plattform. Letztendlich versuchen sie eine Plattform drunter zu bauen, bei der 80% aller Funktionen für alle Fahrzeuge funktionieren und der Rest nur für die neuesten Fahrzeuge, die noch mehr Möglichkeiten hardwareseitig bereitstellen können. Damit ich nicht drei Jahre später feststelle, okay die Software, die ich im Auto drin habe, sieht einfach veraltet aus.

*Frage 19: Genau das ist auch das Hauptproblem, was ich jetzt eben herausgehört habe. Und wenn das dadurch gelöst oder verbessert werden kann, wäre das ja schon ein ziemlich wichtiger Step.*

IP: Wobei man darf auch da nicht vergessen, dass ein Smartphone, so bei Google zumindest, nach zwei Jahren weggeworfen werden kann, weil es dann praktisch keine Sicherheitsupdates mehr bekommt.

I: Klar auch da sind die Zyklen der Entwicklung sehr viel schneller.

IP: Klar und es geht auch um einen anderen Wert. Es tut schon weh, wenn man ein Smartphone für ein paar 100 € nach zwei bis drei Jahren wegwerfen muss. Das können Sie den Kunden, einem Fahrzeug-Kunden, nicht erklären. Wir unterstützen deine Hardware-Plattform nicht mehr, das heißt du hast keine Navigation bzw. keinen Radio mehr.

I: Das stimmt das würde die Kunden nicht wirklich begeistern bzw. zufriedenstellen.

IP: Wir haben auch immer noch klassische Modelle, die fahren immer noch auf der Straße und kosten auch immer noch viel Geld. Mit diesem Ansatz können wir nicht alles von der Smartphone Industrie übernehmen, dort sind die Hardware Zyklen relativ kurz bzw. die Kunden sind sehr viel schneller bereit, was Neues zu kaufen und werfen das Alte dann weg. Z.B. sie bekommen kein neues Sicherheitsupdate auf dem Android-Smartphone, also kaufen sie sich ein neues Telefon. Bei IBM z.B. haben wir alle zwei Jahre ein neues Handy bekommen. Denn wenn es keine Sicherheitsupdates mehr von Android gab, dann fielen praktisch die Firmen Apps weg und man durfte sich ein Neues bestellen.

I: Das sind ja auch teilweise Tricks, die dann mit Absicht eingebaut werden. Genau. Ja das ist ein sehr interessanter Gedanke die beiden Industrien zu vergleichen und Gemeinsamkeiten und Unterschiede zu sehen.

*Frage 20: Worin sehen Sie die größte Herausforderung, wenn es darum geht traditionelles Qualitätsmanagement mit agilen Arbeitsweisen zu verbinden?*

IP: Bei agilen Arbeitsweisen haben Sie eigentlich per Definition ein Team vor Ort, das dann praktisch zusammenzuarbeiten und das dann praktisch durch persönliche Gespräche sehr viel klärt. Sie brauchen keine oder sehr wenig Dokumentation, da permanente Kommunikation stattfindet in einem überschaubaren Rahmen. Bei komplexen quasi Produkten wie beim Fahrzeug, gibt es dann aber nicht nur ein Team, das für sich ein unabhängiges Produkt entwickelt, sondern es ist eingebettet in ein größeres Umfeld. Das heißt sie haben immer auch Abhängigkeiten zwischen vielen Teams, die deshalb mehr Kommunikation und schriftliche Vereinbarungen benötigt. Und das gibt dann ein Spannungsfeld zwischen agil intern und gewisser Verbindlichkeit nach außen zu anderen Systemen. Das dann richtig zu machen. Wenn man z.B. sagt, ich mache eine Plattform, wo ich möglichst viel an Basisfunktionen anbiete, was in Zukunft auch benutzt werden wird, selbst wenn aktuell noch keine Anwendung dafür gibt. Dann lasse ich die einzelnen Entwicklungsteams darauf und die Sachen zusammenstellen, die Sie brauchen. Und wenn Sie mehr brauchen mach ich die Anforderung auf die nächste Hardware-Generation drauf. Das wäre ja so das Idealbild, welches momentan noch nicht Realität ist. Bei den Entwicklungszyklen im Hardware-Bereich, da müssen sie einfach Qualität sicherstellen in allen Lebenslagen, sprich Temperatur, Hitze, USA, Südafrika, Deutschland und Sibirien. Das Auto muss überall funktionieren. Da braucht es einfach sauber entwickelte Hardware, welche auch extreme klimatische Bedingungen über Jahre hinweg aushält. Wir hatten bei uns aktuell auch Probleme, da war der Temperatursensor nicht für die Temperaturen sauber ausgelegt, das heißt der ging regelmäßig kaputt. Die Klimaanlage ist dann auch ausgefallen, das ist auch kein Spaß im Hochsommer ohne Klimaanlage dazustehen. Und es macht dann auch keinen Spaß im Hochsommer zu testen. Da sagt man auf einmal, ok hätten die anderen mal bisschen besser getestet, dann wäre das wohl nicht passiert. Das ist das Grundproblem zwischen Fahrzeug-Entwicklung, welche einfach Laufzeiten hat und andererseits dieser Smartphone-spezifischen Schnelligkeit von neuer Software und neuen Features. Z.B. Apps draufschmeißen und wieder runter schmeißen. Das sinnvoll unter einen Hut zu bringen, das ist herausfordernd. Man benötigt eben eine Entkopplung und saubere Plattform-Strategien. Das heißt man muss im Fahrzeug eine Plattform-Architektur aufbauen.

## Interview 2

### **Interviewpartner: Qualitätsmanager im Bereich Connected Car mit dem Schwerpunkt End-to-End Testing** (Abkürzung: IP - Interviews Partner, I - Ich)

I: In welchem Bereich sind Sie tätig?

IP: (Also ich bin im Bereich QM/QPD tätig.) Da geht es um das Thema Qualitätsmanagement für digitale Dienste. In dem Bereich bin ich seit einem Jahr. Davor war ich in der IT. (Den Bereich QM/QPD gibt es auch erst seit anderthalb Jahren davor war das im Telematik-Umfeld verordnet, was dann doch eher Hardware lastig ist.) Dieses Herauslösen der digitalen Dienste als eigenen Bereich gibt es in der Form seit anderthalb Jahren. Also in Summe seit 2003 im Automobilbereich (bei Daimler).

I: Welche Stellung besetzen sie in ihrem Bereich?

IP: Ich bin Qualitätsingenieur – das ist die offizielle Stellenbeschreibung. Es ist keine Managementstelle, sondern eine Sachbearbeiter Stelle.

I: Wie lange sind sie bereits im Bereich Qualitätssicherung im Bereich Connectivity Produkte tätig?

IP: Wie gesagt bin ich jetzt seit einem knappen Jahr dabei.

I: Haben sie schon bereits Vorerfahrung im Bereich der Softwareprodukte gemacht?

IP: Ich war davor in einem E-Commerce Projekt für drei Jahre und war dort hauptverantwortlich für das Thema Qualität und stark eingebunden in die Testing-Prozesse, in die Abläufe und in die Planung.

I: Für welche digitalen Produkte im Bereich Connectivity sind sie aktuell tätig?

IP: Also wir teilen uns im Team auf und ich bin jetzt beim Thema In-Car-Office unterwegs. Also alles was sich um die Connected Car Service App (Mercedes-me App) dreht und auch das Thema Personalisierung wird von mir betreut. Hier geht es um das Thema Profile im Fahrzeug sprich Multiprofil-Management, wo man seine eigenen Einstellungen dann eingeben kann und auch mehrere Benutzer im Fahrzeuge haben kann, die sich dann Präferenzen, ob das nun Licht, Radiosender oder Klimaeinstellungen sind, hinterlegen können. Das sind die spezifischen Themen.

I: Wie sind die Bereiche Entwicklung und Qualitätssicherung für Connectivity-Produkte in die Unternehmensorganisation eingegliedert? Sind es eigenständige Funktionsbereiche? Oder ist das Qualitätsmanagement in die Entwicklung integriert?

IP: Grundsätzlich machen wir einen regelmäßigen Testlauf im Moment ist das vier Mal im Jahr zu aktuellen Fahrzeugen der Connected Bandbreite von Connectivity Diensten. Zwischen diesen vier Testphasen teilen wir uns wie eingehend erwähnt auf die unterschiedlichen Dienste auf.

I: Hier habe ich einen Entwicklungszyklus für Software abgebildet. Angefangen von der Anforderungsfeststellung, grob und fein Entwurf, Implementierung, Test und Integration und so weiter. Eben wie man es aus der Literatur kennt. An welchen Stellen entlang dieses Zyklus gibt es denn Schnittstellen zwischen der Entwicklung und dem Qualitätsmanagement?

IP: Also die größte Schnittstelle ist beim Thema Erprobung und Inbetriebnahme. Da gibt es heute ein festgelegtes Quality Gate, in dem alle Dienste durch den Qualitätsbereich getestet und freigegeben werden. Es gibt im Bereich grob- und Feinentwurf eine mehr oder weniger gefestigte Zusammenarbeit im Sinne von Einbindung von Kundenstudien und Kundenfeedback. Außerdem gibt es im Moment noch stärker im Aufbau befindliche Integration im Umfeld Test und Integration. Dort haben wir eine Crowdfunding-Plattform aufgesetzt und wollen als Teil der Testphase das Thema Kundentest oder Crowd-Test anbieten. Ansonsten beim Thema Wartung und Weiterentwicklung sind wir Stand heute stark involviert. Das ist ein Bereich, in dem es noch Herausforderungen gibt. Auch auf Seiten der Entwicklung. Und ja das Thema Außerbetriebnahme, da haben wir zu verschiedenen Diensten eine Meinung, aber da gibt es an dieser Stelle keinen geregelten Prozess.

I: Wenn Sie die Zusammenarbeitsweise betrachten, sehen Sie dann die Schnittstellen zwischen Entwicklung und Qualitätsmanagement als ausreichend für die Sicherstellung der Qualität an?

IP: Nein also Stand heute definitiv nicht. Also besonders in der frühen Phase sind wir oft nicht besonders gut integriert. Dort werden in der Entwicklung sehr häufig Dinge bereits vorbereitet und implementiert. Die Qualitätssicherung ist erst relativ spät involviert und gerade wenn es an konzeptionelle Themen geht sind

wir da zu spät am Prozess dran, um noch rechtzeitig Änderungen einzufordern. Eingebunden sind wir heute und wahrscheinlich auch traditionell eher am Ende, was auch ein bisschen undankbar ist, weil wenn man erst ganz am Schluss das Ergebnis erblickt. Wenn dann etwas nicht passt und man seine Kommentare einbringt, dann schlägt das auch dementsprechend größere Wellen, weil schon entsprechend viel Aufwand in das Thema gesteckt wurde und das Thema somit auch schon fast fertig ist. Das ist dann auch manchmal ein undankbarer Job.

I: Es spricht auch entgegen dem, was man kennt, dass man sagt, man sollte möglichst am Anfang des Entwicklungsprozesses Fehler ausmerzen, da sie später sehr viel mehr Kosten verursachen.

IP: Genau und es sind auch nicht nur Bugs und Defekte drinnen, sondern es sind teilweise einfach konzeptionelle Themen. Da haben wir als Qualitätsmanagement eine relativ breite Sicht auf die digitalen Dienste im Konzern. Und die Entwicklungsingenieure in RD haben eher eine eingeschränkte Sichtbarkeit und können sich auch nur auf ihren Dienst fokussieren. Es wäre da schon sinnvoll, wenn man in der frühen Phase den Blick vom Qualitätsmanagement von links nach rechts bekommen würde. Das würde sicherlich dem ein oder anderen Konzept guttun.

I: Wie werden Kundenanforderungen an das Produkt ermittelt?

IP: Was wir machen... aber da gibt es auch keinen Regelprozess. Was wir im Telematik-Umfeld und jetzt immer mehr sporadisch in den (Connect) Diensten machen, ist frühzeitig Kunden Prototypen und Ideen zu zeigen und auf dieser Basis die Kundenrückmeldungen einzusammeln. Heute in der früheren Phase und nicht kontinuierlich, sondern eher einmalig. Die Königsdisziplin ist es, dass auch umzusetzen, vorzusehen im Konzept, und dann auch nochmal eine Schleife mit dem Kunden zu drehen, zu fragen, ob das wirklich auch das ist, was er sich vorgestellt hat. Da sind wir heute dabei, aber auch nicht im Rahmen eines Regel-Prozesses und auch beim Thema Crowd-Testing, da nutzen wir eine interne Crowd, das bedeutet, dass sind interne Mitarbeiter des Konzerns, welche einigermaßen Themenfremd sein sollen und daher auch eher die Meinung eines Kunden bieten als die eines Experten.

I: Wer steht im Austausch mit dem Kunden ist das Entwicklungs-seitig oder Qualitätsmanagement-seitig?

IP: Also in der frühen Phase findet das gemeinsam statt. Die Organisation der Kundensitzungen findet durch den Qualitätsbereich statt. Die Bereitstellung der Inhalte, Fahrzeuge und auch Tests oder Prototypen wird durch die Entwicklung sichergestellt.

I: Sie haben erwähnt, dass sie schon mit Softwareprodukten und Hardwareprodukten Erfahrungen haben. Wo sehen Sie die größten und bedeutendsten Unterschiede zwischen Qualitätssicherung von Hardwarekomponenten und Connectivity Produkten?

IP: Die größte Herausforderung ist sicherlich, dass ich bei Hardwareprodukten starrere Zeiträume habe. Ich habe weiter voneinander entfernte Iterationsstufen als bei einer Softwareproduktion. Bei einer Softwareentwicklung kann ich schnell integrieren, man kann schnell agieren und einen neuen Stand erzeugen, schneller Mal einen A-B-Test aufbereiten und die Kunden einbinden. Die Hardware-Welt so wie wir sie kennen ist doch eine sehr starre Wasserfall-getriebene Welt. In der Softwareentwicklung, da ist es sicherlich aus Qualitätssicherungs-Sicht notwendig, Ideen zu generieren wie man dieses agile Umfeld, das Software-Umfeld eben, mit agilen Qualitätssicherungsmaßnahmen absichert. Im Vergleich zu der Wasserfall-getriebenen (Fahrzeug-Entwicklung), wo ich auch klare Phasen, Abnahmen und Meilensteine habe, da tun wir uns leichter Freigaben oder Freigabe-Schritte durch eine Qualität einzuplanen als im agilen Software Umfeld. Das ist sicherlich die größte Herausforderung.

I: Aus dem Gespräch kam bereits heraus, dass agile Arbeitsweisen in der Entwicklung angewendet werden, in welcher Form sind denn agile Arbeitsformen bei der Entwicklung von Connectivity Produkten etabliert?

IP: Sehr stark. Nahezu alle Dienste werden sehr nah an Scrum entwickelt. Es finden Sprintplanungen statt, es gibt Entwicklungszyklen, die viel kürzer sind als in der Hardware-Welt. Es werden regelmäßig neue Stände generiert, die zwar nicht sichtbar für den Kunden sind, aber es werden regelmäßig Software Inkremente bereitgestellt, die dann auch theoretisch getestet werden könnten von der Qualität. Allerdings müssen wir uns da aus Qualitätsmanagement-Sicht prozessual anpassen, um auch in diesem kurzzyklischen Testphasen drin zu sein. Aber im Großen und Ganzen was das Thema Connectivity-Dienste oder Connected Car Service App (Mercedes-me) Dienste anbelangt wird zu 80% mit agilen Methoden entwickelt.

I: Im Bereich der agilen Methoden sehen sie da eine Unterstützung der Unternehmenskultur?

IP: Ich verstehe die Frage nicht, was meinen Sie damit.

I: Agile Arbeitsweise werden hauptsächlich in der Entwicklung angewendet und hauptsächlich in der IT angewendet. Beispielsweise gibt es dort nicht diese klassischen, hierarchischen Strukturen wie im Automobilbereich. Damit sind sie mit sehr viel flacheren Hierarchien verbunden. Oftmals werden im Team Rollen verteilt ohne eine direkte Führungskraft zu haben.

IP: Also die gibt es schon. Also man versucht es im Moment zu verheiraten. Auf der einen Seite die hierarchische Aufstellung, die wir nach wie vor haben und es gibt Ansätze, das war auch in meinem alten Umfeld so, dass man sich in Swarms zusammenschließt und dort entsprechend Ergebnisse generiert. Das halte ich auch für sinnvoll. Das ist aber eher heute im IT, After Sales und Marketing-Umfeld der Fall. Da RD eine sehr klassische, Hierarchie-getriebene Organisation ist, ist das heute nicht so. Dort versucht man noch immer die klassischen agilen Arbeitsweisen zu kombinieren oder unterzubringen in einem hierarchischen Aufbau. Das funktioniert je nach Bereich mehr oder weniger gut.

I: Frage zwölf hat sich geklärt.

I: Welche Techniken und Methoden kommen dabei zum Einsatz? Sie haben bereits Crowdfunding genannt. Gibt es noch weitere Test-Maßnahmen, wie automatisierte Tests, die angewendet werden?

IP: Ja der Entwicklungsbereich insbesondere der Bereich, der sich mit Software beschäftigt, setzt sehr stark auf Testautomatisierung. Insbesondere die Kollegen, die sich um die Apps kümmern, nutzen Testautomatisierung zur Qualitätssicherung vorab. Die Qualitätssicherung selbst, also wir versuchen das nicht zu tun, weil wir würden das natürlich einfordern, aber wir stellen uns im Moment so auf, dass wir die Kundenperspektive einnehmen und uns praktisch heute in die Kundenschuhe zwingen und nicht so stark in automatisierte Tests einsteigen, sondern wirklich einen naiven Kundenblick auf die Themen zu bekommen. Weg von den Testergebnissen und den ganzen Test und Defekt Management vorab. Natürlich erwarten wir, dass die Dinge getestet sind und funktionieren wir sind da aber erst einen Schritt später dran, aber Testautomatisierung findet da statt. Das ganze Thema Test- und Defekt Tracking ist leider verteilt auf mehrere Tools, also die Telematiksysteme nutzen da andere Tools als die klassische Connectivity-Welt. Man versucht aber alles jetzt in die Gira Confluence Ecke zu ziehen um eine gemeinsame Planung mit der Hardware-lastigeren Welt hinzubekommen. Und auch die Features und das Defekt Management über diese Tools entsprechend zu planen. Da sind wir aber noch nicht da, wo wir hinwollen. Ansonsten nutzen wir Crowdfunding Stand heute vorrangig beim Thema Usability-Test. Beim Thema funktionales Testing oder System-Testing haben wir Crowdfunding noch nicht im Einsatz, da es komplex ist, wenn man mit einem Fahrzeug testet. Da braucht man immer eine Fahrzeug-Benutzer-Relation und das ist in der Vorbereitung immer relativ aufwändig. Gerade wenn man eine größere Gruppe hat, die damit testen will ist es sehr aufwändig. Von daher Crowdfunding Stand heute nur im Bereich Usability auf Basis von Klickdummies oder Prototypen.

