



Projektgruppe CHILL 2.0

Zwischenbericht

10. Juli 2019

Version 1.0

Madalin-Ioan Branea	Daniel Ott
Alexander Camu	Mohammadjavad Ranji
Niklas Entrup	Noelle Rousselle
Marius Hubatschek	Tobias Schmitz
Artem Jarovoj	

Prof. Dr. W. Damm	(Abt. Sicherheitskritische Eingebettete Systeme)
Prof. Dr. M. Fränzle	(Abt. Hybride Systeme)

Carl von Ossietzky Universität Oldenburg

OFFIS e. V. - Institut für Informatik

IAV GmbH Ingenieurgesellschaft Auto und Verkehr

Versionshistorie

Tabelle 1.: Versionshistorie

Version	Datum	Autor(en)	Änderungen
0.1	01.05.2019	Tobias Schmitz	Dokument angelegt
0.2	29.04.2019	PG CHILL 2.0	Leitfadeninterviews dokumentiert
0.4	13.05.2019	PG CHILL 2.0	Vision erstellt
0.5	20.05.2019	PG CHILL 2.0	User Stories erstellt
0.6	27.05.2019	PG CHILL 2.0	Erste Version der Projektpläne und Use Cases erstellt
1.0	09.07.2019	PG CHILL 2.0	Zwischenbericht zum 1. Review

Inhaltsverzeichnis

Versionshistorie	iii
1 Einleitung	1
1.1 Motivation	2
1.2 Aufbau des Dokuments	2
2 Leitfadeninterview	5
2.1 Gestaltung des Leitfadens	5
2.2 Ablauf	7
2.3 Herausforderungen	7
2.4 Auswertung	8
3 Related Work	15
3.1 Audi	15
3.2 BMW	16
3.3 Ford	16
3.4 Mercedes	17
3.5 Opel	17
3.6 Porsche	18
3.7 Tesla	19
3.8 Toyota	19
3.9 Volkswagen	20
4 Aufgabenstellung	21
4.1 Motivation	21
4.2 Vision	22
4.3 Lösungsansatz	22
4.3.1 Komponenten	24
4.3.2 Portabilität	26
5 Projektmanagement	27
5.1 Umfangsmanagement	28
5.1.1 Projektumfang	28
5.1.2 Liefergegenstände	28
5.2 Anforderungsmanagement	30

5.2.1	Workflow	30
5.2.2	User Stories	31
5.2.3	Use Cases	32
5.2.4	Anforderungen	33
5.3	Zeitmanagement	40
5.3.1	Vorgehensmodell	40
5.3.2	Meilensteinplanung und -kontrolle	43
5.3.3	Sprintplanung	44
5.4	Qualitätsmanagement	47
5.4.1	Dokumentationsmanagement	47
5.4.2	Verbesserungsmaßnahmen	52
5.5	Ressourcenmanagement	53
5.5.1	Experten	54
5.5.2	Expertenteams	58
5.5.3	Personalplanung	58
5.5.4	Weiterbildung	59
5.5.5	Kontrolle physischer Ressourcen	59
5.6	Kommunikationsmanagement	60
5.6.1	Kommunikationsinhalte	60
5.6.2	Methoden und Technologien	61
5.6.3	Kommunikationsregeln	62
5.6.4	Kommunikation mit Stakeholdern	63
5.7	Risikomanagement	64
5.7.1	Risikoidentifikation	64
5.7.2	Risikoklassifikation	65
5.7.3	Risikoquantifizierung	65
5.7.4	Risikoplanung	67
5.7.5	Strategien zur Risikominderung	68
5.8	Tooldokumentation	69
5.8.1	Projektmanagement Tools	69
5.8.2	Anforderungsmanagement Tools	71
5.8.3	Tools zur Grafikerstellung	71
5.8.4	Tools zur Softwareentwicklung	72
5.8.5	Tools für Dokumente, Präsentationen & Literatur	72
5.8.6	Sonstige Tools	73
6	Anforderungserhebung	75
6.1	Stakeholderanalyse	75
6.2	User Stories Katalog	79
6.2.1	Allgemein	80
6.2.2	Musik	80
6.2.3	Navigation	81
6.2.4	Nutzereingaben	82
6.2.5	Rollos	82