I: Wo sehen Sie im Entwicklungsprozess die größten Herausforderungen für qualitätssichernde Eingriffe seitens des Qualitätsmanagements.

IP: Das ist definitiv in der frühen Phase in der Ideen- und Konzeptphase unseres Erachtens nach ist das Thema Kundenfeedback heute zu wenig berücksichtigt. Ich denke, da teilen wir uns mit anderen Firmen das Problem. Man ist heute sehr stark durch das Thema Geschwindigkeit, welches im Rahmen der Digitalisierung überall propagiert wird, gezwungen als Entwickler schnelle Ergebnisse zu erzeugen. Jede Schleife, die ich da mit einem Kunden vorab noch drehen muss, ist natürlich Aufwand und kostet Zeit. Unserer Meinung nach, ist es Zeit die man sich hinterher spart. An dieser Stelle sehen wir die Gewichtung an der falschen Stelle. Wir würden gerne viel früher das Thema Kundenfeedback und das Thema Konzept-Review verankern. Als dann hinterher eine Lösung, die dann nicht den Kundenwünschen entspricht, hinterher rennen zu müssen und diese noch einmal nachbessern zu müssen. Also definitiv frühe Phase.

I: Was hat das für Folgen, wenn das Kundenfeedback in den frühen Phasen nicht ausreichend berücksichtigt wird?

IP: Das hat zur Folge das dann am Ende eine Lösung herauskommt, bei der wir in der Nutzung sehen, dass der Dienst von den Kunden dann nicht so angenommen wird. Das sieht man dann an Verkaufszahlen, das sieht man an Analytics-Zahlen. Das muss auch ein Grund haben. Wir sehen an dieser Stelle auch das Thema, dieses breite Bild, das ich bereits erwähnt habe, herzustellen. Es ist wichtig, dass wir aus der Qualität bei dem digitalen Dienst, der da gerade entwickelt wird, auch nach links und rechts sehen. Damit könnte man vielmehr Synergien mit Diensten, die es schon gibt, erzeugen. Das ist sicherlich ein Thema, was dann hinterher für den Kunden und seine Dienste sehr abgetrennt aussieht und es macht kein integriertes Bild. Das klassische Bild, was man von Apple kennt, dass alles mit allem super zusammen funktioniert und hoch integriert ist. Dieses Bild hinterlassen wir heute nicht bei jedem Kunden, aber das könnten wir, wenn wir A frühzeitig das Kundenfeedback einfließen lassen würden und B auch Erfahrungswerte aus dem Konzern nutzen würden und die Dienste somit integrierter gestalten würden.

I: Frage 16 hat ist geklärt.

I: Welche Mitarbeiterkompetenzen sind in einem agilen Team vorhanden? Gibt es da verschiedene Rollen?

IP: Das ist schwer zu sagen, weil wir in der Entwicklung auch sehr viele Themen fremdvergeben haben. Meistens ist es so, dass der Product Owner im Automobilunternehmen (bei der Daimler AG) sitzt. Zumindest in diesen Diensten, in denen ich tiefergehend integriert bin. Die Entwicklungsteams, das Thema Testing und der Scrum Master sind meistens beim Lieferanten. Das funktioniert einigermaßen gut in den Themen, in denen ich unterwegs bin. Sobald allerdings dann mehrere Lieferanten (auf der Daimler Seite) und mehrere Organisationen involviert sind, hat der Product Owner die Themen zusammen zu fassen und zu priorisieren. Das klappt mehr oder weniger gut, sag ich mal. Ebenfalls eine große Herausforderung ist das Thema, wie bringe ich dann die Dienste, die noch sehr Hardware-belastet sind, zusammen. Also die Software, die im Fahrzeug entsteht, die wird ein bisschen so entwickelt, wie die digitalen Dienste, die jetzt noch oben drauf gebaut werden. Von daher ist es herausfordernd für den Product Owner diesen beiden unabhängigen Zeitleisten zusammenzubringen und auch die Softwareinkremente, die da erstellt werden, dann im Hinblick auf das Thema Testing und Abnahme zeitlich zusammenzubringen. Das quasi alle Anteile an so einem digitalen Dienst, ob es Fahrzeuganteile sind oder ob es Backend-Anteile oder Frontend-Anteile sind, die zum richtigen Zeitpunkt zusammen zu bringen. Diese Steuerung ist da sicherlich eine der größten Herausforderungen. Ich weiß nicht, ob das die Frage hundertprozentig beantwortet falls nicht einfach noch mal nachfragen.

I: Doch doch das ist sehr interessant gewesen, weil ich nicht wusste, dass Scrum Master vom Lieferanten gestellt werden. (Oder von Partnerunternehmen wie z.B. Mercedes-Benz.io.) Ich dachte, dass das in einer Abteilung verankert ist.

IP: Bestenfalls sind das wirklich Swarms, wo man sich auch fest zu diesem Zusammenarbeitsmodell committed hat und da funktioniert es dann auch ganz gut. Schlechtesten falls ist das noch eine sehr starke Auftraggeber-Auftragnehmer-Beziehung, bei der dann tatsächlich der Product Owner im Automobilunternehmen (auf der Daimler Seite) sitzt. Wenn man da zuhört, ist man wieder in dem Thema Lasten- und Pflichtenheft-Beziehung sprich, ich sag dir, was du machen sollst und ich schau mir danach das Ergebnis an. Wie gesagt im Marketing- und Sales-Bereich machen wir das schon. Die Ansätze würden uns auch im Entwicklungsbereich guttun.

I: In welcher Form werden Mitarbeiter im Team oder in der Abteilung gefördert bzw. weitergebildet?

IP: Wir fokussieren uns auf die Schritte im Entwicklungsprozess, wo wir tatsächlich den größten Handlungsbedarf sehen. Schauen uns dort verwendete Methoden an und schulen die Mitarbeiter gezielt für diese Methoden. Insbesondere das Thema frühe Phase, wie mache ich einen Usability Test, wie definiere ich User-Persona, wie mache ich beim Crowdfunding, Think-aloud Tests, was ist überhaupt Think-aloud. Solche Sachen, die man an der Stelle in dieser Phase gebrauchen kann, machen wir schon. Da gab es jetzt auch eine erste größere Runde, wo man ein Basistraining angeboten hat. Darüber hinaus macht man auch tiefergehende Trainings z. B. habe ich persönlich schon zwei Zertifizierung zum Thema Usability und User-Experience gemacht. Das eine war eine Grundlagenschulung und das andere war eine Advanced im Bereich User-Requirement-Training. Das Thema, wie berücksichtige ich Usability-Aspekte im User-Requirement-Engineering und das Thema Testing sind sicherlich Themen, welche noch weiter ausbaufähig sind. Also Usability-Testing ist etwas, das auch noch ausbaufähig ist, aber ich glaube die Themen sind da bekannt. Es gab Grundlagenschulungen und in den einzelnen Themenfeldern je nachdem um welche Mitarbeiter es sich handelt, wurden nochmal Schwerpunkte gesetzt durch das Angebot von tiefergehenden Trainings. Diese werden dann über eine interne Schulungsorganisation durchgeführt.

I: Welche Methoden und Werkzeuge werden zur ständigen Verbesserung eingesetzt?

IP: Wir machen vier Mal unsere größeren Testphasen, bei denen es eine Vorbereitungsphase und eine Testphase gibt. Hinterher gibt es dann immer noch eine Art Lessons-learned. Die Ergebnisse versucht man danach auf die nächsten Testphasen anzuwenden. Darüber hinaus, wenn Sie nach Methoden und Werkzeugen fragen, wüsste ich nichts. Das heißt nicht, dass es nichts zu verbessern gibt. Klar wollen wir uns verbessern und an verschiedenen Stellen effizienter werden. Wir müssen auch effizienter werden, weil die Welt sich auch relativ schnell dreht und wir da mitgehen müssen, ob das nun richtig ist oder nicht ist die Frage. Das Thema Qualität und Geschwindigkeit bekommt man immer plastisch gesagt, muss sich nicht ausschließen, aber ich sage das Risiko, dass es sich ausschließt, ist relativ hoch. Insbesondere wenn die Geschwindigkeit das Hauptziel eines Projektes ist, dann wird es als Qualitätsmensch schwer. Das ziehen wir tatsächlich dann auch immer als Lessons-learned raus, wenn wir aus Sitzungen mit der Entwicklung rauskommen. Da haben wir das Gefühl, dass wir ein bisschen bremsen müssen, können es aber nicht, da die Geschwindigkeit vom Management vorgegeben wird. Das ist ein ziemlich anstrengender Spagat.

I: Worin sehen Sie die größte Herausforderung, wenn es darum geht traditionelles Qualitätsmanagement mit agilen Arbeitsweisen zu verbinden?

IP: Ich denke, dass Thema kam in einigen meine Antworten auch schon heraus. Das Thema klassisches oder traditionelles Qualitätsmanagement gerade in der Automobilindustrie, das hatten sie auch schon eingangs erwähnt, ist sehr Hardware-lastig und sehr stark gebunden an klassische Phasen-orientierte Prozesse auch Wasserfall-Prozesse und diese Welt zu verheiraten mit einer Software Welt die viel schnell-drehender ist mit ihren agilen Arbeitsweisen, ist sicherlich die größte Herausforderung. Also zum einen das Thema die Vorgabe alles muss schnell entwickelt und schnell umgesetzt werden und da eine Qualität reinzubringen. Und die andere Herausforderung ist, wenn ich integrierte Systeme habe, wo ich sowohl eine Fahrzeugkomponente benötige, welche nach einem anderen Entwicklungsprozess und Zyklus entwickelt wird, und eine Softwarekomponente, die wesentlich schneller fertig ist, diese beiden Zeitleisten und Entwicklungszyklen zusammenzubringen, Meilensteine aufzusetzen oder Testphasen und Qualitätsabnahmen einzubringen. Das ist sicherlich eine sehr große Herausforderung. Wir sehen das auch regelmäßig, dass durch diese schnell-drehende Software Welt regelmäßig die Hardware Welt abgehängt wird. Dementsprechend ist die Akzeptanz für die Software Themen bei den Hardware Kollegen unterdurchschnittlich. Für die dreht sich das an dieser Stelle viel zu schnell die kennen ihre Prozesse und glauben auch, dass es wichtig ist, dass alles phasenorientiert nach einem starren Entwicklungsprozess abläuft. Die können da noch nicht so richtig mit umgehen, dass die Software Welt schneller dreht und dass da Dinge schneller entwickelt werden. Das Ganze bietet auch Chancen, weil in dem Moment, in dem ich etwas schneller entwickle, kann ich im Fall eines Fehlers auch schneller etwas fixen. Wenn ich aus Sicht der Hardware Welt denke, wenn wir da Probleme mit irgendwelchen Fahrzeugen haben dann muss ich ja nicht sagen was für Maßnahmen dann eingeleitet werden. Rückrufaktionen, Aktionierung, Fahrzeuge, die im Werk verbleiben bis die Fehler behoben werden, Teile, die nachgebessert werden, all das ist natürlich sehr viel aufwendiger als in der Software Welt, wo ich viel agiler und schneller Themen fixen kann. Das ist sicher etwas, dass man als Qualitätsmanagement auch berücksichtigen muss. Also wenn ich überlege, will ich jetzt wirklich einen kompletten Entwicklungsprozess aufhalten, nur weil an der Software Ecke noch Fehler drin sind, sage ich, man akzeptiert an der ein oder anderen Stelle auch noch einen Fehler. Man setzt das Produkt live vor Kunde, hat aber gleichzeitig schon geplant das Thema innerhalb der nächsten zwei oder drei Wochen über ein Software Update zu beheben, dass dann remote zu Verfügung gestellt wird. Ich kenne das aus dem alten Umfeld, auch wir hatten das im E-Commerce Umfeld. Auch so haben die einen Bug-freies Release gehabt und wir haben auch nie ein Release freigegeben wo keine Bugs drin waren das waren immer Themen wo wir dann am Schluss auch noch akzeptiert haben. Wir akzeptieren das und fixen das dann auch innerhalb der nächsten zwei bis drei Wochen. Das ist auch eine Chance und da muss auch das Qualitätsmanagement an der ein oder anderen Stelle ein Auge zudrücken. Sagen okay wir sind jetzt noch nicht ganz fehlerfrei aber wir können das zumindest den Kunden so zumuten, klammer auf aber haben schon einen festen Fixtermin geplant.

### Interview 3

**Interviewpartner: Lead-Link der Scrum Master eines Software Unternehmens, welches Software- und Connectivity Produkte für die Automobilindustrie entwickelt (Abkürzung: IP - Interviews Partner, I - Ich)**

I: Welcher Branche ist das Unternehmen zugeordnet und in welchem Bereich bist du tätig?

IP: Also meine Rolle ist Scrum Master für verschiedene Produkte hier bei uns im Haus. Wir sind ein Software-Unternehmen, das digitale Produkte für den Automobilbereich [REDACTED] betreut mit einem sehr breiten Spektrum. Ich bin gleichzeitig in der Rolle als Lead-Link der Scrum Circles aktiv und ein Stück weit das Thema Enablement in unserer Disziplin voranzutreiben. Das heißt wir haben hier im Unternehmen [REDACTED] auch 15 Scrum Master, die das Team coachen, und die leite ich ein Stück weit an, welche Methoden nutzen wir, wie arbeiten unsere Teams zusammen. Um eben Gemeinsamkeiten zu identifizieren, Best Practices zu identifizieren und Synergien zwischen den Produkten aufzudecken.

I: Wie lange bist du bereits für die Entwicklung von Softwareprodukten tätig?

IP: In der Softwarebranche bin ich jetzt seit zehn Jahren. Vorher bei Mittelständlern und Beratungen und jetzt hier im Unternehmen [REDACTED] seit drei Jahren.

I: Was hast du denn studiert bzw. was hast du denn für einen Background?

IP: Ich habe ursprünglich angefangen mit einer Ausbildung im IT-Bereich als IT-Systemkaufmann. Ich habe deswegen auch relativ lang in technischen Themen gearbeitet. Bis es dann irgendwann in die Projektleitung ging und Projektmanagement dazu kam und so weiter, dann über die Beratung auch bei Unternehmensname [REDACTED] im Projektmanagement angefangen. Und hab dann den Schritt gemacht und als Scrum Master angefangen als das Thema Agilität immer stärker bei uns ins Rollen kam und das hat mich unheimlich interessiert und deswegen habe ich mich dann ein Stück weit weg von der Technik hin zu Prozessen bewegt. Sprich wie arbeiten wir zusammen und da kommt auch das Thema zum QM und QR.

I: Welche Produkte begleitest du denn konkret in der Entwicklung?

IP: Die Produkte, die ich momentan betreue, sind stark im Bereich E-Commerce angesiedelt. Das heißt alles was das Automobilunternehmen [REDACTED] online verkauft von Schlüsselanhängern, T-Shirts über Fahrzeug-Trainings und wirklich auch Fahrzeuge, gebraucht und neu. Und dort im speziellen die technische Plattform, auf der mehr oder weniger alle Online-Sales-Aktivitäten bündelt.

I: Wie sind denn die Bereiche Entwicklung und Qualitätssicherung in die Unternehmensorganisation eingegliedert? Sind das eigene Funktionsbereiche oder ist das integriert?

IP: Bei uns ist es vergleichbar mit dem Circle der Scrum Master. Wir sind innerhalb des Unternehmens [REDACTED] also nicht in der Produktstruktur, sondern in der Organisationsstruktur in Circle, in holokratische Circle organisiert und diese sind nach Disziplinen aufgestellt. Das heißt, es gibt den Circle in dem Agilität bearbeitet wird also das Thema Agility, da sind alle Scrum Master und Product Owner drin. Es gibt einmal das Thema für Design und für Konzept, das ist eine Disziplin. Dann gibt es einen eigenen Bereich für das Thema Produkt-Support im Endeffekt dort liegen unter anderem Qualitätsmanagement drin, aber auch noch andere Themen, die sehr eng mit dem Life Cycle eines Produkts zusammenhängen. Also z. B. das Release Management und Content. Und das Thema Entwicklung an sich umfasst im Endeffekt alle technischen Spezialisten, angefangen von Backend-Entwicklung bis Frontend-Entwicklung, Architektur, IT-Consulting. Die dort allerdings innerhalb des Entwicklungsbereichs nochmal unterteilt sind in die einzelnen Spezialisten. Also eine Einheit für das Thema Frontend, eine Einheit für das Thema Backend, da gibt es nochmal verschiedene Spezifika für die verschiedenen Backend-Standards. Genau das sind so die Organisationseinheiten an der Stelle. Also um die Frage zu beantworten, Quality Assurance, wie es bei uns heißt, ist bei uns im Endeffekt ein eigener Bereich, der zumindest organisatorisch nicht an der Entwicklung hängt.

I: Ich habe hier auf dem Fragenkatalog einen traditionellen Software-Entwicklungszyklus abgebildet. Ich werde mich beim Fragen öfters darauf beziehen und da ist es ganz gut, wenn man das nochmal vor Augen hat. Eben von der Anforderungsfestlegung bis zur Außerbetriebnahme, wie man es eben auch aus der Literatur kennt. Teilweise kann die Softwareentwicklung ja stark abweichend von der traditionellen Weise, wie hier im Schaubild dargestellt. Wir haben es schon oft angesprochen die agilen Arbeitsweisen werden in der Softwareindustrie ja schon seit Jahren eingesetzt. In welcher Form oder in welchem Umfang sind denn agile Arbeitsweisen bei euch in der Entwicklung etabliert?

IP: Also ich kann relativ zuversichtlich sagen, dass eigentlich neun von zehn Produkten bei uns mit agilen Methoden arbeiten. Es gibt immer eine Ausnahme, bei der eher klassisch gearbeitet wird. Die agilen Teams arbeiten mit den unterschiedlichsten Methoden und Scrum ist ein Stück weit das verbreitetste und einfachste zum Adaptieren. Es gibt Teams die Arbeiten mit einem Kanban-Modus aufgrund von verschiedenen Rahmenbedingungen heraus. Und es gibt Teams, die in skalierten agilen Frameworks arbeiten. Z. B. Nexus wo dann mehrere Teams an einem Produkt zusammenarbeiten. Das hat dann Hintergründe, wie das man möchte schneller oder mehr liefern und einfach deswegen auch skalieren muss. Wobei das auch immer ein Stück weit ein ungewünschtes Phänomen ist in den agilen Arbeitsweisen, dass man mehr liefern und schneller liefern möchte, weil eigentlich durch das Thema Skalierung auch schnell Ineffizienzen auftreten, die man einfach in einem einzeln laufenden Team gar nicht beobachten kann.

I: Nochmal zu diesem Thema Nexus: Das ist also auch eine agile Arbeitsweise, die aber eben aus einem Projekt mehrere Teilprojekte macht und die wieder zusammenlaufen lässt?

IP: Nexus ist mehr oder weniger eine Skalierungsform von Scrum. Es gibt noch andere Skalierungs-Frameworks wie z. B. Lap oder Fate. Wir haben aber für uns entschieden, dass Nexus das Ideal ist, weil es relativ wenig Overhead produziert. Wir haben in dem Produkt konkret vier Teams, die komplett cross-funktional besetzt sind, also vier Teams à ca. acht Personen. Und die Nexus Schicht bringt im Endeffekt das, was man einfach braucht, wenn man vier Teams am gleichen Produkt am gleichen Quellcode arbeiten lässt. Die Integrationschicht im Endeffekt. Also die Nexus-Integrationschicht kümmert sich darum, dass am Ende einer Iteration ein integriertes Inkrement produziert wird. Dieses Management der Integration macht mehr oder weniger dieses Team.

I: Es wurde schon teilweise erwähnt, welche Vorgehensmodelle bzw. Projektmanagementmethode wird bei euch angewendet. Trotzdem nochmal auf die Frage zurückzukommen, aus welches Mitarbeiterkompetenzen setzt sich denn so ein Projektteam zusammen? Gibt es da klare Rollenverteilungen?

IP: Also unsere klassischen Scrum Teams und das bildet die Mehrzahl hier, haben im Endeffekt drei Kern-Rollen. Einmal der Product Owner, das ist ein Stück weit der Proxi des Kunden und derjenige, welcher die Produktdivision erlebbar machen soll für das Team. Dann der Scrum Master, der das Thema Prozesse und Framework owned. Das heißt er schaut, dass die Art und Weise, wie das Team arbeitet und auf Basis welcher Werte und welcher Prozesse und Strukturen, ein Stück weit dem Grundsatz und der Idee entspricht. Und die dritte Rolle ist im weitesten Sinne erstmal das Scrum Dev-Team. Und jetzt gibt es natürlich abhängig davon was für ein Produkt ich entwickeln möchte, extreme Spezialisierung. Also es kann sein, dass ich ein Dev-Team habe, was nur aus Frontend-Entwicklern besteht, weil ich eben nur eine Applikation baue, die auf schon bestehenden Backend-Systemen aufbaut. Also ich habe irgendwelche Backend-Systeme mit Fahrzeugdaten und möchte diese Fahrzeugdaten in einer User-Layer visualisieren, also mache ich ein Team aus reinen Frontend-Entwicklern. Das wäre ein Beispiel. Eher ein unüblicher Fall, der Regelfall ist eher, dass man das Ganze schon cross-funktional aufsetzt. Das heißt ein Team ist so unabhängig, dass es keine Abhängigkeiten zu anderen Teams hat und selbstständig Inkremente entwickeln kann. Das heißt ich habe Front-Entwickler, die die Präsentationsschicht Richtung User bauen, ich habe Backend Entwickler, die hintendran für die Datenversorgung zuständig sind. Idealerweise hat man dann noch einen UX-Designer, der sich um das Design und Konzept kümmert. Also der das, was die Frontend-Entwickler entwickeln aus User-Sicht optimiert. Dann gibt es einen fest zugeordneten QR Kollegen im Team. Da gibt es auch verschiedene Ausprägungen vom Testmanager bis zum Quality-Schulungs-Engineer, der auch sehr stark Testautomatisierung und auch sehr auf Detail-Ebene schon unterwegs ist. In manchen Teams trifft man auch auf beide Formen. Und dann gibt es noch Spezialisten-Rollen, wenn die Produkte komplexer werden, wie z. B. IT-Consultants, die wirklich auch technische Konzepte ein Stück weit tiefer ausarbeiten müssen oder Architekten, die auch wenn es darum geht, welche Drittsysteme spielen denn da zusammen gewisse Grundlagen, aber vielleicht auch bei der Initialisierung von so einem Projekt machen müssen. Du hast auch noch gefragt nach Lieferanten. Also mittlerweile machen wir einen Großteil der Wertschöpfung intern. Wir wurden ja relativ stark skaliert in den letzten drei Jahren. Vor drei Jahren waren wir noch bei über 90% externer Wertschöpfung. Ich kann dir nicht die ganz aktuelle Zahl sagen, aber mittlerweile sind es mehr als deutlich über 50% interne Wertschöpfung. Aber durchaus auch noch externe Gewerke.

I: Und die stellen dann gewisse Software Parts zu Verfügung oder liefern richtig?

IP: Genau.

I: Zum ersten Teil dieses Software-Lebenszyklus der Anforderungsfeststellung, wie werden denn die Anforderung des Kunden an das Produkt ermittelt?

IP: Also idealerweise gibt es erstmal eine Problemstellung. Also wir möchten irgendein Problem lösen von einem Kunden der z. B. sich ein Fahrzeug konfigurieren möchte. Um dieses Problem zu lösen, gibt es dann unterschiedliche Konstellationen. Indem man erstmal eine Produktvision entwickelt, das heißt der Product

Owner ist dann schon entweder mit den Business Units in Kontakt oder mit Regionen, um dann die Produktvision ein Stück weit zu übersetzen und zu sagen, hey okay wo soll das denn hingehen. Und dann abgeleitet von dieser Produktversion einen ersten MVP zu entwickeln. Das heißt was wäre denn eine Grundfunktionalität, mit der man das Ganze dann testen kann. Da gibt es dann Ausbau-Varianten von Builds. Also entweder man macht eine klassische prototypische Implementierung, die dann wirklich schon innerhalb von ein bis zwei Wochen mit dem Kunden getestet werden kann. Oder den Fall, den wir eigentlich häufiger haben, dass wir dann auch schon sehr schnell einen sinnvollen MVP-Status festlegen und sagen, ok das ist ein sinnvoller MVP. Fragen dann, was bedeutet das jetzt im Detail, und brechen dann einfach diesen MVP im Sinne von Anforderungen auf kleine Bausteine herunter. Das ist im Endeffekt die Aufgabe des Product Owners, der da auch abhängig von seinem persönlichen Skill-Set Unterstützung bekommen kann von z. B. einem IT-Consultant oder Business Consultant, um einfach diese Beschreibung von Anforderungen in Business Spezifikationen auch umzusetzen.

I: Zu welcher Zeit findet denn diese Anforderungsaufnahme statt? Man kennt es von agilen Arbeitsweisen, dass so ein Produkt auch mehr oder weniger wächst durch den Dialog mit dem Kunden wohingegen das traditionell eher einmalig stattfindet. Wie ist das denn bei euch?

IP: Man versucht normalerweise schon einen MVP so zuzuschneiden, dass man in der Scope relativ fix ist. Dieser Scope lässt sich dann meistens am Anfang von einem Projekt nicht zu 100% direkt in kleine Inkremente zerlegen, die dann auch umsetzbar sind. Sondern das wird zu guten Zwei-Drittel sehr klar schon beschrieben. Und im Entwicklungsprozess selbst durch die Iterationen hinweg, wird dann klarer, wie das letzte Drittel aussehen soll. Da wird auch die Anforderung nochmal ein bisschen klarer, um dieses Drittel dann auch noch zu leisten. Das ist so die Art und Weise. Oft kommen dann über diesen Zeitraum auch Dinge von außen, auf die man reagieren muss. Das heißt z. B. das irgendwelche Drittsysteme, mit denen ich geplant habe, gewechselt werden müssen, da ändert sich durchaus mal eine Schnittstelle. Oder das auch erste User-Tests, die ich in den Iterationen selbst drin habe, ergeben haben, dass ich z. B. mein User-Interface umbauen muss, weil der User es nicht versteht. Das heißt auf dem Weg der Entwicklung zu einem MVP-GoLive hin, kommt es durchaus auch in regelmäßigen Abständen zu Scope-Veränderungen bzw. konzeptionellen Veränderungen kommen.

I: Wie verläuft die Zusammenarbeit hinsichtlich Integration von Software-Komponenten, die jetzt hauptsächlich von euch erstellt werden, in die Hardware-Komponenten, welche dann beim Kunden produziert werden, in diesem Fall bei den OEMs?

IP: Da bin ich vielleicht nicht zu 100% der ideale Ansprechpartner hierzu. [REDACTED]

[REDACTED] Und wenn wir dann mehr oder weniger auch das Thema Online Sales machen, das heißt die Dienste auch online kaufbar machen. Dann orientieren wir uns natürlich, was unsere Softwareentwicklung angeht, an den Release-Zyklen der Connect-Services. [REDACTED]

[REDACTED] Heißt aber für dieses konkrete Team, welches Richtung Connect-Services arbeitet, dass für die nur diese vier Release-Zyklen im Jahr relevant sind. Und sie auch auf diese Meilensteine dann hingearbeitet, weil dann eben die komplette Testphase losläuft. Von dem Dienst zu kaufen über den Dienst zu aktivieren über den Dienst im Fahrzeug auch zu testen, das ist eine relativ lange Kette von QM Prozessen, die dann auch durchlaufen wird. Das heißt das Team selbst arbeitet da agil im Entwicklungsmodus, sie arbeiten aber auch auf diese Meilensteine hin. Das heißt sie arbeiten agil bis zum Meilenstein X und dann geht es weiter bis zum nächsten Meilenstein. Und deshalb sind die aufgrund dieser sehr, sehr langfristigen Planung dann auch schon mehr oder weniger ein bis zwei Quartale voraus. Also all das, was sie gerade entwickeln wird dann mehr oder weniger erstmal in den "Kühlschrank" gelegt und sobald dann die Fahrzeug-Entwicklung auch soweit ist, geht es dann in den nächsten Schritt.

I: Das heißt es gibt praktisch schon in dem Automobil [REDACTED] eine Art Plattform mit verschiedenen Schnittstellen und darauf werden die Dienste dann praktisch entwickelt und können dann dort angedockt werden? Also die Schnittstellen stehen schon zur Verfügung und man muss dann nicht immer warten bis ein neues Modell rauskommt?

IP: Genau.

I: Wie stellt ihr denn in Unternehmen die Qualität von Software sicher? Gibt es da irgendwelche Methoden und Prozesse, Qualität sicherzustellen?

IP: Das Thema Qualität ist im Speziellen bei allen Teams und Produkten auch ein Stück weit immer wieder großes Diskussionsthema. Also ich hatte ja vorher schon ganz kurz erwähnt, wir haben quasi einen Quality Assurance Circle, wo alle unsere Kollegen drin sind, um sich einen fachlichen Austausch auch in ihrer Disziplin zu ermöglichen. In den Produktteams selbst ist dann normalerweise ein QA-Spezialist drin von-bis. Es kann sein, dass er sehr technisch bis hin zu einem, der eher Quality managed, der hat dann die Aufgabe ein Stück weit die Aktivitäten zu koordinieren und im Endeffekt sich auch darum zu kümmern, dass die Testpyramide eingehalten wird. Also das heißt, dass die Entwickler ihre Unit-Tests schreiben. Aus der reinen Software-Ansicht. Das heißt, dass ich während dem Build einer Software schon feststellen kann, ob die Unit-Test Coverage erreicht ist und was für Fehler es da gibt. Dann in einer zweiten Stufe der Testpyramide im Endeffekt die Aufgabe Integrationstest zu machen. Das heißt das Zusammenspiel mit allen Komponenten des Produktes Frontend, Backend, gegebenenfalls Drittsystem die Integration sicherzustellen. Und dann auf der Endstufe den End-to-End-Test aus einer Sicht. Das heißt vom ersten User Kontakt bis zur letzten Interaktion die gesamte Software dann auch End-to-End durchtesten. Das ist die Teststruktur, die man als Ziel vorgibt mehr oder weniger im Softwareumfeld. Was unsere einzelnen QA-Kollegen in den einzelnen Teams machen, kann durchaus unterschiedlich sein. Ich würde mal sagen, unsere Teams selbst sind sehr Qualitäts-bewusst. Das heißt unsere Entwickler machen auch schon ein Stück weit die Qualitätssicherung, in dem sie sich gegenseitig Reviewen, oder beim Pair Programming auch schon sich gegenseitig Reviewen. Die sind auch eigentlich dafür verantwortlich, dass Unit-Tests geschrieben werden. Also die schreiben die Unit-Tests, auf denen dann quasi die erste Stufe der Automatisierung stattfinden kann. Das ist mal etwas ausführlicher, wie wir das Thema hier leben.

I: Das heißt es gibt nicht diese ganz konkreten, vordefinierten Schnittstellen zwischen Produktentwicklung und Qualitätssicherung, sondern das ist aufgrund dieses Projekt-Charakters, dass die da praktisch in ständigem Austausch miteinander stehen?

IP: Ja genau also beispielsweise wir haben jetzt hier eine Plattform auf der acht bis zehn Entwicklungsteams arbeiten und jedes dieser acht bis zehn Teams hat einen QA-Kollegen an Bord, der rein aus der Team-Perspektive auf das Thema Qualität schaut. Aber dadurch, dass alle auf der gleichen Plattform unterwegs sind, stimmen die sich auch intern untereinander ab. Ok es geht zusammen auf das Release XY zu. Es gibt gemeinsame Regressionen es gibt gemeinsame Smoke-Test, die durchgeführt werden müssen auf den verschiedenen Umgebungen von Integration bis Produktion. Um einfach Fehler auszuschließen oder auch die Teams ein Stück weit beeinflussen zu können auf dieser Plattform, genau solche Dinge auszuschließen.

I: Die Fragen 11 und 12 haben wir mit der Frage zehn geklärt.

I: Was hältst du für notwendig oder was sind Bedingungen an die Unternehmenskultur, die aus deiner Sicht förderlich sind für agile Arbeitsweisen, um diese auch erfolgreich zu etablieren? Also man kennt es aus der Automobilindustrie, da hat man sehr steile Hierarchien, das widerspricht ja auch diesem agilen Konzept, wohingegen bei IT Unternehmen das gänzlich anders aussieht.

IP: Also da muss man auch sagen, dass wir in einer sehr komfortablen Situation bei uns sind. Da wir quasi eigentlich kein klassischer Supplier sind, sondern eine 100%-Tochter. [REDACTED] Damit haben wir ein Stück weit ein anderes Standing plus wir haben von unserer Geschäftsführung [REDACTED] auch das Commitment, genau in diesen agilen Konstrukten zu arbeiten. Was hinzu kommt und was die Kultur ein Stück weit unterstützt, wir sind auch als Organisation sehr agil aufgestellt, das heißt wir haben ein Organisations-Framework, der nennt sich Holacracy, welcher die Zusammenarbeit in sogenannte Circle modelliert und eben nicht über Hierarchien. In meinem Fall z. B. owne ich den Circle der Scrum Master und gebe da auch Ideen, Motivation und Leitfaden mit rein, aber ich bin von keinem der Scrum Master Führungskraft im Sinne von Jahresgesprächen, Gehalt und sowas. Das ist da strikt getrennt. Der klassische Manager ist in verschiedene Rollen aufgesplittet. Und durch das Commitment vom Topmanagement, also wir sitzen direkt unterm Bereichsleiter [REDACTED] und wir sind 250 Leute, macht das vieles schon sehr viel einfacher. Dadurch, dass wir auch von einem Start-Up von damals zehn oder 12 Menschen sehr schnell groß geworden sind, hat das natürlich auch sehr starken Einfluss auf die Kultur. Das heißt wir haben auch sehr viele Menschen, die einen Agentur-Background haben und von daher einfach auch die agilen Werte an sich leben. Das heißt es gibt eine unheimliche Feedbackkultur und es wird Wert gelegt auf Offenheit/Openness. Also ich kann an der Stelle in alle Produkt Backlog reinschauen, ich kann in alle Teams reinschauen, ich kann an deren Zeremonien teilnehmen, da ist ein unheimlich hoher Kommunikationsfluss, der sichergestellt wird. Sprich das Menschen sich über Produkte austauschen und Menschen miteinander sprechen. Also die Kultur ist die Basis mehr oder weniger für agiles Arbeiten. Wir haben auch festgestellt in Phasen, wo wir stärker gewachsen sind und in sehr kurzer Zeit viele Menschen dazu gekommen sind, da hat man gemerkt, dass unterschiedliche Kulturen erstmal zueinander finden müssen. Damit diese gemeinsamen Werte auch wieder initiieren werden müssen, da alles was an Methode und Framework dann auch wirklich in den Produkt Teams eingesetzt wird, eben auf agilen Werten fußt. Wenn die nicht verstanden

sind und wenn Menschen grundsätzlich ein anderes Verständnis haben, wird es meistens schwieriger und da spielt Hierarchie ganz klassisch auch mit.

I: In welcher Form werden denn Mitarbeiter bei euch gefördert bzw. weitergebildet?

IP: Also das Thema Weiterentwicklung fällt so ein Stück weit in das Thema Enablement bei uns mit rein. Dadurch das wir in Disziplinen aufgestellt sind. Also z. B. den QA- oder Scrum Master-Circle. Dieser Circle kümmert sich, um die Weiterbildung in der jeweiligen Fachdisziplin. Darüber hinaus gibt es auch Company-weite Förder-Methoden, wie z. B. dass es regelmäßige Company-Retrospektiven gibt oder auch Events, an denen wir unsere Produkte und Entwicklung vorstellen. Es gibt auch ganz klassische Weiterbildungsmaßnahmen also externe Schulungen und solche Geschichten, wobei auch gerade wir als Scrum Master-Circle selbst Schulungen intern entwickeln und anbieten. Das ist ein sehr großes Thema, was wir uns seit letztem Jahr auf die Fahne geschrieben haben. Das ist auch diese Coaching-Stärke in die Organisation reinbringen. Das heißt wir müssen auch nicht mehr auf teure Schulungen schicken, sondern wir können das auch intern machen also hands-on im Team oder an externen, dafür ausgewählten Schulungstagen hier intern machen.

I: Welche Methoden und Werkzeuge werden zum Zwecke ständiger Verbesserung in die Arbeitsweise etabliert?