Inhaltsverzeichnis

6.2.6	Sitzprofil	83
6.2.7	Klimatisierung	83
6.2.8	Innenraumbeleuchtung	84
6.2.9	Fenster	85
6.2.10	App	85
6.2.11	Priorisierung	86
6.2.12	Datenverwaltung	87
6.2.13	Szenarien	87
6.2.14	Systementwicklung	88
6.3	Use Case Katalog	88
6.3.1	App Vorbereitung von Fahrten	88
6.3.2	App Profilverwaltung	89
6.3.3	Benutzereingaben	90
6.3.4	Datenverwaltung	91
6.3.5	Innenraumbeleuchtung - Dynamische Einstellungen	92
6.3.6	Innenraumbeleuchtung - Statische Einstellungen	93
6.3.7	Klimatisierung - Dynamische Einstellungen	93
6.3.8	Klimatisierung - Statische Einstellungen	94
6.3.9	Musik - Dynamische Einstellungen	95
6.3.10	Musik - Statische Einstellungen	96
6.3.11	Navigation - Dynamische Einstellungen	97
6.3.12	Navigation - Statische Einstellungen	97
6.3.13	Priorisierung	98
6.3.14	Rollos und Fenster - Dynamische Einstellungen	99
6.3.15	Rollos und Fenster - Statische Einstellungen	99
6.3.16	Sitzprofil	100
6.3.17	Systementwicklung	100
6.3.18	Szenarien	101
6.4	Anforderungskatalog	102
6.4.1	Top-Level Anforderungen	106
6.4.2	System im Fahrzeug	115
6.4.3	App	158
6.4.4	Server	173
7	Fazit	181
8	Ausblick	183
Akronyme		185
Glossar		187
Literaturverzeichnis		189
A Entscheidungskatalog		195

A.1 Team	195
A.2 Termine	195
A.3 Protokoll	196
A.3.1 Inhalt	196
A.4 Dokumentation/Dokumente	196
A.5 Anforderungserhebung	196
A.6 Präsentationen	197
A.7 Urlaub	197
A.8 Vorgehensmodell	197
A.9 Aufwandsabschätzung	197
A.10 Tätigkeitsberichte	197
A.11 Aufgaben	198
A.12 Review Prozess	198
A.13 Tools	198
B Herausforderungskatalog	199
C Meilensteinkatalog	201
D Meilensteinkontrolle	203
E Sprintkatalog	205
F Sprint Retrospektive	209

Abbildungsverzeichnis

2.1	Kilometer pro Jahr der Fahrzeuge	9
2.2	Automobilhersteller & Baujahre der Fahrzeuge	9
2.3	Fahrzeugausstattung	10
5.1	Beispiel eines Use Case Diagramms aus dem Abschlussbericht der PG CHILL 1.0	32
5.2	Schablone für Anforderungen nach Andreas Winter	36
5.3	Vorgehensmodell	43
5.4	Offizielles Logo der PG CHILL 2.0	49
5.5	Beispiel für drei Bilder nebeneinander	49
5.6	Workflow der Tasks in Jira	52
6.1	Use Case App Fahrtvorbereitung	89
6.2	Use Case App Profilverwaltung	90
6.3	Use Case Benutzereingaben	91
6.4	Use Case Datenverwaltung	92
6.5	Use Case Innenraumbeleuchtung - Dynamische Einstellungen	92
6.6	Use Case Innenraumbeleuchtung - Statische Einstellungen	93
6.7	Use Case Klimatisierung - Dynamische Einstellungen	94
6.8	Use Case Klimatisierung - Statische Einstellungen	95
6.9	Use Case Musik - Dynamische Einstellungen	96
6.10	Use Case Musik - Statische Einstellungen	96
6.11	Use Case Navigation - Dynamische Einstellungen	97
6.12	Use Case Navigation - Statische Einstellungen	98
6.13	Use Case Priorisierung	98
6.14	Use Case Rollos Fenster - Dynamische Einstellungen	99
6.15	Use Case Rollos Fenster - Statische Einstellungen	100
6.16	Use Case Sitzprofil	100
6.17	Use Case Systementwicklung	101
6.18	Use Case Szenarien	102

Tabellenverzeichnis

1	Versionshistorie	iii
2.1	Leitfragen	6
2.2	Priorisierung	7
2.3	Demographische Daten	7
2.4	Quintessenzen der Interviews	11
2.5	Kategorien	12
2.6	Priorisierungsergebnisse	13
5.1	Aspekte für nicht-funktionale Anforderungen	34
5.2	Anforderungsvorlage	37
5.3	Beispiel für eine Tabelle mit <i>tabularx</i>	49
5.4	Tabelle mit <i>tabularx</i> und <i>keepXColumns</i>	50
5.5	Tabelle mit <i>tabularx</i> und <i>convertXColumns</i>	50
5.6	Tabelle mit <i>tabularx</i> , <i>multirow</i> & <i>multicolumn</i>	50
5.7	Tabelle mit Labels für die Dokumentation	51
5.8	Expertendefinition	55
5.9	Expertenrollenänderungen	57
5.10	SWOT-Matrix	64
5.11	Risikoklassen	65
5.12	Diskretisierte Skala für Eintrittswahrscheinlichkeiten von Risiken	65
5.13	Skala der Auswirkungen von Risiken	66
5.14	Risikomatrix	66
5.15	Risikobewertung	67
6.1	Stakeholder des Projektes	76
8.1	Zeitplan	183
B.1	Herausforderungskatalog	199
C.1	Meilensteinkatalog	201
D.1	Meilensteinkontrolle	203
E.1	Sprintkalatog	205