IP: Der Sinn von z. B. Scrum ist der regelmäßige, sogenannte empirische Rückblick. Das heißt jede Zeremonie hat den Auftrag die vergangene Periode zu betrachten und mit Blick auf die nächste Periode zu planen und zu adaptieren, was zu verbessern ist. Also das heißt im Endeffekt jede Zeremonie, die wir machen hat einen kleinen Auftrag, das heißt in unserem Daily ist der erste Schritt, das Gelernte im nächsten Zyklus anzuwenden. Das heißt wenn ich nachher in das Team Daily gehe, erzähle ich woran ich gestern gearbeitet habe und das ist die erste Möglichkeit kontinuierliche Verbesserung einzuleiten. Das heißt ich habe gestern an etwas gearbeitet, festgestellt, dass das nicht funktioniert und nehme mir dafür heute folgende Gegenaktivitäten vor. Allein durch dieses Sharing in unserem 15-Minuten-Daily gibt es schon ganz kleine Verbesserungen. Und diese Art der Veränderung und kontinuierlichen Verbesserung findet eigentlich in jeder Zeremonie statt. Die Hauptzeremonie ist immer am Ende der Iteration, die normalerweise zwei Wochen lang ist, die Retrospektive. In dieser schaut das Gesamtteam auf unser Gesamt-Inkrement und sagt, okay wie lief denn die Iteration, was kann man besser machen, wo hat es gehakt, was waren Schwierigkeiten und dann einfach die Maßnahmen auch plant und umsetzt, um einfach diese kontinuierliche Verbesserung erlebbar zu machen. Also im Endeffekt ständig und täglich.

I: Worin siehst du die größte Herausforderung, wenn es darum geht das Qualitätsmanagement mit agilen Arbeitsweisen zu verbinden?

IP: Also ich glaube, es ist an sich nicht schwierig. Es bedarf einem unheimlichen Kommunikations- und Aufklärungs-Aufwand, aber an sich ist es durchaus möglich. Wir kommen auch immer häufiger an die Punkte, an denen gerade auf der Business-Seite der Ansprechpartner mit riesen großen Go-Live und Cut-over-Plänen mit End-to-End Testing und mit von-bis und Involvement von X externen Testern und Suppliern und Zulieferern. Da wird das ganze Thema Qualität eben aufgeblasen im Sinne von Organisationen, Menschen und Plänen, die im Endeffekt vielleicht das Thema Qualität nicht so stark beeinflussen. Das heißt das Verständnis schaffen, wie wir hier arbeiten, Verständnis schaffen, das wir Qualitätsmanagement machen schon von Tag eins an, weil das in unsere Prozesse etabliert ist. Das heißt nach den ersten zwei Wochen kommt schon ein gesichertes Inkrement heraus. Das heißt man braucht gar nicht diese Big-bang-Veranstaltung ganz am Ende, sondern man braucht dort einfach nur noch einen geregelten Ablauf eine gemeinsame Abstimmung. Gerne auch noch mal End-to-End etwas ausführlicher, aber durchaus nicht diesen ganz großen Aufwand. Das heißt um noch mal zurück zur Frage zu kommen, die größte Herausforderung ist mehr weniger die Aufklärung, wie agiles arbeiten funktioniert, und wie externe Business Partner sich einbringen können, um schon im Produktentwicklungs-Zyklus dieses Qualitätsmanagement auch nachvollziehen zu können. Also das die schon wissen, wenn ein Meilenstein X eintritt, wurde schon vier Wochen, sechs Wochen, acht Wochen vorher Qualität gesichert und dass die dabei waren, um zu sehen was passiert ist. Dass das auch für sie relevante Minimeilensteine sind, die sie eigentlich schon in ihren Qualitätsmanagement-Plan überführen können und deswegen gar nicht mehr diese extremen Organisations-Overhead-Geschichten fahren müssen. Das ist glaube ich die größte Herausforderung.

## Interview 4

### **Interviewpartner: Projektleiter-Assistenz und Scrum Master aus einem IT- und Strategie-Beratungsunternehmen, welches in verschiedenen Industrien tätig ist**

(Abkürzung: IP - Interviews Partner, I - Ich)

I: Welcher Branche ist dein Unternehmen zugeordnet und in welchem Bereich bist du denn tätig?

IP: Das Unternehmen, in dem ich arbeite ist der Unternehmensberatungs-Branche zugeordnet mit Schwerpunkt IT-Beratung im Speziellen Business Intelligence, Big Data also Informations- und Managementsystem und da haben wir so eine Art Rundumbetreuung. Das heißt, komplett Neu-Implementierung, Neu-Design und Strategieberatung und ja das volle Programm. Im Innovation Lab sind wir auch was agile Arbeitsweisen angeht sehr affin. Wir sind in so ziemlich sämtlichen Geschäftsbereichen von Versicherung bis Handelsunternehmen und in vielen der größeren Unternehmen in Deutschland tätig.

I: Welche Produkte werden dann in deinem Bereich bzw. in deinem Unternehmen hergestellt und entwickelt?

IP: Also von uns? Was wir da entwickeln?

I: Genau.

IP: Ja wir entwickeln eine große Planungssoftware also ich bin z.B. Bereich SAP-BWP tätig. Und da werden mit dem Web Excel Add-in Analysis-for-Office so ziemlich alle Module oder Bereiche eines Unternehmens geplant. Also sprich Umsatz, Sachkosten, Personal und das kann dann Filial-genau auf den Tag genau für das Geschäftsjahr geplant werden. Also im Anschluss werden die Daten dann halt konsolidiert und auch wieder zurück ins BW-System zurückgeschoben und dann finden da auch die Reports statt. Und sowas machen wir, aber wir können auch die wildesten anderen Sachen machen. (lacht) Also wir haben auch z.B. Big Data Kollegen, die versuchen die Entwicklung des Super-Preises durch Predictive auf die Spur zu kommen. Alles von Betrieb und Support machen wir auch. Also alles was mit Business Intelligence und Daten zu tun hat. Daten ist unser Kerngeschäft.

I: Welche Stellung besetzt du denn in diesem Bereich? Was ist dein Aufgabenbereich?

IP: Ich bin Consultant und seit zwei Jahren bin ich im Projektmanagement tätig. Und ich bin auch Scrum Master habe auch die Zertifizierung Prince2 Agile. Ich weiß nicht, ob du den Standard kennst? Das ist ein Projektmanagement-Standard und Prince steht dabei für Project Inner Control Environment. Das ist so ein recht guter Projektmanagement-Standard. Davon gibts dann noch ne quasi Weiterentwicklung, die sich im speziellen mit der agilen Arbeitsweise beschäftigt. Also ich bin als PMO gerade tätig. Also ich unterstütze den Projektleiter. Ja das ist ein Großprojekt das sollte jetzt schon seit zwei Jahren... wir haben das übernommen von einem anderen Dienstleister, wo es dann aufgrund der Komplexität nicht so gut gelaufen ist. Und durch agile Anpassung haben wir das Projekt jetzt schon bis zu einem zweiten Release erfolgreich durchgeführt. Und ja jetzt sind wir quasi am dritten Release. Das mache ich.

I: Wie lange bist du denn schon in diesem Bereich tätig oder sagen wir mal, generell in der Entwicklung von Softwareprodukten oder Projektmanagement von Softwareprodukten?

IP: Also bei Unternehmensname [REDACTED] bin ich seit über drei Jahren und fünf Monaten und davor war ich auch im SAP-Bereich, aber mehr Support und Datenentwicklung. Also noch nicht so nah an der BI, wie ich es jetzt bin.

I: Wie sind denn die Bereiche Entwicklung und Qualitätssicherung in die Unternehmensorganisation eingegliedert? Sind das eigene Funktionsbereiche oder ist das Qualitätsmanagement in gewisser Weise in die Entwicklung schon integriert?

IP: Also das ist grundsätzlich bei Beratungsunternehmen eben auch von Projekt zu Projekt unterschiedlich. Wir sind da natürlich flexibel je nachdem wie die Unternehmen das handhaben wollen. Wir haben natürlich auch einen Standard bei uns Prince und wir haben natürlich auch bestimmte Qualitätsmanagement-Zyklen und Best Practices, die wir anwenden, aber an sich sind wir da sehr agil und passen uns an. Also wir haben da... also Qualitätsmanagement ist im Entwicklungsprozess schon integriert, mal mehr mal weniger.

I: In welcher Form und in welchem Umfang sind den agilen Arbeitsweisen bei dir im Unternehmen oder im Projekt integriert?

IP: Also bei uns jetzt im aktuellen Projekt da sind wir... da war der Gedanke, dass wir uns nach Scrum komplett organisieren. Nachdem wir das Projekt übernommen haben, haben wir agile Methoden schon eingeholt. Also wir haben z.B. durch den Product Owner und Teilprojektleiter eine recht gute Struktur geschaffen, welche eine sehr intensive Abstimmung mit Anforderungsgebern anbietet. Und wir benutzen auch z.B. Gira und arbeiten agil nach der Kanban Methode. Wir arbeiten jetzt nicht wirklich mit Sprints, aber wir haben uns stark an Scrum angenähert. Arbeiten aber nicht wirklich nach Scrum. Wir haben da verschiedene Workflows in Gira die uns helfen während der Entwicklung ein Auge auf Qualität zu haben. Da gibt es dann... also du weißt ja wir haben ein Softwareentwicklungsprojekt, da gibt es auch im klassischen SAP-System immer eine Entwicklungsumgebung und eine Qualitätssicherungsumgebung und eine Produktion. Und das ist auch bei uns im Workflow so abgebildet. Und wenn irgendetwas, sage ich mal auf E fertig entwickelt ist, ist dann ein Workflow, ein entscheidender Workflow-Schritt, dass das dann zur Q kommt, da wird es dann getestet teilweise vom Product Owner, teilweise dann von den verschiedenen Ländern, die später die Software nutzen. Und wenn die dann das Okay geben, dass eine bestimmte Funktion oder sei es ein bestimmter Bug, der geändert wurde, dann kann das auch geliefert werden und da wird es dann auch noch mal auf Funktionalität geprüft.

I: Wie schaut es bei euch aus, wenn mehrere Teams an einem Projekt arbeiten? Wie stimmt ihr euch ab? Wie werden da Software Inkremente integriert?

IP: Was verstehst du unter mehrere Teams? Also Teams mit unterschiedlichen Aufgaben oder was meinst du?

I: Ich meine du hast eine Software, die erstellt werden will, oder ein Projekt, und splittest das in verschiedene Teilprojekte. Und sagst dann, okay jedes Teilteamprojekt hat jetzt die Aufgabe diesen oder jenen Software-Part zu erstellen und am Ende läuft das dann wieder zusammen und muss integriert werden in ein Drittsystem oder in ein Gesamtsystem.

IP: Okay also wir haben uns da im jetzigen Projekt so aufgestellt, dass es zwei Teilprojektleiter gibt, die verschiedene Teams unter sich haben. Also je nach Modul ein Team und in jedem Team gibt es noch einmal ein Lead-Entwickler also einen führenden Entwickler, der steuert die anderen Teams innerhalb der Teams. Und wir haben dann auch ein paar Focus-Module, die keinem direkten Teilprojektleiter zugewiesen sind, sondern wo die Chefentwickler quasi selber den Hut aufhaben. Und wir haben innerhalb dieser Teilprojektleiter-Teams kontinuierliche Abstimmungen unter der Woche und dann gibt es jeden Mittwoch einen Gesamt-Jour-fix zwischen dem Projektleiter und den Teilprojektleitern und Teams. Da werden dann die wesentlichen Punkte beschrieben und besprochen. Das geht so ungefähr zwei Stunden das Meeting und da werden dann eben Impediment-Probleme besprochen und er wird dann ein Statusbericht erstellt und besprochen der dann auch an den Lenkungsausschuss also den Projektleitungsausschuss geht. Und so haben wir immer ein gutes Bild, was gerade in den einzelnen Modulen passiert bzw. wo die Schmerzen sind. Und wenn dann halt von uns eingegriffen werden muss oder eskaliert werden muss, steht das recht schnell parat. Also die Informationen, die wir brauchen. Wir sind ein ganz gut eingespieltes Team. Und das haben wir dann über solche regelmäßigen Termine ganz gut abdecken können... die Kontrolle.

I: Wie ist das denn mit den Mitarbeiterkompetenzen... Wie setzt sich da ein Projektteam denn zusammen? Sind denn da ganz verschiedene Backgrounds in so einem Team?

IP: Bei Scrum ist es ja auch so, dass man ein Team von verschiedenen Generalisten hat, wo jeder alles kann. Wir haben bei uns schon sehr spezialisierte Kollegen, die in gewissen Modulen schon sehr affin sind, sei das jetzt Sachkosten oder Umsatz das sind schon recht taffe Module und Berechnungen und recht kompliziert und gar nicht so einfach. Das sind alles schon Entwickler, die Erfahrung haben in SAP BI, also das ist so eine Art Planung. Und dann haben wir natürlich, die eher individuell auf die Datenbereitstellung spezialisierten (Entwickler). Und ja so haben wir durch jedes entsprechende Softwaremodul also für jeden Teilaspekt der Planung haben wir dann verschiedene Experten. Und die sind überwiegend auch extern sei es von unserer Firma [REDACTED] aber auch noch ein-zwei anderen Entwicklungsfirmen oder Freelancer, die dann so schwierig am Markt zu bekommen sind. Und so arbeitet man dann teilweise auch mit der internen BW-Abteilung zusammen oder dem Support und so versucht man dann die Herausforderung zu meistern. Und dann haben wir natürlich auch noch administrative Funktionen. Das wären einmal dann der Projektleiter. Wir haben praktisch zwei Projektleiter, einen technischen also meinen Chef und dann haben wir noch einen fachlichen Projektleiter, der quasi die Schnittstelle regelt. Das hätte ich vielleicht auch noch sagen sollen, also unser Projekt ist angegliedert am IT-Teil und am Controlling-Teil. Also bei uns in der Branche sind wir gerne in der Mitte von beiden Abteilungen, also wir arbeiten dann mit beiden zusammen. Das ist eben auch wichtig zu sagen. Und um mit beiden Seiten kommunizieren zu können, haben wir zwei Projektleiter. Das hat sich eben im zweiten Jahr so entwickelt. Am Anfang war das nur der technische Projektleiter, der den Hut auf hatte und danach kam dann der jetzt fachliche Gesamtprojektleiter hinzu, der den Hut auf hat für die Product Owner, die alle aus dem Controlling kommen und dann mit den unterschiedlichen Stakeholdern kommunizieren und die jeweiligen Anforderungen sammeln dann

priorisieren. Also wir arbeiten auch ähnlich wie bei Scrum mit einer Backlog, wo die Anforderungen dann priorisiert werden und mit Stories und Tasks werden die dann zu Epics zusammengefasst und in den entsprechenden Entwicklungszyklen realisiert.

I: Du hast es schon angesprochen, ihr arbeitet auch teilweise mit Lieferanten zusammen. Sind die dann von anderen Softwarefirmen oder wie kann man sich das vorstellen?

IP: Also mit anderen Lieferanten da meinte ich in erster Linie, dass wir noch andere Dienstleister also andere Beratungsunternehmen haben, weil wir halt so ein großes Projekt sind mit über 20 Entwicklern. So meinte ich das. Aber als Lieferant oder als anderen Stakeholder haben wir natürlich auch andere Unternehmensbereiche von unseren Kunden, auf die wird dann auch angewiesen sind. Z.B. bei der Datenbereitstellung, dass wir die entsprechenden Zugänge für diese Systeme haben damit wir die Daten anbinden können. Sei es jetzt per direkten Verbindung oder eben Flat-File-Beladung. Und da ist es dann halt auch im Hinblick agiler Arbeitsweisen kompliziert, weil wenn man selber agil arbeitet, aber die Abteilung, die einem die Daten zustellen sollen nicht agil arbeitet, wird's auch schwierig. Also da muss man dann manchmal auch kontinuierlich Druck machen. Das sind dann auch die Grenzen, wenn man in einem großen, übergreifenden Projekt agil arbeitet, aber die Schnittstellen eben nicht agil arbeiten. Und dann wird da... die haben eben auch ihre Arbeiten und müssen schauen, dass ihre Ziele und Aufgaben fertiggestellt werden und wenn da halt ein Projekt ankommt dann kann das manchmal sehr schrecklich werden.

I: Das habe ich jetzt auch schon öfters mitbekommen von vielen Seiten, dass es schwierig ist, diese Schnittstellen dann richtig aufzubauen.

I: Wie werden denn Kundenanforderungen an das Produkt ermittelt? Zu welchem Zeitpunkt findet das denn statt? Ist das ein kontinuierlicher Prozess oder ist das ein einmaliger Prozess?

IP: Also idealerweise ist das am Anfang eines Sprints oder am Anfang eines Entwicklungszyklus. Das ist schon sehr hilfreich, wenn alle Informationen feststehen würden, aber leider ist das nie so. Also es kann natürlich sein, dass sich etwas ändert, es kann aber auch sein, dass neue Anforderungen hinzukommen, weil es z.B. Unternehmensentscheidungen gibt. Also wir gehen mit solchen Anforderungen und mit sich ändern Anforderungen insofern um, als dass wir einen Changemanagementprozess definiert haben. Und wir haben einen bestimmten Schwellenwert definiert, wenn eine neue Anforderung sei es jetzt in ticketform oder generell kommt die einen gewissen... die eine gewisse Anzahl von Arbeitstagen überschreitet... Ich glaube, wir haben da zehn Arbeitstage (PT) definiert, muss das sogar in den Lenkungsausschuss. Also da wird dann auch abgestimmt zwischen... Also vorher wir sich noch mit den TPLs und den Projektleitern abgestimmt. Und wenn es dann tatsächlich heißt ja das muss gemacht werden, weil sonst das und das nicht funktioniert oder das anders keinen Mehrwert bringt, dann müssen wir das in den Lenkungsausschuss bringen und da wird dann nochmal abgestimmt, ob etwas getan werden muss. Also wir haben sehr oft und viel mit sich ändernden Anforderungen zu tun. Und ja wir mögen das eigentlich nicht manchmal kommt es dann aber eben trotzdem. Wir haben auch eine spezielle Priorisierung eins bis drei und quasi alles das mit einer Prio eins wird umgesetzt im aktuellen Release, wenn dann neue Anforderung vom Kunden kommen mit Prio eins, die aber noch nicht gecheckt wurden, werden wir auch schon oft ungehalten. Aber das passiert leider sehr oft und dann muss man sich eben abstimmen und kommunizieren und meistens ist das am Schluss doch nicht so wild und es kann am Ende doch noch umgesetzt werden, wenn es jetzt in den entsprechenden Zyklus passt. Aber das ist eben agil. Es wäre ja natürlich fantastisch, wenn im Vorfeld der Scope schon definiert wäre, aber das ist eben auch der Vorteil an der Agilität, dass man dann noch, wenn's passt, Sachen aus der Backlog nachziehen kann und diese dann auch umsetzen kann.

I: Wie lange ist bei euch denn ein Sprint?

IP: Also da wir nicht wirklich nach Scrum arbeiten, ist eigentlich bei uns der Sprint die Zeit bis zum Release. Also wir haben eine Releasezeitpunkt und bis dahin muss dann eben gearbeitet werden und bis dahin müssen die entsprechenden Module eben fertig sein. Also das würde jetzt keinen Sinn machen, einen vier Wochen Sprint zu haben, weil unserer Planungstool eben nur nach einem bestimmten Punkt überhaupt benutzt und beplant. Es werden halt auch unterschiedlich Module nach und nach freigegeben. Es macht keinen Sinn, dass man ein inkrementelles Produkt schon nach vier Wochen liefert, weil das dann kein Wert hat, wenn man damit nicht arbeiten kann. Das ist in einem großen IT Projekt eben ein bisschen anders.

I: Am Ende eines Releases steht dann schon auch ein erster Prototyp oder ein testfähiges Produkt richtig?

IP: Ja es werden da natürlich kontinuierlich in den unterschiedlichen Modulen Prototypen entwickelt. Und wenn die Entwicklung abgeschlossen ist und eine spezielle Berechnung oder spezielle Kennzahlen, die beplant werden müssen, fertig sind dann könnte man natürlich schon von einer Art Prototyp sprechen, da gebe ich dir Recht ja. Aber wir haben da keine festen Zyklen, sondern es gibt eben Meilensteine, die dann eben in der Zusammenarbeit mit Projektleiter, Teilprojektleiter und POs definiert wurden. Bis dann und dann muss das und das fertig sein. Ja und wenn es das dann ist, dann sind wir glücklich. (lacht)

I: Bei digitalen Produkten also im Automobilbereich ist es ja so, dass du oftmals eine Hardware Komponente im Auto hast und eben eine Software oder eine App, die dann praktisch mit diesen Hardwarekomponenten kommuniziert. Wie ist das denn bei euch, wenn ihr Software erstellt, die dann auf irgendwelchen Hardware-Komponenten vom Kunden funktionieren müssen? Wie läuft das dann mit der Integration oder auch mit der Zusammenarbeit ab?

IP: Mit Hardware?

I: Genau.

IP: Also wir sind natürlich als Beratungsunternehmen und Partner auch von verschiedenen Software Häusern natürlich sehr davon abhängig, dass wir dann zugesagte... das fängt dann schon z.B. bei bestimmten Lizenzen an. Da muss man dann halt vom Kunden, den man dann auch berät und entwickelt, die zuvor abgestimmten Softwareprodukte dann auch tatsächlich lizenziert bekommen. Wenn man jetzt zum Beispiel bei SAP mit einer ganz neuartigen Software arbeitet, die dann plötzlich nicht fertig wird, dann können sich natürlich gesamte Projekte verschieben und insofern geht man schon... also im Idealfall sind die Hardwarekomponenten dann schon da, bevor wir das Projekt starten. Aber so ist es natürlich in der Realität oft nicht. Jetzt bei uns im IT-Bereich haben wir natürlich nicht oft mit irgendwelchen Hardware-Komponenten zu tun so wie du im Automobilbereich zu tun hast. Daher sehe ich bei uns eher, dass es um Lizenzen oder um ganze Produkte geht, die dann schnell entwickelt werden. Bei uns ist auch noch das Thema, dass es dann zu neuen Releaseständen kommt also Updates oder neuen Services. Da sind wir dann auch von der IT-Abteilung abhängig von der BW-Abteilung das da entsprechend Updates, die erforderlich sind, z.B. um irgendwelche Bugs oder Errors von Kunden zu beseitigen, da sind. Da ist man natürlich immer auf die interne Abteilung angewiesen. Also in dem Sinne als Hardware-Lieferanten, obwohl es eigentlich Software ist, wenn du verstehst worauf ich hinauswill. Das ist bei uns die Hardware, von der ich spreche. Oder wenn man die Datenbanktechnologie nimmt. Also die Datenbank kann an sich auch als eine Art Mischform von Software und Hardware angesehen werden. Ansonsten das wir wirklich ein Produkt haben, was... Ich überleg gerade... schwierig. Ansonsten fällt mir da gerade nichts ein, was wir da hardwaretechnisch gemacht haben und noch besonders in meiner Erinnerung geblieben ist oder von anderen Projekten, von denen ich gehört habe. Da sind wir eher Software-seitig unterwegs.

I: Wie wird die Qualität von Softwareprodukten in der Entwicklung sichergestellt? Wird da Testing eingesetzt? Oder welche Methoden kommen da zum Einsatz?

IP: Also wir haben bei uns im Projekt, da wir einen sehr großen sind, haben wir einen eigenen Testmanager sogar eine eigene Firma, die für das Testen verantwortlich ist. Da gibt es dann erstmal interne Tests, die quasi von den Testern nach einem bestimmten Testkatalog durchgeführt werden. Und dann haben wir natürlich auch von Kundenseite bestimmte Ländertests, wo praktisch auf Freiwilligkeit verschiedene Länder insgesamt bis zu 200 Tester definieren, die dann schon einen (?)Test durchführen. Und dann werden quasi in einem speziellen Gira die Fehler gesammelt und dann auch möglichst beseitigt bzw. priorisiert und kategorisiert. War das jetzt ein Anwenderfehler oder ist das jetzt eine mangelnde Schulung gewesen? Das sind bei uns die Punkte. Aber das kann natürlich auch von Projekt zu Projekt unterschiedlich sein. Teilweise ist so das Testmanagement oder das Testing ein Punkt ist, wo Unternehmen gerne Einsparen wollen. Das ist natürlich ein zweischneidiges Schwert, dass kann natürlich Kosten einsparen, aber es kann natürlich auch das Genick brechen.

I: Gibt es denn fest definierte Schnittstellen entlang dieses Zyklus, wo eben die Entwicklung oder das Qualitätsmanagement zusammenarbeitet?

IP: Teilweise sind das Qualitätsmanagement oder das Testmanagement als Ausprägung... also das Qualitätsmanagement ist eine Ausprägung des Testmanagements. Also wir haben jetzt keinen eigenen Qualitätsbeauftragten, sondern die Qualität liegt im Auge des Betrachters also das ist eine subjektive Betrachtung und Wahrnehmung. In unserem Falle sind das dann die 30 verschiedenen Länder, die das Planungstool nutzen. Die Qualität wird dann halt quasi durch deren Zufriedenheit mit dem Tool definiert, aber wir können natürlich im Vorfeld die Qualität sicherstellen, dass die einzelnen Berechnungen, Kennzahlen und Planungsfunktionen dann auch tatsächlich richtig funktionieren. So einen richtigen Qualitätsbeauftragten gibt es in unseren Projekten nicht. Also das wird dann einfach von den verschiedenen Teilprojektleitern und Product Ownern und dann von den Ländern definiert. Du kannst auch gerne nachfragen, wenn dir etwas unklar ist oder wenn ich an der Frage vorbei spreche.

I: Ja klar ich höre mir das nur erstmal an und muss dann auch nochmal darüber nachdenken und stelle dann schon auch Zwischenfragen. Also die Fragen, die ich bisher gestellt habe, sind nicht alle auf dem Fragenkatalog enthalten. Gerade bei diesen qualitätssichernden Maßnahmen... Man muss ja schon schauen, dass diese Software, die man eben erstellt und entwickelt, dass die dann eben auch auf Drittsysteme integriert werden kann und dass das dort dann auch reibungsfrei auf der Umgebung läuft richtig?

IP: Ja genau das ist bei uns auch besonders wichtig. Es gibt ja so die IT-Umgebung. Ich spreche jetzt nicht unbedingt von den großen Serverlandschaften oder von den großen SAP-Systemen, sondern der IT-Infrastruktur im Sinne von den Rechnern, die die Kollegen dann später benutzen. Die kann auch im schlimmsten Fall heterogen sein. Also das man dann... das heißt z.B. Windows 10 wird gerade ausgerollt und dann hast du Länder und Nutzer, die bereits Windows 10 mit Office 365 haben und dann hat man Länder, die ältere Windows und Office Version nutzen. Und ja dann beginnen die Probleme, denn man entwickelt auf einer festen, standardisierten Umgebung und wenn dann halt trotzdem Unterschiede in der Landschaft sind, kann es Probleme geben. Wir haben dann natürlich eine Art Order von hoher Stelle an die IT-Leiter der unterschiedlichen Länder geschickt, dass das und das sicher gestellt werden muss, aber manchmal stellt man dann fest, dass bestimmte Infrastrukturen, die in einem Land genutzt werden, in dem anderen Land gar nicht möglich sind zu bestellen. Und dann beginnen natürlich die Kopfschmerzen, aber bisher ist das alles gut gegangen. Problematisch wird es dann, wenn z.B. das Excel Add-in von SAP dieses Analysis for Office das wir benutzen, wenn da dann z.B. ein Upgrade vor einer Releaseversion ansteht, dann kann es ganz schön knifflig werden. Und in den letzten Jahren ist ein Upgrade dieses Tools, ja wie soll man sagen, nicht sehr erfolgreich durchgeführt worden von der entsprechenden IT-Abteilung. In diesem Jahr haben wir dafür dann verschiedene funktionale Tests und Abnahmen damit das dann in diesem Jahr kein großes Problem mehr wird.

I: Und bei diesen Abnahmetests, da wird dann auch End-to-End Testing vollzogen, richtig?

IP: Genau es gibt auch verschiedene Testformen. So eine Art Smoketests, wo dann einfach durchgeführt werden. Dann gibt es natürlich auch noch, na wie nennt sich das, Integrationstest, dass dann halt bestimmte übergreifende Systeme miteinander zusammenarbeiten in den verschiedenen Modulen. Ja da sind aber auch verschiedene Testkataloge definiert und an denen wird dann versucht, alle Fehler frühzeitig zu entdecken. Klar kann da immer mal etwas durchrutschen denn bestimmte Testkonstellationen, die so absurd sind, kann man sich nicht ausdenken, dass kann immer mal passieren, da muss man dann eben rechtzeitig reagieren und dann hat man das schon ganz gut im Griff.

I: Wenn man über agiles Arbeiten spricht, dann sagt man oft, dass das diese von der Kultur des Unternehmens in gewisser Art und Weise unterstützt werden muss. Z.B. gerade in Unternehmen mit einer sehr steilen Hierarchie passt das meistens nicht so gut zusammen, weil das ja zu diesen (agilen) Prinzipien auch sehr konträr ist. Wie ist das bei euch in der Unternehmenskultur? Wie siehst du da agile Arbeitsweisen unterstützt?

IP: Also bei uns Intern auf jeden Fall, da werden agile Arbeitsweisen standardmäßig unterstützt. Ich habe den Eindruck, dass auch bei unseren Kunden gerade in der IT Abteilung das agile Arbeiten sehr stark genutzt wird, aber nicht komplex übergreifend, sondern eher in bestimmten Teilabteilungen. Gerade im Business Intelligence Bereich oder in der Entwicklung von großen Managementsystemen, da hat man den Eindruck, dass wirklich nach Scrum gearbeitet wird und dass das da auch wirklich genutzt wird. In anderen Abteilungen gerade in konservativen Unternehmen ist es natürlich recht schwierig, da mit einer komplett neuen Arbeitsweise anzukommen. Es werden dann gerade in konservativen Unternehmen auch Sachen gefordert, die z.B. bei Scrum nicht standardmäßig vorgesehen sind. Also z.B. Statusbericht oder die Nutzung eines Projektleitungsausschusses. Die sind ja per se auch nicht direkt in Scrum erforderlich oder geplant. Da muss man dann eben dementsprechend reagieren, dass das dann auch trotzdem... also die Managementanforderungen trotzdem getroffen werden. Aber agil ist ja auch nicht immer Scrum, sondern es gibt ja auch verschiedene andere Arbeitsweisen z.B. Kanban oder Prince2 Agile, die dann solche konservativen Vorgaben, welche man aus dem alten Projektmanagement kennt, dann auch trotzdem berücksichtigen.

I: In welcher Form werden denn Mitarbeiter bei dir in der Abteilung oder im Team gefördert oder weiterentwickelt?

IP: Du meinst in Richtung Agilität?

I: Eher generell also einmal hinsichtlich der fachlichen Ausbildung, dass man sagt, okay wir wollen eben auch immer an den brandaktuellen Technologien dran sein und immer den neuesten Expertise-Stand haben und deshalb eben die Weiterbildung von Mitarbeitern anstoßen. (Und auf der anderen Seite auch hinsichtlich der agilen Arbeitsweisen.)

IP: Die findet sehr gut und sehr oft statt. Das hängt natürlich auch von den einzelnen Mitarbeitern und Vorgesetzten ab, dass man da eine entsprechende Zertifizierung anstrebt. Aber das wird sehr oft gemacht. Sei es nun im IT-Bereich, dass man da für bestimmte Technologien gewappnet ist, aber auch standardmäßig werden bestimmte Projektmanagement Skills wie z.B. Scrum auch als Schulung angeboten. Teilweise auch mit Zertifizierung also gerade die Unternehmensberatungen sind da auch erpicht, dass da sehr viel in die Mitarbeiter investiert wird und insofern hat man oft das Gefühl, dass Zertifizierungen oder Schulungen

stattfinden. Auch bei Scrum. Aber so Zertifizierung sind da nicht immer geplant und es geht eher um das Wissen. Aber man hat manchmal das Gefühl... Jetzt nicht bei dem jetzigen Kunden... Aber so Zertifizierung werden dann gefühlt nicht so verstanden, nicht auf der breiten Front. Gerade wenn das irgendeine teure Zertifizierung sind, dann haben die Unternehmen manchmal Angst, dass die Mitarbeiter sich dann davon machen und die teuren Zertifizierungen mitnehmen. Also das ist zwar selten der Fall, aber man hört immer mal wieder Geschichten.

I: Ja klar. Aus der Unternehmensberatung hört man das schon öfters mal, dass da abgeworben wird oder so. Man ist während seinen Projekten auch lang genug miteinander in Kontakt oder in Kooperation.

IP: Das sind so die Punkte, die mir zu Zertifizierung und Weiterbildung einfallen und ja gut gerade bzgl. agil. Wenn man z.B. jüngere Kollegen einstellt... Das man dann einfach dieses Training-on-the-Job ermöglicht. Also das man quasi jüngere Kollegen entwickelt und an diese agilen Arbeitsmethoden heranzuführt. Das ist eben auch noch ein guter Punkt dieses Training-on-the-Job. Oder vielleicht auch als Schlagwort die Jobrotation. Das kann dann, wenn man jetzt ein großes Unternehmen hat oder eben verschiedene Arbeitsmethoden, Projektmanagement Methoden benutzt werden. Das man dann z.B. auch ermöglicht in eine Abteilung zu wechseln, wo agil gearbeitet wird. Also das wäre natürlich auch eine gute Weiterbildung für die Kollegen.

I: Was für Methoden und Werkzeuge benutzt ihr denn, damit ihr euch auch ständig verbessert und ständig reflektiert? Sprich was gab es denn für Probleme im letzten Sprint, was kann man besser machen. Was für Methoden sind denn da bei euch etabliert?

IP: Also wir machen nach großen Releases regelmäßig so eine Art operatives Team-Meeting, wo wir dann mit fast über 30 Leuten zusammensitzen und die entsprechenden wichtigen Punkte besprechen. Wenn ein Software-Release realisiert wurde, kann es auch sein, dass man danach eine Art Lessons-learned macht und da auch bestimmte Punkte anspricht. Und da haben wir verschiedene Meeting Typen, wo wird das besprochen. Anders als z.B. bei Scrum gibt es jetzt keine Sprint-Retrospektive, aber so etwas Ähnliches kann man bei uns mit Lessons-learned beschreiben, wo man bestimmte Punkte diskutiert.

I: Ich denke auch, dass das meistens dann auch nur ein anderer Name aber das gleiche gemeint ist.

IP: Das stimmt.

I: Worin siehst du denn die größte Herausforderung, wenn es darum geht das traditionelle Qualitätsmanagement mit den agilen Arbeitsweisen zu verbinden?

IP: Also ich würde einfach mal festlegen, dass gerade im Qualitätsmanagement feste Zyklen der Kontrolle vorgesehen sind und wenn dann wirklich agil ständig sich ändernde Anforderungen... also wenn die Anforderung an die Qualität sich während des Entwicklungsprozesses ändern, kann es natürlich für das Qualitätsmanagement sehr schwierig werden. Und gerade, wenn ich jetzt z.B. an Scrum denke, da wäre es dann natürlich sinnvoll, das Qualitätsmanagement nicht unbedingt als separate Abteilung zu sehen. Das es halt sinnvoll wäre, wenn in dem entsprechenden Scrum-Team eine entsprechende Person wäre, die sich, wenn es ein wichtiges Thema ist, um die Aspekte der Qualität kümmert. Das man in dem Entwicklungsprozess eben schon Qualität agil abbildet und nicht etwa als nachfolgenden Prozess. Ich sollte mich immer noch auf Software konzentrieren nicht unbedingt auf Hardware richtig?

I: Genau eher auf Software.

IP: Genau deshalb würde ich schon sagen, dass es wichtig wäre innerhalb des Scrum Teams oder des agilen Teams auch den Punkt Qualität oder Qualitätsmanagement zu verankern. Das man dann da die entsprechenden Kontrollinstanzen während des Sprints lenkt und in der Entwicklung berücksichtigt. Nicht das man noch eine separate Qualitätsmanagement-Abteilung hat, die dann noch mal aktiv werden muss. Das ist jetzt meine Einschätzung, wenn man sich auf Scrum konzentriert. Natürlich kann man das in verschiedenen anderen agilen Arbeitsweisen ähnlich halten, dass man dann das Qualitäts-Team kontinuierlich während der Entwicklung auch schon integriert.

I: Ja ist interessant, weil gerade viele eben sagen, dass man das Qualitätsmanagement nicht als einzelner Block angewendet, sondern wirklich mehr integriert werden sollte. Damit da auch wirklich bessere Kommunikationswege geschaffen werden, weil oftmals eben schon viel getestet wird, was aber gar nicht ans Qualitätsmanagement berichtet wird. Was dann auch wieder in gewisser Art und Weise unter den Tisch fällt.