Tabellenverzeichnis

F.1 Retrospektive vom 27.05.2019	209
F.2 Retrospektive vom 12.06.2019	210
F.3 Retrospektive vom 01.07.2019	210

1. Einleitung

Die Anzahl an Funktionalitäten und Features in Fahrzeugen nimmt stetig zu. Dadurch werden die Erwartungen der Käufer an ihr nächstes Fahrzeug anspruchsvoller. Der Fokus liegt dabei nicht mehr ausschließlich auf Geschwindigkeit und Leistung. Komfortfunktionen und die damit verbundenen Verbesserungen des Fahrerlebnisses werden immer beliebter. Auch hier bietet der Markt von Ledersitzen bis hin zu gehobenen Komfortsystemen eine Bandbreite an Optionen. Ein wichtiger Aspekt in dieser Hinsicht ist die dahinterliegende Technik.

Die Projektgruppe Configurating & Helping, Independent Learning Limousine (CHILL 2.0) setzt sich mit der Entwicklung eines serienmäßig einsetzbaren Komfortsystems für Fahrzeuge auseinander. Der Fokus dieses Systems liegt auf der automatischen Konfiguration des Fahrzeugs durch ein intelligentes System und die damit verbundene Unterstützung des Nutzers vor und während der Fahrt. Die Projektgruppe lebt dazu einen Industrieprozess und setzt sich mit den auftretenden Herausforderungen und Chancen einer sicherheitskritischen Entwicklung auseinander. Gleichzeitig zeigt die Projektgruppe, wie sich ein solches Komfortsystem in einen realen Versuchsträger integrieren und für den Nutzer anwenden lässt.

Es handelt sich bei diesem Forschungsprojekt um die Weiterentwicklung des Projektes CHILL 1.0 (Cooling & Heating, Independant Learning Limousine). Dies wird als CHILL 2.0 (Configurating & Helping, Independant Learning Limousine) weitergeführt. Das Forschungsprojekt erstreckt sich über einen Zeitraum von zwölf Monaten (April 2019 - April 2020) und wird von Studenten der Carl von Ossietzky Universität durchgeführt. Das Projekt wird vom OFFIS e.V. - Institut für Informatik und der IAV GmbH betreut und unterstützt.

Die Projektgruppe ist in der konkreten Funktionalität und dem Umfang des Komfortsystems weitestgehend frei und bestimmt damit selbst die zugrundeliegende Aufgabenstellung.

Damit eine zugrundeliegende Aufgabenstellung definiert werden kann, hat die Projektgruppe zunächst eine Orientierung in Form einer Vision definiert. Zur Erarbeitung der Vision wurden verschiedene Problematiken von aktuellen Komfortsystemen berücksichtigt und mit bestehenden Komfortsystemen verglichen. Ebenso fließen die Wünsche und Bedürfnisse der Nutzer in die Vision ein, welche mit Hilfe von Leitfadeninterviews identifizieren wurden.

1. Einleitung

1.1. Motivation

Wie bereits erwähnt existiert eine große Anzahl an Komfortsystemen und Assistenzsystemen, die dem Nutzer ein angenehmeres Komforterlebnis schaffen sollen. Beispiele hierfür sind das Anlegen und Arbeiten mit Nutzerprofilen, um Spiegel- und Sitzeinstellungen, Wunschttemperatur und Musikplaylists bei Fahrtantritt eingestellt zu haben. Diese Anzahl an Funktionen kann zur Unübersichtlichkeit und Überforderung des Nutzer führen. Viele dieser Systeme sind so komplex und umfangreich, dass manche Nutzer mit der Einrichtung und Nutzung überfordert sind und dadurch das volle Potenzial dieser Systeme nicht ausschöpfen können [1, 2, 3].

Eine Gefahr dieser komplexen Systeme ist, dass die hohe Anzahl der Bedienelemente, die dem Fahrer zur Verfügung stehen, die Ablenkung im Straßenverkehr erhöht. Speziell Touchscreens bringen ein erhöhtes Sicherheitsrisiko mit sich, da der Fahrer seinen Blick von der Straße abwenden muss[4]. Ein weiteres Problem ist, dass aktuelle Systeme etablierter Automobilhersteller zwar über umfangreiche und durchdachte Assistenz- und Komfortsysteme verfügen, diese sind jedoch an die jeweilige Automarke gebunden und nicht fahrzeugübergreifend einsetzbar.

Ein weiteres Problem liegt bei einer Vielzahl von Fahrzeugen, die nicht die Funktionalität von Nutzerprofilen bieten und von mehreren Personen genutzt werden. Dabei kann es sehr zeitaufwendig sein, vor jeder Fahrt seine persönlichen Einstellungen vorzunehmen. Dieses Problem ist für Nutzer von Car-Sharing oder Firmenflotten umso größer, da sie sich auf immer andere Fahrzeuge mit anderen Bedienungen einstellen müssen.

In dem von der Projektgruppe zu entwickelnden System sollen diese Problematiken beseitigt werden.

1.2. Aufbau des Dokuments

Zur Identifikation der Nutzerwünsche wurde ein Leitfadeninterview erstellt und durchgeführt, welches im Kapitel 2 Leitfadeninterview dokumentiert ist. Zusätzlich wurde eine Marktrecherche betrieben und im Kapitel 3 Related Work festgehalten. Die zentrale Vision des Projektes sowie die daraus abgeleitete Aufgabenstellung werden im Kapitel 4 Aufgabenstellung eingeführt und kurz beschrieben.

Das fünfte Kapitel ist das umfangreichste und dokumentiert die einzelnen gebrauchten Management Ansätze die im Projekt verwendet wurden. Unter anderem werden hier zum Beispiel das Risikomanagement und das Anforderungsmanagement beschrieben. Ein Fazit zur Projektgruppe wird im Kapitel sechs gezogen. Das siebte Kapitel dient dazu einen Ausblick über den weiteren Projektverlauf zu liefern.

1.2. Aufbau des Dokuments

Der zweite Teil des Berichts besteht aus mehreren Anhängen die von A bis L aufgezählt werden. Die Anhänge sind die Ausarbeitung mehrerer schon beschriebenen Kapitel sowie auch im Arbeitsverlauf entstandene Dokumente wie die Sprint Retrospektive oder der Risikokatalog.

2. Leitfadeninterview

Im Rahmen der Projektgruppe (PG) CHILL 2.0 wird ein System entwickelt, welches darauf abzielt, den Komfort für den Nutzer zu erhöhen, in dem die Bedürfnisse und Vorlieben des Nutzers im Mittelpunkt stehen. Daher ist es wichtig ein Verständnis dafür zu schaffen, wo die Wünsche und Anforderungen des Nutzers liegen. Aus diesem Grund sollen die Nutzer in die Entwicklung des Systems miteinbezogen werden.

Damit in der Zieldefinition des Systems die Wünsche der Nutzer berücksichtigt werden können, wird ein Interview durchgeführt. Das Hauptziel des Interviews besteht darin, zu erfahren, wie ein Fahrzeug den Nutzer im Komfortbereich unterstützen kann. Um den Nutzer in seinen Antworten nicht einzuschränken bzw. zu beeinflussen, wird ein Leitfadeninterview als Interviewtyp gewählt.

Leitfadeninterviews bieten die Möglichkeit qualitative Daten von den späteren Nutzern des Systems zu ermitteln [5]. Bei einem Leitfadeninterview werden vor dem Interview Themen bis hin zu konkreten Fragen definiert, die angesprochen werden sollen, der sogenannte Leitfaden. Dabei sollen die Fragen offen gestaltet werden. Das bedeutet dem Befragten werden keine Antwortmöglichkeiten, wie zum Beispiel „Ja“ oder „Nein“, vorgegeben. Dadurch wird der Befragte in seinen Antworten weder beeinflusst noch eingeschränkt. Der Leitfaden dient später dazu, die Antworten vergleichbar zu machen. Jedoch können während des Interviews weitere Fragen hinzugefügt werden. Ebenso ist es möglich, dass der Leitfaden während des Interviews verändert oder angepasst werden muss. Dies kann passieren, wenn neue Aspekte angesprochen werden, die für die Anforderungserhebung wichtig sind. Alle Änderungen oder hinzugefügte Fragen müssen während des Interviews notiert werden [6]. Leitfadeninterviews sollen mit einer Einleitungsfrage beginnen, welche den Befragten in das Thema einführt und dazu verleitet, viel zu erzählen. Im Weiteren werden die Hauptfragen gestellt, welche im Verlauf komplexer werden. Am Schluss des Interviews wird eine Abschlussfrage gestellt und dem Befragten gedankt [6].

2.1. Gestaltung des Leitfadens

Bei der Gestaltung des Leitfadens wird darauf geachtet die Fragen offen zu gestalten und einfache Begriffe zu verwenden. Ebenso werden alltagsnahe Situationen beschrie-

2. Leitfadeninterview

ben, damit sich die Befragten mit diesen identifizieren können. Das Interview ist in zwei Hauptteile mit den gleichen Fragen gegliedert (siehe Tabelle 2.1 Leitfragen).