IP: Also ich denke, zum Qualitätsmanagement. Wenn es so eine Abteilung gibt, kann es natürlich da unterstützen und auch Coachen. Also das man den Scrum-Teams die Werkzeuge an die Hand gibt und dann unterstützt oder die Tests professionell begleitet. Das wäre auch noch eine Möglichkeit, dass man da

Shared Service Qualitätsmanagement etabliert und dann in der agilen Entwicklung nach Scrum da entsprechend coachen kann und die Tests professionell begleiten kann. Das wäre auch noch eine denkbare Idee. Da fallen mir dann z.B. so Tools wie HP, ALM oder auch Gira mit den entsprechenden Add-ons ein, wo dann das Testing abgebildet wird. Wir haben da z.B. Das Tool Xporter, da können dann entsprechende Testfälle abgebildet werden und man kann auch die entsprechenden Fortschritte darüber einsehen und tracken. Und wenn dann halt das Qualitätsmanagement da unterstützt und auch diese Prozesse als Standard festlegt. Ich glaube, dass kann auf jeden Fall bei der agilen Arbeitsweise Vorteile bringen.

I: Ja definitiv. Wie heißt dieses Tool Xporta?

IP: Ich weiß jetzt nicht ob das Xreporter oder nur -porter heißt. Das ist, glaube ich, ein kostenpflichtiges Add-on für Gira und da können dann ganze Testkonzepte und Testfälle abgebildet werden. Und die Tester können dann... Also man sieht dann mit so einer Art Statusbars, wie der Status der einzelnen Testfälle ist oder welches Modul jetzt z.B. fertig ist. Also da gibt es dann so eine bestimmte Managementübersicht, also ein schönes Dashboard. Und das ist natürlich auch für das Management schön, wenn man dann hört, die und die Tests sind so und so erfolgreich abgeschlossen worden, so viele Fälle sind positiv, so viele sind negativ, bei diesen hat noch nicht herausgefunden. Das ist auf jeden Fall eine sehr gute Hilfe und Konkurrenzprodukte wären HP oder ALM. Also Application Lifecycle Management und da ist dann eben auch Qualität oder Testing ein wichtiger Punkt.

I: Ja das ist sehr interessant, wenn man gerade Richtung Kommunikation zwischen den einzelnen Abteilungen oder während des Sprints in den agilen Arbeitsweisen denkt. Das ist oft das Thema, was etwas zurückbleibt. Das gesagt wird, was ist eigentlich gemacht worden und was für Tests wurden bestanden, welche Testes wurden nicht bestanden. Das man da eben dann noch gezielter angreifen kann und nicht generell einen Katalog von A bis Z durcharbeitet, sondern sagt, was ist denn auch wirklich notwendig.

IP: Genau. Und gerade jetzt bei Gira, dass man da so Dashboards erstellen kann oder auch Statusreports direkt generieren kann, da ist die Kommunikation dann auf jeden Fall auch mit klassischen anderen Abteilungen dadurch gegeben. Das gibt dann auch ein gutes Gefühl, wenn man dann schwarz auf weiß sehen kann, aha die testen wirklich, da kann man sich drauf verlassen. Und ja das ist schon eine wichtige Sache.

## Interview 5

Interviewpartner: Verantwortlicher und Nachfolger für Fehlerbehebungen von Connectivity-Produkten/Services nach Markteinführung im Automobilbereich (Abkürzung: IP - Interviews Partner, I - Ich)

I: Welcher Branche wird das Unternehmen zugeordnet? In welchem Bereich bist du tätig?

IP1: [REDACTED] Ich der Automobilindustrie zugeordnet. Und in welchem Bereich? Ich bin im Bereich Connectivity-Umfeld unterwegs also das heißt, ich bin überall so ein bisschen involviert in die Entwicklung nicht einem bestimmten Zweig.

I: Welche Stellung besetzt du in diesem Bereich und was ist dein Aufgabengebiet?

IP1: Ich habe mehrere Aufgabengebiete. In meiner Stellung bin ich auf der Sachbearbeiter Ebene als Ingenieur ohne Personalverantwortung oder Budgetverantwortung. Mein Aufgabengebiet ist zum einen die Fahrzeuginbetriebnahme und Service-Aktivierung. Das heißt ich schaue, dass die Fahrzeuge, die Konnektivitäts-Dienste bekommen, dass man die auch testen kann. Aber auch das bei bestimmten Fahrevents die Fahrzeuge einfach funktionieren in diesem Konnektivitäts-Bereich. Das heißt ich schaue, dass die Fahrzeuge richtig am Backend angemeldet werden, dass die Dienste aktiv gehen und das die Dienste im Fahrzeug auch erlebbar sind. Das ist der eine Part von meinen Aufgaben der zweite Part ist die Q-Teamarbeit. Das heißt wir bekommen Kundenbeanstandungen rein über bestimmte Prozesse und wenn diese eben dementsprechend priorisiert werden, landen die bei uns im Q-Prozess, wo wird dann den Fehlerabstellvorgang starten, wo wir Fehler dann analysieren, verarbeiten und Lösungsansätze ins Feld bringen.

I: Das heißt von Produkten die bereits vom Kunden benutzt werden?

IP1: Ja genau also quasi die Beanstandungen aus dem Feld. Also das heißt wenn jetzt jemand mit seinen Konnektivitäts Diensten in irgendeiner Form unzufrieden ist dann ist es möglich, dass er sich beschwert, und wenn sich da mehrere Leute beschweren zu einem bestimmten Thema dann landet das bei uns im Q-Team also wird in die Q-Arbeit reineskaliert.

I: Wie lange bist du schon im Zusammenhang mit digitalen Produkten tätig?

IP1: Im Zusammenhang mit digitalen Produkten bin ich nun sei, lass mich überleg, seit zwei bis vier Jahren unterwegs. Und war davor auf der Dienstleister-Seite für die Fahrzeug-Aktivierungs-Themen zuständig. Ich bin jetzt seit Juni letzten Jahres im Automobilunternehmen [REDACTED] tätig. Davor beim Dienstleister, der genau dieselben Aufgaben ausgeführt hat. Und dann bin ich quasi ins Unternehmen eingewechselt, weil sich das so ergeben hat.

I: Was ist dein akademischer Background? Bist du Software-Ingenieur oder hast du mit der Softwareentwicklung bereits Erfahrung?

Nein mit Software habe ich wenig Erfahrung. Also ich kann Code grob lesen, aber ich bin weit davon entfernt Software zu schreiben. Mein akademischer Background: Ich bin Master of Science in Physik. Also ich bin ein Quereinsteiger. In der Automobilindustrie bin ich als Quereinsteiger reingekommen, habe aber auch einen technischen Hintergrund. Vor meinem Studium habe ich technisches Zeichen Richtung Maschinenbau als Ausbildung absolviert. Also von daher habe ich schon eine technische Ader, sage ich jetzt mal.

I: Was für Produkte im Konkreten begleitest du da?

IP1: Im Prinzip begleite ich kein Produkt an sich, sondern quasi das Zusammenspiel von diversen Systemen und Produkten. Zum einen ist am Fahrzeug die TCU, die Communication Unit, das ist quasi das Telefon von dem Auto und auf der and deren Seite haben wir die Back-End Systeme, mit denen das Auto kommuniziert. Und ich bin sozusagen der Mittelsmann, der eben schaut, dass diese Systeme miteinander arbeiten. Das heißt eigentlich alle Konnektivitäts-Dienste, aber eben auf einem sehr High-Level. Also auf einem sehr hohen Level. Das heißt ich schaue nicht in die Tiefe rein. Ich schau nur in die Tiefe rein, wenn ich feststelle, dass irgendetwas schief läuft. Dann analysiere ich und leite an die entsprechenden Verantwortlichen weiter.

I: Das heißt, dass können dann auch Fehler hinsichtlich Software und hinsichtlich Hardware sein die dann dort zurückgemeldet werden und analysiert werden?

IP1: Das sind dann meistens tatsächlich Softwarefehler. Meistens sind das Softwarefehler, die dann hochpoppen, wenn die Systeme anfangen miteinander zu arbeiten.

IP2 kommt kurz zum Meeting dazu, muss aber nochmal gehen.

## Anhang

I: Wir waren stehen geblieben bei Frage 5 und du meinstest, dass die meisten Fehler Software-basierend sind.

IP1: Genau Softwarefehler im Steuergerät oder eben im Backend System, wenn man das grob umschreiben möchte.

I: Das heißt dann aber auch, dass diese Bandbreite an Produkten das gesamte Portfolio begreift und betrifft.

IP1: Richtig ja im Prinzip schon. Also im Backend, um es jetzt mal schwammig auszudrücken, laufen bis zu 150 Microservices, die jeweils ein Produkt abbilden. Also wenn du z.B. den Live Traffic haben willst, dann ist das ein Microservice, der auf unserem Back-End läuft und da gewartet wird und weiterentwickelt wird.

I: In welchem Zusammenhang steht euer Bereich mit der Entwicklung und mit der Qualitätssicherung?

IP1: Kann ich so nicht sagen, weil mir so ein bisschen der Background fehlt. Also es gibt Qualitätssicherungsmaßnahmen. Auf jeden Fall. Also es geht nichts, ohne auf Herz und Nieren geprüft worden zu sein, auf dem Markt. Wir jetzt von der Q-Team-Arbeit aus... Da ist das eben so das aus dem Kundenumfeld die Fehler an uns herangetragen werden und die dann von der entsprechenden Abteilung behandelt und abgestellt werden. Je nachdem wie kurzfristig das möglich ist. Es kann sein, dass sie in zwei Tagen, aber es kann auch sein, dass es in zwei Monaten erst erledigt ist. Beantwortet das deine Frage?

I: Ich habe noch eine Zwischenfrage. Dann ist das praktisch so: Der Fehler entsteht, der Kunde meldet das an und euch, das trifft ein und ihr analysiert das dann und leitet das dann an die Entwicklung bzw. an das Qualitätsmanagement weiter, richtig?

IP1: Also, wenn wir einen Fehler haben und der Kunde beanstandet das dann gibt es mehrere Wege, über die das laufen kann. Es kann eben entweder über die Werkstatt laufen. Also der Kunde fährt in die Werkstatt, ganz klassisch. Also die älteren Herrschaften, die fahren in die Werkstatt und sagen, mein Live-Traffic funktioniert nicht. Ich habe die Live-Traffic Lizenz eingekauft aber die hat es nicht verlängert. Solche Fehler also alles im digitalen Bereich. Oder z.B. der Reifendruck, der mir in der Handy App angezeigt wird, stimmt nicht überein mit dem Reifendruck, der im Fahrzeug angezeigt wird. Das sind so Fehlerbilder, die entstehen können. Und der Kunde hat die Möglichkeit das in der Werkstatt zu melden oder eben über das CAC, Customer Assistance Center. Das heißt, er kann direkt aus dem Fahrzeug anrufen und sich beschweren. Dann nimmt das das CAC auf, überprüft ob es diesen Fehler schon gibt, überprüft ob es dafür eine Sofortmaßnahme gibt, wenn nicht dann läuft das quasi in den Backdesk zur Erstanalyse. Wenn dort keine Zuordnung zu einem Fehlerfall stattfinden kann, wandert das dann eben weiter in den Second-Level-Support und dann landet es im End-Level support, das sind dann die CFOs oder Komponenten-Verantwortlichen und die müssen dann dafür Sorge tragen, dass die Fehler abgestellt werden. Das ist der reguläre Prozess. Es wandert quasi vom Kunden... sagen wir mal es gibt 20000 Beanstandungen pro Monat, was jetzt irgendeine Zahl ist, wandern von diesen 20000 Beanstandungen dann nach dieser Kaskadierung ungefähr 500 bis 1000 Fehler im End-Level, weil die anderen 19000 Fehler davor schon abgehandelt und abgearbeitet werden. Um diese 500 bis 1000 Fehlerfälle da kümmern sich die zuständigen CFOs und POs und KVs.

I: Das heißt, das sind dann die zuständigen Entwicklungsabteilungen?

IP1: Genau, das sind dann die zuständigen Entwicklungsabteilungen oder sogar einzelne Entwickler.

I: Noch mal eine Verständnisfrage: Bei diesem Weg vom Kunden in das System und dann bis hoch zu den CFOs. Das teilt ihr dann auch mit ein, richtig?

IP1: Nein nicht direkt das macht das CAC schon eigenständig und der Backdesk auch. Die haben da die Erfahrung und die Expertise zu sagen, dass der Fehler in dem und dem System liegt und wer dafür verantwortlich ist.

IP2 kommt zum Meeting dazu.

I: Wir sind jetzt bei Frage 7 angekommen. In der Softwareentwicklung kann die Entwicklung wie oben dargestellt teils stark abweichen, da sich die Phasen überlappen oder in inkrementellen Zyklen ablaufen. Stichwort agile Arbeitsweise - Agile Zusammenarbeit/Entwicklung wird schon seit Jahren in der SW-Industrie eingesetzt, in welcher Form/Umfang sind agile Arbeitsformen bei euch etabliert, bzw. wie kommt ihr konkret damit in Berührung?

IP1: Ich antworte mal mit ja. (lacht)

IP2: Wir haben so genannte Releases...

IP1: Wir haben agile Arbeitsformen ja und das entscheidet jedes Team für sich, in welcher Form sie arbeiten.

IP2: Das ist ja agil.

IP1: Also meistens arbeiten sie in einer agilen Form und nicht in einem klassischen Wasserfallmodell, sondern eher in einer agilen Form, wie eben Scrum. Also es ist eigentlich Scrum basiert. Aber die Ergebnisse laufen dann in einzelne Releases zusammen, wo die Ergebnisse dann gebündelt live geschaltet werden und ausgerollt werden.

IP2: Genau, wir haben jetzt 8 Releases im Jahr. Das ist neu.

I: Acht Releases? Davor waren es vier richtig?

IP1: Davor waren es vier, aber jetzt wird umgestellt auf acht Releases pro Jahr.

IP2: Genau. Es gibt quasi für jedes App-Feature, was man entwickeln kann... Da kann sich ein Komponentenverantwortlicher melden und das dann mit einer agilen Projektgruppe erstellen. Das nennt sich dann Customer Feature Owner, also CFO.

IP1: Ja genau und diese werden eben dann ausgerollt, wobei es auch Bestrebungen gibt quasi eine CI/CD-Pipeline also Continuous Integration oder Continuous Delivery aufzubauen in der App-Welt. Also das heißt, in der Connected Car Service App [REDACTED] gibt es Bestrebungen (dazu), weil die App-Welt so schnelllebig ist. Man bekommt die ganze Zeit neue Updates für diverse Apps, wenn man das auf seinem Handy anschaut und deswegen passt das mit diesen acht Releases pro Jahr nicht mehr dazu. Deswegen ist in der App Welt der Zyklus weitaus schneller und wir arbeiten auch daran, dass quasi noch schneller zu machen indem wir quasi die ganze Zeit neue Features oder Updates auf die Produktiv-Umgebung schieben. Weil eine vier Wochen alte App in unserer heutigen Zeit einfach schon zu alt ist.

I: Das heißt das Ziel ist es, dass wie in der Kommunikationsindustrie zu machen? Also wie auf einem Smartphone? Man schickt also kontinuierlich Updates für diese Apps und Dienste raus?

IP1: Ja genau. Das Ziel ist es über kurz oder lang so flexibel zu werden, dass wir quasi sagen können, okay wir haben heute einen Fehler entdeckt, wir fixen das morgen und übermorgen ist das dann live. Also im Prinzip ist die Bestrebung schon dahin, dass wir da so flexibel werden. Allerdings ist das in der Automobilindustrie halt immer... Oder auch im Fahrzeugaufbau ist man eben nicht so flexibel wie bei den Back-End Systemen. Wir haben eine Abhängigkeit zum Fahrzeug und wenn wir für das nächste Fahrzeugmodell [REDACTED] ein bestimmtes Feature drin haben wollen und bestimmte Daten aus dem Fahrzeug brauchen... Das nächste Fahrzeugmodellname [REDACTED] kommt nächstes Jahr auf den Markt, das Neue, dann sind wir jetzt schon zu spät dran um da noch etwas zu reißen. Da waren wir sogar schon vor einem halben Jahr zu spät dran gewesen, um zu sagen, wir brauchen übrigens den und den Wert aus dem Fahrzeug. Die Fahrzeugentwicklung wird auch immer kurzlebiger aber die Fahrzeugentwicklung ist so eine Art Planwirtschaft. Man sagt, wir bringen das Fahrzeug 2020 auf den Markt, das heißt, wir müssen anfangen mit der Komponentenentwicklung zweieinhalb Jahre davor, der Software oder Change Freeze ist eineinhalb Jahre davor und dann müssen wir eben schauen, dass wir das Fahrzeug fehlerfrei bekommen. Also Fahrzeuge sind nicht so flexibel, dass man da agil die Software implementieren kann. Die Software wird agil entwickelt, aber wir müssen natürlich schauen, dass die Komponente A und die Komponente B miteinander reden können und da gibt es einen bestimmten Releasezyklus, wo wir bestimmte Software-Versionen zusammenbringen müssen.

IP2: Hast du denn auch noch Interviews mit Release- und Roll-Out Teams?

I: Ich hatte ein Interview mit einem Test-Manager im Entwicklungsbereich. Release selbst noch nicht nein. Wäre aber bestimmt interessant.

IP2: Das würde ich dir auch noch raten, da wir hier nicht die Experten sind, würde ich mich noch mal an die wenden.

IP1: Es gibt auf jeden Fall ein Releaseprozess und es gibt zwei Fraktion, die einen, die lieben und leben ihn, und die anderen, die hassen ihn. Eigentlich wird die Idee von agil ausgebremst, weil sie sagen, ne es gibt ein Release bis dahin müssen die und die Features gebracht werden, damit wir das dann Online stellen können. Und das bremst das Ganze dann auch schon etwas aus. Aber wenn du mit denen reddest, werden sie sagen, dass das natürlich notwendig ist. Und ich sehe das auch als notwendig an, dass man einen großen Plan hat, den man verfolgt und nicht hier mal eine App aufmacht und da mal eine App auf den Markt schmeißt und dann feststellt, blöd das holen wir wieder zurück und machen ein Update. Das ist schön und gut, aber in so einem großen Umfeld schwierig komplett nach Plan zu agieren.