Im ersten Teil beziehen sich die Fragen auf den Befragten als Fahrer. Im zweiten Teil soll der Befragte sich in die Rolle des Mitfahrers versetzen. Diese Einteilung erlaubt es, Anforderungen unterschiedlicher Rollen im Fahrzeug zu identifizieren. Ebenso dienen die Fragen zur Rolle als Mitfahrer dazu, den Befragten auf Aspekte zu lenken, die nichts mit dem Fahren des Fahrzeugs zu tun haben. Alle Fragen beziehen sich auf das Hauptziel des Interviews: die Ermittlung der Wünsche und Bedürfnisse der Fahrzeughalter. Als Einleitungsfrage (Frage 1) werden die Nutzer gefragt, was für sie Komfort im Auto bedeutet. Ziel dieser Frage ist es, alle möglichen Faktoren im Fahrzeug mit einzubeziehen, um eventuell neue Aspekte zu beleuchten. Im Hauptteil des Interviews beziehen sich die Fragen hauptsächlich auf das Fahrzeug der Befragten. Hierzu gehört die Frage, wie ihr Fahrzeug sie im Komfortbereich unterstützen könnte. Ebenso wird gefragt, was die Befragten an ihrem Fahrzeug verbessern wollen (Frage 3) und was sie an ihrem Fahrzeug stört (Frage 4). Dadurch sollen Probleme der aktuell auf dem Markt befindlichen Fahrzeuge identifiziert werden. Um innovative wünschenswerte Funktionen im Fahrzeug zu ermitteln, wird die Frage „*Was könnten Autos Deiner Meinung nach zukünftig verbessern?*“ (Frage 5) in den Leitfaden eingebaut.

Frage 6 ist eine, für ein Leitfadeninterview untypisch, geschlossene Frage (siehe Tabelle 2.2 Priorisierung). Die Frage dient zur Einschätzung, wie wichtig dem Befragten Komfort im Vergleich zu anderen Faktoren, wie Umweltfreundlichkeit und Vermeidung von Datenerfassung, ist. Diese geschlossene Frage dient dazu eine konkrete Antwort auf die genannten Vergleiche zu erhalten. Dadurch kann der grundlegende Fokus der Vision (siehe Abschnitt 4.2 Vision) festgelegt werden.

Die Abschlussfrage (Frage 7, siehe Tabelle 2.1 Leitfragen) bezieht sich auf ein lernfähiges System für Fahrzeuge, welches zugeschnitten auf den Fahrer/Mitfahrer den Komfort erhöhen soll. Die Befragten sollen mögliche Funktionen für ein solches System nennen. Diese Frage hat das Ziel, direkt auf das zu entwickelnde System bezogene Funktionen und Anforderungen zu ermitteln.

Tabelle 2.1.: Leitfragen

Nr	Leitfragen
1	Was bedeutet für Dich Komfort?
2	Wie könnte Dich Dein Auto im Komfortbereich unterstützen?
3	Was würdest Du an Deinem Auto verbessern?
4	Was stört Dich an Deinem Auto?
5	Was könnten Autos Deiner Meinung nach zukünftig verbessern?
7	Wir entwickeln ein lernfähiges System für Autos, welches zugeschnitten auf den Fahrer/Beifahrer den Komfort erhöhen soll. Welche möglichen Funktionen fallen Dir für ein solches System ein?

Tabelle 2.2.: Priorisierung

Begriff 1	1	2	3	4	5	Begriff 2
Komfort						Umweltfreundlichkeit
Komfort						Vermeidung von Datenerfassung

2.2. Ablauf

Jeder Teilnehmer der PG CHILL 2.0 soll zwei bis drei Interviews durchführen. Dadurch sollen unterschiedliche Nutzergruppen einbezogen werden. Für das Interview gibt es eine Vorlage, an der sich die Interviewer orientieren. Die Vorlage enthält eine genaue Beschreibung des Ablaufs, Abfragen zu demographischen Daten (siehe Tabelle 2.3 Demographische Daten) und den Leitfaden. Durch die Vorlage soll gewährleistet werden, dass jeder Interviewer den Ablauf und den Leitfaden des Interviews kennt und sich an die Richtlinien eines Leitfadeninterviews hält.

Tabelle 2.3.: Demographische Daten

Frage
Alter
Geschlecht
km pro Jahr
Nutzung
Autotyp
Baujahr
Ausstattung

Zu Beginn des Interviews soll den Befragten für ihre Zeit gedankt werden. Außerdem soll das Hauptziel des Interviews erklärt werden, um den Rahmen des Interviews festzulegen. Daraufhin sollen die demographischen Daten erfasst werden. Hierzu gehört das Alter, das Geschlecht, gefahrene Kilometer pro Jahr sowie Daten zum Auto des Befragten. Nachdem alle Daten erfasst wurden, soll das eigentliche Interview geführt werden. Am Ende des Interviews soll den Befragten nochmals gedankt werden und gefragt werden, ob die Befragten Interesse an den Ergebnissen der Studie haben.

2.3. Herausforderungen

Während des Interviews sind Herausforderungen aufgetreten, die im Folgenden erläutert werden. Es konnten wenig innovative bzw. relevante Informationen ermittelt werden. Dies lag wahrscheinlich an drei unterschiedlichen Faktoren. Zum Einen hätten die Fragen mehr auf Innovationen abzielen müssen. Ebenso hatten die meisten

2. Leitfadeninterview

Befragten nicht ausreichend Informationen über Technologien, die schon entwickelt bzw. auf dem Markt sind. Daher sind oftmals Funktionen genannt worden, die bereits existieren. Zum Dritten hat es sich als schwierig für die Befragten herausgestellt, Ideen für Funktionen zu entwickeln.

Eine weitere Herausforderung war, dass die Interviewer unterschiedlich gut darin geübt sind, Interviews zu führen. Aus diesem Grund wurden wenig Nachfragen gestellt bzw. das Interview auf die Befragten angepasst. Dadurch ist der Vorteil des Leitfadeninterviews und mögliche Informationen von den Befragten verloren gegangen. Sollten im weiteren Projektverlauf weitere Interviews durchgeführt werden, wird vorher ein Interviewworkshop gegeben.

2.4. Auswertung

Für das Interview wurden insgesamt 22 Nutzer befragt, davon waren 12 Frauen und 10 Männer. Die weiblichen Befragten waren zwischen 18 und 61 Jahren alt und das Durchschnittsalter lag bei 45,6 Jahren. Die männlichen Befragten waren zwischen 22 und 64 Jahre alt und das Durchschnittsalter der Männer lag bei 47,1 Jahre. Von den 22 Befragten besaßen 14 ein eigenes Fahrzeug, sechs ein gemeinsames Familienfahrzeug und drei einen Firmenfahrzeug. Der Nutzungszweck der Fahrzeuge der Befragten wurde in Stadt und Autobahn aufgeteilt; 15 Fahrzeuge der Befragten werden hauptsächlich in der Stadt benutzt und acht auf der Autobahn. Die gefahrenen Kilometer pro Jahr (siehe Abb. 2.1 Kilometer pro Jahr der Fahrzeuge) zeigen, dass die meisten Befragten im Schnitt zwischen 15.000 bis 20.000 km im Jahr fuhren. Zwei Befragte fuhren mit ihren Fahrzeugen weniger als 5000 km und fünf Befragten fuhren mehr als 20.000 km im Jahr.

Die von den Befragten am häufigsten genannten Fahrzeugmodelle sind von den Herstellern Volkswagen (VW) (5), Audi (3) und BMW (3) (siehe Abb. 2.2a Automobilhersteller der Fahrzeuge). Die meisten Befragten besitzen ein Fahrzeug aus dem Baujahr 2016 (5), gefolgt von Fahrzeugen aus dem Jahr 2015 (4). Die Baujahrspanne reicht von 1998 bis 2018 (siehe Abb. 2.2b Baujahre der Fahrzeuge). Die Befragten gaben die Fahrzeugausstattung ihrer Fahrzeuge an, diese reichten von Standard- bis Vollausstattung. Sieben Fahrzeuge der Befragten haben laut eigener Aussage eine Standardausstattung, sieben eine gehobene Ausstattung und fünf eine Vollausstattung (siehe Abb. 2.3 Fahrzeugausstattung).