IP2: Das Problem ist auch bei uns – Vielleicht kannst du dich da auch nochmal zu informieren. Also die Rolle des Komponentenverantwortlichen in dem Ganzen, denn der verantwortet ein Feature von der Idee bis zum Ende in einem Sprint, aber was danach passiert das betrifft dann die Qualitätsprozesse. Das ist so ein bisschen das, wo gerade Schwierigkeiten herrschen, weil die viel mit ihren Tickets selbst abarbeiten können,

aber ganz schwerwiegende Fälle, die bekommen wir dann und da sehen die in ihrer Rollenverantwortlichkeit oftmals gar nicht, dass sie das auch machen müssen. Also die denken oft auch gar nicht bis zum Kunden, weil die so in ihrer Software-Welt sind. Also das ist mein Eindruck, seitdem ich hier bin. Und das haben sie auch gesagt, dass sie da auch ein bisschen überfordert sind mit den Sachen, die sie verantworten. Und das ist auch etwas, wo wir ansetzen wollen. Denn es gibt eben diesen Qualitätsprozess, der aber für das Automobilunternehmen [REDACTED] generell für die Produktion geschaffen ist. Und es ist ein bisschen schwierig, den da einzubauen, also das zu vereinbaren, weil es so viele Bugfixing Systeme gibt und in der Software das so schnellebig ist und es oftmals eben... Also die Nähe zum Kunden da manchmal fehlt. Und das soll ja eigentlich mit agil geschaffen werden, aber naja das ist gerade nicht so. Das ist mein Eindruck. Ich möchte das jetzt auch nicht zu sehr feststecken. Aber das ist das, was ich mitbekommen habe und ich habe auch vorher mit agilen Sachen viel gearbeitet.

I: Das heißt diese Komponentenverantwortliche... Komponentenverantwortlicher ist dann ja erstmal bisschen fehlleitend oder? Ist das dann einer, der sich eher um die Hardware kümmern oder hat er dann auch mit der Software zu tun?

IP2: Komplett mit Software.

IP1: Also Software ist in diesem Sinne auch eine Komponente.

IP2: Also stell dir vor, es gibt bei der Connected Car Service App [REDACTED] Features, z.B. haben wir ein ganz spannendes Thema Incar/Office, dass man in seinem Auto vorne auch sein Outlook-Terminkalender hat und alles Mögliche. Das hat sich einer oder eine überlegt und die hat das dann quasi entwickelt und dann auch ausgerollt, also von Anfang bis Ende dafür verantwortlich. Dann gibt es da auch Dinge, die beachtet werden müssen, da das dann auch weltweit ausgerollt wird.

I: Ja okay alles klar, und wenn dann praktisch diese Fehler zurückkommen, dann ist es nicht wirklich geklärt, wie die das Regeln sollen bzw. ob die das Regeln sollen.

IP1: Die CFOs sehen das nicht mehr in ihrer Verantwortung also da ist oftmals eine sehr starke Abneigungshaltung mit der Aussage, ja ich entwickle und die Operations ist nicht mein Bier. So auf diese Art. Aber das stimmt nicht. Wenn ich eine Pizza ausliefere, bin ich auch für die Qualität der Pizza zuständig.

IP2: Aber da ist eben wie gesagt, das Problem der Rollenklarheit auf der einen Seite und auf der anderen Seite, dass es sehr viele, verschiedene Prozesse gerade gibt. Also gerade Bugfixing Systeme, die auch so schon genutzt werden. Wo der normale Fehlerabstellprozess schwierig ist, das zu integrieren. Aber deswegen wollen wir deine Arbeit lesen. (lacht)

I: Ja mal schauen, in welcher Form ich euch da helfen kann bzw. auch Verbesserungsmöglichkeiten herausarbeiten kann. Jetzt sind wir etwas abgekommen, was aber eigentlich auch ganz gut ist, da das Themen sind, die euch beschäftigen, die ich oftmals mit den Fragen so gar nicht adressieren kann, weil ich ja nicht im Unternehmen bin und von einem Blick außerhalb darauf schaue. Genau, ich muss mich nun kurz etwas sammeln und schauen wo ich weiter machen.

IP2: Also wir haben jetzt noch zehn Minuten

I: (lacht) Du machst es jetzt nicht besser.

IP1: Also ich würde vorschlagen, wir machen mal mit Punk 8 weiter. Die Schwierigkeit aus unserer Erfahrung, welche durch agile Arbeitsweisen, nachgelagertes Qualitätsmanagement und Schadensteilanalyse entstehen... Das haben wir gerade ein bisschen umschrieben, was da die Probleme sind. Durch diese agile Arbeitsweise ist natürlich diese Schnellebigkeit da. Man sagt, okay ich mache heute A und morgen das fertig und übermorgen mache ich B und so weiter. Und man übersieht eben ganz leicht, dass wir ja von einem Fahrzeuglebenszyklus auch reden. Wenn wir heute ein Fahrzeug verkaufen, dann ist das auch noch in zehn Jahren auf der Straße und was machen wir mit den Features, die wir da ins Fahrzeug reingebracht haben? Die müssen auch noch in zehn Jahren funktionieren. Und ein Fahrzeug ist eben auch nicht so flexibel und so agil wie ein Smartphone. Aktuell ist das zumindest so. Und deswegen ist diese Zeitspanne, die man betrachten muss einfach viel, viel länger. Weil am Ende des Tages verkaufen wir immer noch Autos und wenn diese Autos in 10 Jahren immer noch fahren, müssen wir auch sicherstellen, dass alle Dienste funktionieren. Dann müssen wir auch dafür Sorge tragen, dass die Konzepte, die wir heute entwerfen und morgen im Auto auftauchen, eben auch immer noch in mehreren Jahren da sind und funktionieren. Und das ist ein bisschen etwas, das in dieser agilen Arbeitsweise verloren geht. Dieser Weitblick. Denn man denkt eben immer nur in diesen Zwei-Wochen-Zyklus, denn ein Sprint dauert zwei Wochen. Man denkt von Sprint zu Sprint und sagt, ok in den nächsten zwei Wochen mache ich das, in den nächsten zwei Wochen mache ich das und dann rollen wir es aus und dann mache ich bisschen Verbesserung und dann mache ich noch ein bisschen Verbesserung und dann rollen wir das wieder aus. Aber das macht man ja nicht die nächsten

10 Jahre. Und das ist so ein bisschen das Gefühl. Dass die Software, die wir entwickeln, am Lebenszyklus des Fahrzeugs hängt, das fehlt ein bisschen meiner Meinung nach. Und es ist aber auch schwierig das sicherzustellen, dass das in zehn Jahren auch immer noch funktioniert. Also wenn ich mir heute ein Fahrzeug [REDACTED] von 1980 kaufe dann wird der noch funktionieren und alles laufen. Alle Teile funktionieren. Aber wenn ich mir in 20 Jahren ein Fahrzeug [REDACTED] von 2020 kaufe, dann ist das nicht mehr sichergestellt, weil wir so viel Verbindungsumfänge zur Software und zum Backend haben, die in 20 Jahren nicht mal mehr am Netz sind. Deswegen ist diese agile Arbeitsweise mit Vorsicht zu genießen, weil wir eben an diese Lebensspanne vom Fahrzeug denken müssen.

I: Ja was halt ganz anders ist als bei einem Telefon bzw. Mobiltelefon, richtig?

IP1: Ja eben. Bei einem iPhone hat man eigentlich jedes Jahr ein neues Modell und spielt auf die alten Modelle ein Software-Update auf. Aber mit dem iPhone fährst du nicht die Straße lang und das ist eben auch noch ein Thema. Das Fahrzeug wird zertifiziert, bevor es auf den Markt kommt. Und dann ist es so für die Straße zugelassen. Sobald du größere Softwareänderung vornimmst, hast du ein Problem, weil dann musst du es neu zertifizieren. Und das ist eben bei einem Smartphone oder bei irgendeiner Software, die auf dem Backend läuft, nicht der Fall. Das ist nicht sicherheitsrelevant. Es hängen eben einfach auch Menschenleben davon ab, ob das Fahrzeug dann immer noch sicher ist, wenn wir die Software ändern. Vor kurzem war in den Schlagzeilen hier, dass ein Tesla gehackt wurde und auf die Gegenfahrbahn geführt worden ist. Das ist einfach diese Software und Konnektivität, da gibt es einfach ein großes Einfallsfenster für mögliche Risiken und das müssen wir immer im Hinterkopf behalten. Diese Fenster müssen in 20 Jahren auch noch zu sein. Aber da arbeitet die Autoindustrie auch daran, diese Softwareentwicklung mit reinzubringen auch dieses agile ins Fahrzeug mit reinzubringen. Z.B. Abgasreinigung ist auch ein gutes Beispiel. Die Abgasreinigung im Nachhinein quasi abschalten über Remote. Das könnte man machen, aber dann ist das Fahrzeug nicht mehr für die Straße zugelassen und nicht mehr fahrtauglich. Da muss man sehr aufpassen meiner Meinung nach.

I: Gerade bei der Zusammenarbeit mit Lieferanten gibt es da auch zusätzliche Probleme, wenn z.B. die Lieferanten einen Dienst entwickeln und den dann an euch liefern? Das es da eben Fehler gibt, die vom Kunden gemeldet werden?

IP1: Naja da muss der Lieferant eben auch nachliefern. Aber jetzt ist eben die Frage, wir bringen heute das Fahrzeug auf dem Markt aktuell noch eine Beauftragung mit diesem Lieferanten in drei Jahren stellen wir fest, da passieren Fehler mit diesem Softwaremodul, aber der Lieferant ist überhaupt nicht mehr da. Da arbeiten wir natürlich sehr viel mit Produkthaftung und so weiter. Da muss man eben auch schon schauen, dass das abgesichert ist. Also wir haben Lieferanten, weil wir die Entwicklungsarbeit auch nicht allein stemmen können. Eben darum, dass wir eben Autos bauen eigentlich und diese Softwareentwicklung erst vor fünf bis zehn Jahren dazu kam. Und da ist eben... Jetzt habe ich den Faden verloren. Da wir mit vielen Entwicklern zusammenarbeiten, schaffen wir uns eben eine Abhängigkeit. Aber diese Abhängigkeit betrifft uns nicht nur ein bis zwei Jahre, sondern den kompletten Lebenszyklus. Wir werden auf keinen Fall einen Lieferanten für die nächsten 20 Jahre beauftragen, dass er mal da ist, wenn Fehler passieren. Das kann er auch nicht im Sinne des Erfinders sein, sonst würde ich als Lieferant ja auch immer Fehler einbauen, damit ich weitere Beauftragung bekomme. Es gibt natürlich Qualitätsvereinbarungen, die erfüllt werden und wir versuchen natürlich alles fehlerfrei zu machen, aber wir dürfen das Know-how natürlich nicht aus dem Unternehmen ziehen lassen. Also da müssen wir auch drauf achten, dass das Know-how auch da ist, wenn es mal benötigt wird. Und nicht an einem Lieferanten abhängt.

I: Genau, weil gerade, wenn dann auch Fehler passieren und diese abgestellt werden müssen, wenn es dafür kein internes Know-how gibt, ist es ja auch auf lange Sicht dann auch wirklich problematisch diese abzustellen, wenn man dann auch immer vom Lieferanten abhängig ist.

IP1: Ja.

I: Die Frage 10 haben wir bereits geklärt innerhalb des Gesprächs.

IP1: Es gibt Life Management Tools von Software, wo der komplette Lebenszyklus abgedeckt wird und getrackt wird. Welche Fehler entstehen, welche Fehler werden abgestellt. Ich glaube das ist so die Frage 10 kurz beantwortet.

I: Dann würde ich zur Frage 11 gehen. Wo seht ihr denn die bedeutendsten Unterschiede von Hardwarekomponenten und eben aber auch Softwarekomponenten im Vergleich zu den Connectivity Produkten?

IP1: Ja das hatten wir vorher schon ein bisschen ausdiskutiert mit dem, dass die Hardware eben in 20 Jahren noch rumfährt. Die Software ist da agiler und schneller umzusetzen und wie gesagt, es gibt Entwicklung in die Richtung das wir Software auch nachträglich noch over-the-air in das Fahrzeug

reinbringen. Tesla ist da ein guter Vorreiter, die machen das quasi schon die ganze Zeit. Die spielen die ganze Zeit die Softwareupdates auf das Fahrzeug drauf. Es ist natürlich nicht bei allen Teilen möglich diesen agilen Arbeitsweg zu gehen. Wie gesagt, bei sicherheitsrelevanten Teilen haben wir natürlich auch zusätzliche Hürden, die wir einhalten müssen. Bei sicherheitsrelevanten Teilen, bei umweltrelevanten Teilen usw., deswegen ist da... da sind so die Unterschiede und die Risiken, dass wir eben zu viel ändern, so dass das Fahrzeug keine Betriebserlaubnis mehr hat. Und dann geht es eben um Retouraktionen usw. Pipapo. Das ist extrem teuer und kostspielig.

I: Wir hatten es vorher auch schon mal angesprochen bei der Frage 12... dieses Release Management. Trotzdem aber nochmal. Wie verläuft denn da die Zusammenarbeit genau bei der Integration von Softwarekomponenten, ob diese nun von der Entwicklung oder auch von Lieferanten gestellt werden, in die Hardware-Komponenten?

IP1: Naja also die Lieferanten, die werden von den CFOs oder Product Ownern oder KVs eben beauftragt, etwas zu entwickeln und dieses irgendwas wird eben... Man stellt einen Bedarf fest, den der Kunde haben wird. Also man muss etwas Voraus denken und sagen, in zwei Jahren ist der Bedarf wahrscheinlich so und so, also entwickeln wir uns in die Richtung. Und da wird festgestellt dieses und dieses Feature haben wir noch nicht, das brauchen wir, bis wann brauchen wir es, welche Entwicklungsschritte sind notwendig, welche Hardware benötigen wir und so weiter. Ja und dann werden diese Schritte eingeleitet und durch ein Release Management getrackt und es wird geschaut, dass das ineinander übergeht und das praktisch alles eingehalten wird.

I: Bei Frage 13: Das hatten wir jetzt auch schon teilweise wie eben die Qualität von digitalen Produkten sichergestellt wird. Gut ein eigener Bereich ist es ja nicht sondern es sind ja mehrere Bereiche, die das tun.

IP1: Ja.

## Interview 6

**Interviewpartner: Product Owner für einen digitalen Fahrzeugsuch-Service eines Software Unternehmens, welches Software- und Connectivity Produkte für die Automobilindustrie entwickelt**

(Abkürzung: IP - Interviews Partner, I - Ich)

I: Welche Branche wird ihr Unternehmen zugeordnet? In welchem Bereich sind Sie tätig? Welche Produkte werden in ihrem Bereich/Unternehmen hergestellt?

IP: Branche: Digital Business.

Bereich: eCommerce

Produkte: Digitale Produkte für die Automobilbranche und verschiedene (Mercedes-Benz) Länderauftritte weltweit

I: Welche Stellung besetzen Sie in ihrem Bereich?

IP: Stellung = Product Owner

I: Wie lange sind Sie bereits in der Entwicklung für SW-Produkte tätig? Haben Sie bereits Erfahrungen in anderen Branchen/Unternehmen gemacht? Sind Sie SW-Entwicklungs-Ingenieur oder haben Sie einen anderen akademischen Background?

IP: Erfahrung im Bereich SW-Produkte: 5 Jahre

Anderer Bereich: klassische Werbung Automobilindustrie / Marketing & Sales im Verlagswesen

Akademischer Background: Bachelor of Arts: Information & Communication Management, Master of Arts: Electronic Media (Corporate Communication)

I: Welche Produkte begleiten Sie in der Entwicklung in Ihrem Unternehmen? Sehen Sie essentielle Unterschiede/Besonderheiten zu anderen SW-Produkten?

IP: Produkt: Fahrzeugsuche

Unterschiede / Besonderheiten: keine

I: Wie sind die Bereiche Entwicklung und Qualitätssicherung in die Unternehmensorganisation eingegliedert? Sind es eigenständige Funktionsbereiche? Oder ist das Qualitätsmanagement in die Entwicklung integriert?

IP: Wir arbeiten agil – dementsprechend ist das gezeigte Schaubild nicht ganz valide. Die Qualitätssicherung der Produkte wird bei uns jeweils im Produkt, durch die Entwickler und Quality Assurance Engineers und Produktübergreifend in den entsprechenden Circles (responsive Unternehmensorganisation) durchgeführt und standardisiert.

I: Bei der Entwicklung von Produkten mit Schwerpunkt Software werden traditionell die im Schaubild dargestellten Phasen durchlaufen. Die Phasen bilden ebenfalls den Lebenszyklus eines Produktes von der Erstellung bis hin zu der Außerbetriebnahme/Recycling ab.

In der Softwareentwicklung kann die Entwicklung wie oben dargestellt teils stark abweichen, da sich die Phasen überlappen oder in inkrementellen Zyklen ablaufen. Stichwort agile Arbeitsweise - Agile Zusammenarbeit/Entwicklung wird schon seit Jahren in der SW-Industrie eingesetzt, in welcher Form/Umfang sind agile Arbeitsformen in ihrem Unternehmen in der Entwicklung etabliert?

IP: Bei uns im Unternehmen (MBio) wird hauptsächlich agil gearbeitet. In Anlehnung an SCRUM müssen wir jedoch Abstriche machen, da die Arbeit im Großkonzern ein 100%ige agile Arbeitsweise nach Scrum nicht immer zulässt. Wir haben aber ein starkes agiles Mindset unter allen Kollegen/Kolleginnen des Unternehmens (der MBio).

I: Mit welchem Vorgehensmodell bzw. Projektmanagementmethode werden SW-Produkte entwickelt? Wie/aus welchen MA-Kompetenzen setzt sich ein Projektteam zusammen? Gibt es eine klare Rollenverteilung? Sind alle an der Entwicklung eines Softwareprodukts Beteiligten im Unternehmen oder gibt es Lieferanten/etc.?

IP: Agil in Anlehnung an Scrum, in manchen Themen mit Kanban.

## Anhang

Teamzusammenstellung:

Product Owner

Software Entwickler

Konzept / Design

Quality Assurance

Scrum Master

Zusätzliche Rolle: Product Manager (als Counterpart zum Product Owner – Aufgaben: Stakeholder Management, Legal Abstimmungen, ...)

Teilweise zusätzlich: IT Consultant, Business Consultant

Alle Rollen werden im Unternehmen abgedeckt. Die Rolle Product Owner und Product Manager können sowohl im Unternehmen (von der MBio) als auch vom Automobilunternehmen (DAG) übernommen werden. Lieferanten werden nur hinzugezogen, wenn die Ressourcen nicht durch uns (MBio) abgedeckt werden können.

I: Wie werden Kundenanforderungen an das Produkt ermittelt? Zu welchem Zeitpunkt findet die Anforderungsaufnahme statt? (Kontinuierlich oder einmalig) Wer steht im Austausch mit dem Kunden?

IP: Anforderungsermittlung (kontinuierlich):

Feedback aus User Tests

Ideen aus dem Produktteam

Ideen aus den Märkten und Regionen

User Testing wird durch das Produktteam angetriggert und durchgeführt. Im Austausch mit dem Kunden stehen somit die Teammitglieder und die Agenturen, die das User Testing umsetzen.

I: Wie verläuft die Zusammenarbeit hinsichtlich der Integration der erstellten Softwarekomponenten in die Hardwarekomponenten? An welcher Stelle sehen Sie im aktuellen Prozess die größten Problematiken/Herausforderungen?