2.4. Auswertung

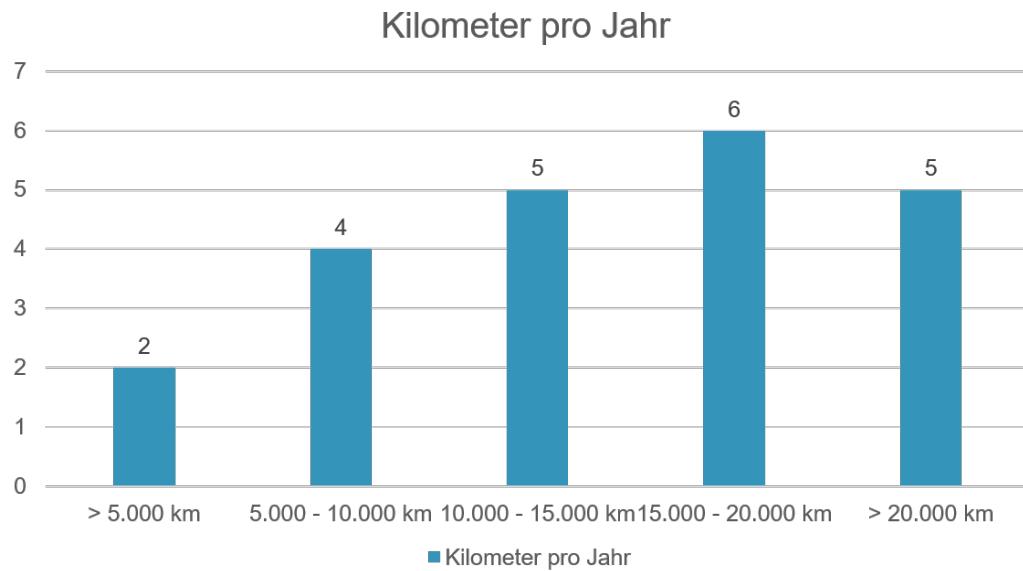


Abbildung 2.1.: Kilometer pro Jahr der Fahrzeuge

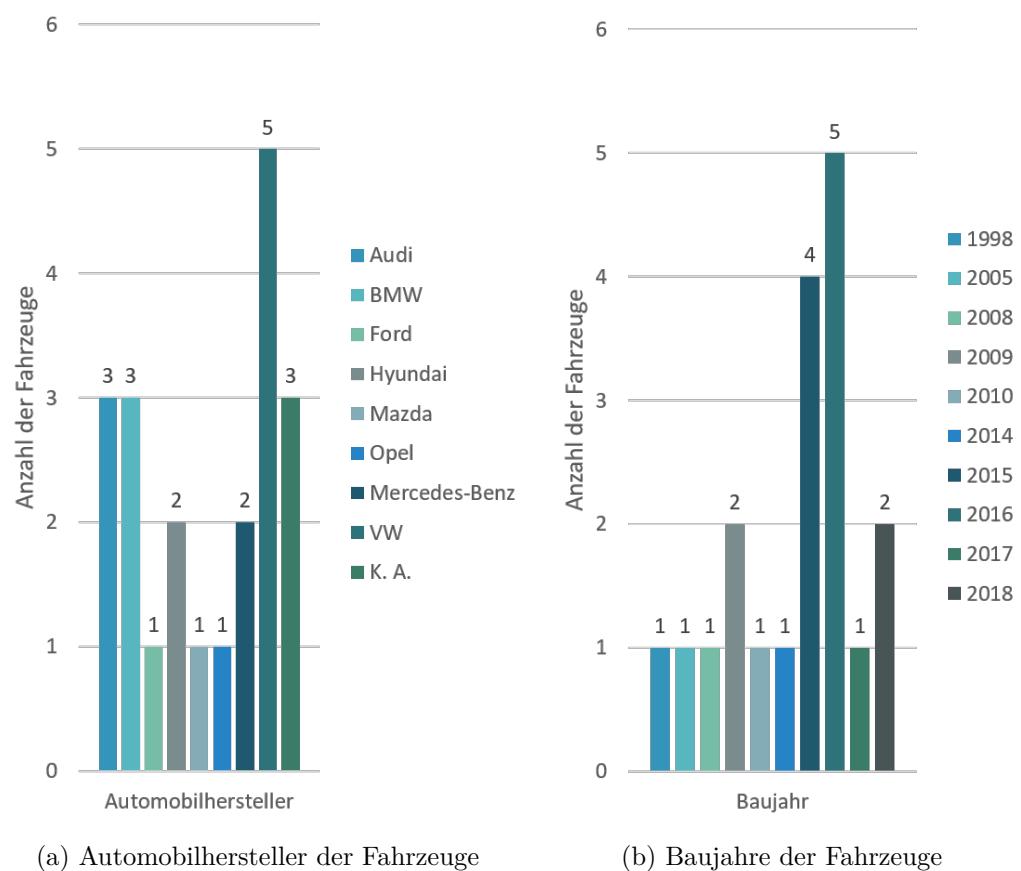


Abbildung 2.2.: Automobilhersteller & Baujahre der Fahrzeuge

2. Leitfadeninterview

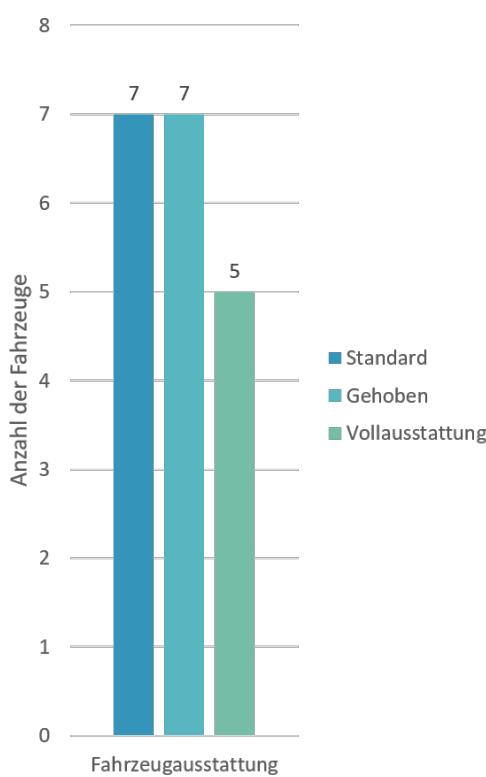


Abbildung 2.3.: Fahrzeugausstattung

Für die Auswertung des Interviews wurden zunächst alle wichtigen Quintessenzen identifiziert und in einer Tabelle gesammelt. Zu jedem Punkt wurde vermerkt, wie oft dieser genannt wurde. Ebenso wurden eventuelle Anmerkungen notiert (siehe Tabelle 2.4 Quintessenzen der Interviews). Im Weiteren wurden auf Basis der relevanten Aspekte Kategorien gebildet, um zu identifizieren, wo die allgemeinen Interessen der Nutzer liegen (Tabelle 2.5 Kategorien). Aus der Tabelle zeigt sich deutlich, dass Aspekte zu der Kategorie Entertainment besonders häufig genannt wurden. Daraus lässt sich ableiten, dass neben dem eigentlichen Fahren auch die Unterhaltung im Fahrzeug eine wesentliche Rolle spielt und vom Entertainmentsystem stark beeinflusst wird.

Eine weitere große Rolle spielen Entertainment und Profileinstellungen. In der Kategorie Profileinstellungen wurde besonders häufig die Funktion Speicherung des Sitzprofils genannt. Fast genauso

häufig wurde gewünscht, dass sich das Fahrzeug vorklimatisieren lässt. Aus diesen beiden Punkten lässt sich ableiten, dass es für ein hohes Komfortgefühl wichtig ist, dass im Fahrzeug vor Fahrtantritt schon bestimmte Sachen eingestellt sind und man direkt losfahren kann.

Ebenso nennenswert ist der Punkt „Sprachsteuerung im Auto“. Ein Beispiel hierfür: „Ich muss gar nicht erst tippen, sondern kann per Sprachsteuerung z.B. den Radiosender wechseln“. Durch eine Sprachsteuerung ist es für den Nutzer einfacher Einstellungen während der Fahrt zu ändern, da eine Sprachsteuerung weniger Aufmerksamkeit, bezogen auf das Sichtfeld, beansprucht als Touch-Eingaben oder Tastenbedienungen. Eine weitere oft genannte Funktion ist die Anbindung an das Smartphone. Smartphones nehmen im Alltag eine große Rolle ein und könnten auch in einem Komfortsystem eines Fahrzeugs eingebunden werden, um bestimmte Apps im Fahrzeug zu verwenden oder um Einstellungen im Fahrzeug vorzunehmen. Zudem können individualisierte Nutzerprofile über das Smartphone angelegt werden.