IP: n/a – da die Produkte nur digital im Web genutzt werden und nicht im Auto selbst

I: Wie stellen Sie in Ihrem Unternehmen die Qualität der SW Produkte sicher? Gibt es hierfür einen eigenen Bereich, welcher dem Qualitätsmanagement/Qualitätssicherung in Hardware-produzierenden Unternehmen gleichkommt? Wenn nicht, welche Methoden und Prozesse stellen Qualität sicher?

IP: Siehe Antwort Frage 5

I: An welchen Stellen entlang des Zyklus gibt es Schnittstellen zwischen Entwicklung und dem QM Bereich für SW-Produkte?

IP: QA direkt im Team. Kontinuierliche Schnittstelle und Zusammenarbeit mit den Entwicklern

I: In welcher der oben genannten Lebenszyklus-Phasen übernehmen Entwicklungsingenieure Qualitätssichernde Maßnahmen? (Bsp. In Form von Testing oder Erstellung automatisierter Tests) Wer übernimmt die Verantwortlichkeit?

IP: Ab Implementierung fortlaufend. Verantwortung im Team, bei der Rolle QA und letztendlich beim PO.

I: Sehen Sie die agilen Methoden/Zusammenarbeit in Ihrem Unternehmen als ausreichend von der Unternehmenskultur unterstützt? Welche Art der Kultur ist aus Ihrer Sicht notwendig um agile Arbeitsweisen erfolgreich zu etablieren?

IP: Ja. Agiles Mindset notwendig in allen Hinsichten, da wir eine responsive Organisation sind, ist auch hier eine gewisse Agilität vorhanden.

Notwendige Kultur:

Agilität bedeutet nicht unstrukturiert vorzugehen

Agilität bedeutet Fail & Learn Culture

## Anhang

Agilität bedeutet klein anzufangen, aber immer mit Release Fokus

I: In welcher Form werden Mitarbeiter in der Abteilung/Team gefördert, weitergebildet und zum Lernen angeregt?

IP:

Enablement Circles mit agilem Fokus für Scrum Master und Product Owner

Zertifizierungsmöglichkeit für Scrum Master und Product Owner

Helping Mindset in der Firma: Fragen können immer gestellt werden

Scrum Master haben den „Auftrag“ die Agilität im Unternehmen zu coachen und zu supporten

I: Welche Methoden und Werkzeuge wurden zum Zwecke ständiger Verbesserung in ihrer Abteilung/Team etabliert? Fließen Verbesserungsbestreben in die Arbeitsweise mit ein?

IP: Regelmäßige Retrospektiven

Sprint

Spezifische Themen

Team Ebene

Cross Teamebene

Organisationsebene

I: Worin sehen Sie die größte Herausforderung, wenn es darum geht traditionelles Qualitätsmanagement mit agilen Arbeitsweisen zu verbinden?

IP: Kann ich leider nicht beantworten

## **Anhang 23: Interviews Ergebnisse Tendenz zu neuen Kooperationen zwischen Automobilherstellern und Software-Start-ups aus dem Bereich Connectivity**

### **Interview 1 Einkauf im Bereich digitale Services eines OEMs**

Mündliches Experteninterview, am 21.03.2019, 32 min - Inhaltliche Zusammenfassung:

#### **Kooperationsmotive im Bereich Connectivity:**

Im Einkauf steht die Suche nach geeigneten Lieferanten für digitale Dienste im Fahrzeug im Mittelpunkt. Der Zugang zu neuen Technologien und Innovationen ist in diesem Bereich als Hauptmotiv zu nennen. Weiterhin spielt die Erhöhung der Entwicklungsgeschwindigkeit (Time-to-Market) und die Vermeidung von hohen Overheadkosten bei alternativen Kooperationen mit Großunternehmen eine Rolle.

#### **Qualitätssicherung innerhalb der Kooperation:**

Grundsätzlich werden bei Produkten, welche in das Fahrzeug kommen, Qualitätsprozesse ausgelöst. Allerdings sind Qualitätsthemen in diesem Bereich sehr vielschichtig und in hohem Maße von der Dienstleistung des Start-ups abhängig. Bei reinen Software-Produkten findet eine Abnahme des Produktes statt. Explizite Maßnahmen finden während der Entwicklung nicht statt. Die Umsetzung von Qualitätsthemen erhöht sich bei langfristigen Kooperationen (Stichwörter: Audit, IT-Security).

Zusammenfassend lässt sich sagen, dass die Laufzeiten und der Umfang von Kooperationen kaum quantifizierbar sind, weshalb sich Qualitätsthemen oftmals abhängig von diesen Faktoren kooperationspezifisch entwickeln.

#### **Differenzen in der Unternehmenskultur:**

Grundsätzlich sind die Kulturen sehr verschieden, insbesondere die gelebten Prozesse. In der konkreten Zusammenarbeit sind überwiegend innovative Bereiche seitens der OEMs beteiligt. Entsprechend sind die Unterschiede der Arbeitsweisen von Start-ups und diesen Bereichen weniger stark ausgeprägt. Generell ist eine gute Kommunikation ein Schlüsselfaktor, um diese Barrieren zu überwinden.

Insbesondere in der Vertragsgestaltung existieren rechtliche Barrieren. Bei OEMs lösen Verträge große Prozesse aus. Sofern ein Start-up bestimmte Vertragsbedingungen nicht akzeptiert, muss an dieser Stelle nachgearbeitet werden. Juristische Ressourcen und Erfahrungen sind hier oft ein Hindernis für Start-ups. Teilweise führt dieser Umstand oder ein fehlendes Vertrauensverhältnis dazu, dass es nicht zum Vertragsabschluss kommt. Start-ups fällt es oft schwer die Sichtweise eines OEMs nachzuvollziehen. Einige Start-ups lösen das Problem, durch das Einstellen eines Experten, welcher bspw. langjährige Erfahrungen im Automotive Bereich hat und die Sorgen und Nöte von solchen Unternehmen kennt.

#### **Welche Faktoren erschweren die Kooperation:**

Neben den zuvor genannten Punkten sind hier insbesondere Qualitätsthemen ein Hindernis. Ein großes Problem ist fehlendes Knowhow von Start-ups zu bestimmten IT-Themen (IT-Security, Free/Open Source Software, Cloud-Themen). Viele Start-ups haben in diesen Bereichen keine Kenntnis bezüglich der Richtlinien und Vorgaben von OEMs.

Darüber hinaus setzen sich einige Start-ups nicht sorgfältig genug mit Vertragsinhalten auseinander.

**Kritische Erfolgsfaktoren:**

Diese Faktoren sind in hohem Maße von der Kooperation und des Projektes abhängig.

**Wünsche für zukünftige Kooperationen:**

Insbesondere bei sehr kleinen Start-ups kommt es vor, dass Kenntnisse zu Grundlagen fehlen, bspw. in der Angebotserstellung. Das ist allerdings eher die Ausnahme.

Grundsätzlich wäre es wünschenswert, wenn Start-ups ein Verständnis für bürokratische Hürden von OEMs entwickeln.

Teilweise forcieren Gründer eines Start-ups einen lukrativen Exit. Hier ist bspw. ein potenzieller Verkauf des Unternehmens an einen Konkurrenten, bei gleichzeitiger Kooperationsbeziehung, sehr problematisch.

**Interview 2 Teamleiter im Bereich IT-Innovationen eines weltweit agierenden Einzelhandelsunternehmens**

Schriftliches Experteninterview, am 17.02.2019, Originalinhalt:

**Kooperationsmotive:**

Es geht hauptsächlich um den Zugang zu neuen Ideen und Technologien. Ein weiteres Motiv sind Geschwindigkeitsvorteile aufgrund fehlender Governance seitens der Start-ups.

**Qualitätssicherung innerhalb der Kooperation:**

Durch ein enges Projektmanagement, welches durch den Corporate aufgesetzt wird. Due Dilligence durch Venture Capital oder ähnliche Investitionen.

**Differenzen in der Unternehmenskultur:**

Der wichtigste Punkt ist eine Kompromissbereitschaft von beiden Seiten. Weiterhin hilft es ein gemeinsames Verständnis für die Kooperation zu entwickeln, um so kulturelle Differenzen überwinden zu können.

**Welche Faktoren erschweren die Kooperation:**

Oftmals sind interne Regeln und Strukturen in Start-ups ein Problem. Darüber hinaus fehlt vielen Start-ups ein Verständnis für den Corporate und damit die notwendige Geduld.

**Kritische Erfolgsfaktoren:**

Wir haben mit einem Single Point of Contact seitens des Corporates gute Erfahrungen gemacht. Das Commitment des Start-ups für die Kooperation ist entscheidend. Kompetenter Sponsor auf Seiten des Corporates.

**Wünsche für zukünftige Kooperationen:**

Vor allem neue Ideen und Impulse und wie zuvor erwähnt, ein hohes Commitment.

### **Interview 3 Mitarbeiter im Bereich Innovations- und Kooperationsmanagement eines OEMs**

Mündliches Experteninterview, am 04.04.2019, 25 min, Inhaltliche Zusammenfassung:

#### **Kooperationsmotive im Bereich Connectivity:**

Die Entwicklung von Innovationen mit etablierten Akteuren aus der klassischen Automobil- und Zulieferbranche gestaltet sich aus mehreren Gründen schwierig. Somit ist als ein zentrales Kooperationsmotiv der Zugang zu Technologie und Innovationen zu nennen. Außerdem lassen sich durch die Kooperation mit Start-ups Zeit- und Kostenvorteile realisieren. Langfristig betrachtet spielt auch die Übernahme von guten Start-ups auf der Agenda.

#### **Qualitätssicherung innerhalb der Kooperation:**

Hier spielt die Organisationsentwicklung eine wichtige Rolle. Weiterhin prüfen Fachbereiche Software auf Code-Ebene (insbesondere bei Connected Software). Es werden bestimmte IT-Richtlinien und Dokumentationspflichten vorausgesetzt. Gerade bei sicherheitskritischen Systemen sind weitere Faktoren zu beachten. Hier kann bspw. die Dokumentation von personenbezogenen Daten zugriffsberechtigter Personen genannt werden. Zusammenfassend kann man sagen, dass OEMs viele Anforderungen stellen, um so eine gewisse Kontrolle garantieren zu können.

#### **Differenzen in der Unternehmenskultur:**

Der zentrale Faktor ist ein beidseitiges Verständnis. OEMs müssen verstehen wie Start-ups denken und andersherum. Wobei anzumerken ist, dass der Kulturwandel in einem Großunternehmen sehr schwer ist.

Die Kommunikation sollte offen und auf Augenhöhe geführt werden. Wichtig sind auch ein fachliches und faires Feedback.

Ein Bindeglied zwischen den zwei Welten ist an dieser Stelle eine große Hilfe.

#### **Welche Faktoren erschweren die Kooperation:**

Große Hindernisse sind:

- Fehlendes intellektuelles Kapital
- Fehlende Manpower
- Preis

Intern betrachtet sind lange Prozesse, wie bspw. der Einkauf und egoistische Verhaltensweisen.

#### **Kritische Erfolgsfaktoren:**

Ein interner Sponsor mit ausreichend Budget und ein hohes Commitment.

Wünsche für zukünftige Kooperationen:

Start-ups sollten besser informiert sein und sich dafür bspw. mit Insidern austauschen. Start-ups sollten realistisch bleiben und nicht Zuviel versprechen. Generell sollten alle beteiligten ihre Egos ausschalten.

## **Interview 4 Projektmanager im Bereich Kooperationsmanagement eines OEMs**

Schriftliches Interview, am 05.04.2019, Originalinhalt:

### **Qualitätssicherung innerhalb der Kooperation:**

Was ich allerdings allgemein sagen kann ist, dass Kooperationen mit Startups allgemein keinen strikten generellen Qualitätssicherungsmaßnahmen unterliegen. Es gibt hier keinen einheitlichen Prozess. Erfolge und Misserfolge werden individuell behandelt und im Rahmen von Meilensteinplänen abgearbeitet und bewertet.

### **Differenzen in der Unternehmenskultur:**

Unternehmenskulturen (OEM vs. Startup) sind durch gezielte Anbindung und Vorbereitung durch Vermittlung von Gesprächspartnern, Mitarbeiteraustauschprogrammen, aber auch durch das tägliche Geschäft, durchaus handlebar. Die Diskrepanzen sind, durch eine Fokussierung und Anbindung auf den Kreis der Willigen, gar nicht so unüberwindbar wie das „draussen“ immer dargestellt wird.

### **Kritische Erfolgsfaktoren:**

Erfolgsfaktoren sind dabei die Klassiker „Umsatz- und Gewinnentwicklung der Geschäftsfelder“, „Mitarbeiterbefragungen- Zufriedenheit zum Projektfortschritt“ usw.

## **Interview 5 IT-Manager Unternehmensberatung**

Mündliches Experteninterview, am 05.04.2019, 46 min, Inhaltliche Zusammenfassung:

### **Kooperationsmotive:**

Als Hauptmotiv würde ich die Erweiterung von Intellectual Property benennen. Weiterhin spielt die Aneignung der von Start-ups gelebten agilen Kultur eine Rolle. Weitere Aspekte sind

- Time-to-Market
- Erweiterung des Service-Portfolios
- Förderung von Geschäftsmodellinnovationen

### **Qualitätssicherung innerhalb der Kooperation:**

In vielen Fällen wird für die Kooperation eine Person für die Qualitätssicherung abgestellt (Dedizierte Rolle). Grundsätzlich steigen entsprechende Maßnahmen, wenn die Kooperation langfristig und größer wird (Organisationsentwicklung). Oft werden gemeinsame Teams gebildet, wobei auch die Entsendung eine Rolle spielt.

### **Differenzen in der Unternehmenskultur:**

Zunächst ist es wichtig, dass beide Seiten kompromissbereit sind, um gemeinsame Wege für die Kooperation zu finden. Oftmals verhindern die Organisationsstrukturen von etablierten Unternehmen ein schnelles Handeln. Etablierte Unternehmen sind historisch gewachsen, weshalb es oft zu größeren Verzögerungen kommt. Das liegt vor allem an zu geringen Entscheidungsberechtigungen von Projektbeteiligten aus etablierten Unternehmen. Ein gemischtes Team kann diese Differenzen durch den Aufbau eines gemeinsamen Verständnisses überwinden. Dafür sollten einfache Regeln für die Zusammenarbeit besprochen werden.

Auf Seiten der großen Unternehmen führt ein schlechtes Wissensmanagement oftmals zu Verzögerungen. Oftmals ist der Zugriff auf vorhandenes Expertenwissen in großen Unternehmen sehr zeitintensiv (Kopfmonopole).

**Welche Faktoren erschweren die Kooperation:**

Zunächst stellt sich oft die Frage, wie die „neuen“ Mitarbeiter eingegliedert werden (Räumlich, rechtlich, strukturell, Steuerung). Meiner Erfahrung nach, bringt eine physische Zusammenarbeit große Vorteile mit sich, insbesondere aufgrund der so gewährleisteten Kommunikation der Partner.

Während der Kooperation machen sich große Differenzen in den Vorgehensweisen und Organisationsstrukturen (Starr vs. Agil) bemerkbar.

**Kritische Erfolgsfaktoren:**

Aus Sicht von Projektbeteiligten aus etablierten Unternehmen ist ein starkes Management-Commitment notwendig. So haben die Akteure mehr Freiheiten und weniger Unsicherheit.

Als zweiten Punkt lässt sich die Kommunikation nennen, hier sollten geeignete Tools und Methodiken genutzt werden. Außerdem sollten beide Partner eine große Offenheit mitbringen.

In der Praxis wurden die Kooperationen oft zu stark durch organisatorische Merkmale von großen Unternehmen überlagert. Ein Beispiel ist hier eine zu starke „Meetingkultur“. Entsprechende Maßnahmen sollten im Sinne der Agilität bewusst auf ein notwendiges Minimum reduziert werden.

Vorab sollte alle Kooperation unter der Berücksichtigung von strategischen Zielen initiiert werden. Eine Kooperation muss strategisch für beide Partner einen Mehrwert mit sich bringen. Entsprechend der Ziele sollte der Grad der Integration festgelegt werden. Das kann durch die Nutzung von fundierten Auswahlprozessen, welche entlang der Unternehmensstrategie gestaltet sind, erreicht werden. „Marketingkooperationen“ sollten so vermieden werden.

**Wünsche für zukünftige Kooperationen:**

Start-ups sollten sich bewusst sein, dass ein etabliertes Unternehmen in bestimmten Strukturen verankert ist. Eine bewusste Auseinandersetzung mit diesen Strukturen führt zu einem höheren Verständnis.

## **Interview 6 IT-Manager Unternehmensberatung**

Mündliches Experteninterview, am 04.05.2019, 27 min, Inhaltliche Zusammenfassung:

### **Kooperationsmotive:**

Das Hauptmotiv für uns ist die Integration von wichtigen Services, welche unsere Kunden nutzen.

### **Qualitätssicherung innerhalb der Kooperation:**

Qualitätsmaßnahmen und Audits werden abhängig vom Partner und Produkt aufgesetzt. Bei der Kooperation mit Partnern, welche bspw. Daten liefern ist in erster Linie zu überprüfen ob entsprechende Daten valide sind. Vor allem, wenn diese für wichtige Funktionen im Fahrzeug genutzt werden. Sofern es bei Kooperationen um Services ohne kritische Schnittstellen geht wird die Sicherung anders aufgesetzt.

### **Differenzen in der Unternehmenskultur:**

Die Differenzen zeigen sich deutlich in den unterschiedlichen Arbeitsweisen. Außerdem müssen sich viele neue Partner müssen im Bezug auf bürokratische Anforderungen wie bspw. Dokumentationen anpassen.

### **Welche Faktoren erschweren die Kooperation:**

Die konkrete Vertragsgestaltung gestaltet sich aufgrund fehlender Expertise der Partner oftmals schwierig und langwierig. Hier werden oft spezifische Verträge ausgearbeitet, wobei auch die Risikoverteilung oftmals schwierig zu verhandeln ist.

### **Kritische Erfolgsfaktoren:**

Es ist immer hilfreich Befürworter aus dem höheren Management zu haben. Auf Kooperationen bezogen, sollten gemeinsame Ziele vereinbart werden und ein regelmäßiger Austausch stattfinden.

### **Wünsche für zukünftige Kooperationen:**

Die Weiterentwicklung der eigenen Organisation hin zu agileren Strukturen. Auf diese Weise kann das bisher nicht sehr ausgeprägte Verständnis vieler Kollegen für Partner aus der „New Economy“ erhöht werden.