Neben der oben beschriebenen Auswertung, wurde Frage 6 (Wichtigkeit von Komfort im Vergleich zu anderen Faktoren) gesondert ausgewertet. In der Tabelle 2.6

2.4. Auswertung

Priorisierungsergebnisse sind die Ergebnisse von Komfort (1) im Vergleich zu Umweltfreundlichkeit (5) sowie die Ergebnisse von Komfort (1) im Vergleich zu Vermeidung von Datenerfassung (5) zusammengefasst. Umweltfreundlichkeit ist den Nutzern im Durchschnitt etwas wichtiger, als der Komfort. Jedoch ist ihnen Komfort im Durchschnitt wichtiger, als die Vermeidung von Datenerfassung. Dies lässt sich damit begründen, dass viele gewünschte Funktionen die Erfassung von Daten voraussetzen.

Alle Ergebnisse des Interviews werden in die Visionsgestaltung des Systems einbezogen und berücksichtigt, damit ein System entwickelt wird, welches von den Nutzern gewünscht ist.

Tabelle 2.4.: Quintessenzen der Interviews

Oberbegriff	Zähler	Anmerkungen
Sitzprofil	9	inkl. Seitenspiegelprofil (3)
Sprachsteuerung im Auto	8	
Vorklimatisierung	8	
Bedienungsfreundlichkeit	7	
Parkplatzsuche	6	
Temperaturprofil	6	
Verkehrsinformationen	5	
Müdigkeitserkennung	5	
Funktionstüchtigkeit vor Fahrtantritt prüfen	5	z.B. ob genug Tank vorhanden ist
Entertainmentsystem für Mitfahrer	4	
Einbindung von externen Services	4	z.B. Spotify
Gespeicherte Profil Einstellungen	3	(Allgemein)
Anbindung an das Smartphone	3	Als Fernbedienung (1)
Internetverbindung im Auto	3	
Musik und Radio Profil	3	
Film gucken	3	
Intelligente Navi Einstellung	3	Voreinstellungen
Auswertung von Fahrdaten	2	Fahrdaten über eine App zugänglich machen
Buch lesen	2	
Lautstärkeprofil	2	
Wetterinformationen	1	
Radarwarnsystem	1	
Automatische Reservierungsvorschläge	1	z.B. für Restaurants, Hotels, etc.
Radio über Touch steuerbar	1	
Standorterkennung des Nutzers	1	z.B. um das Auto automatisch zu öffnen

2. Leitfadeninterview

Inkognito Einstellung	1	Fahrdaten sollen nicht aufgezeichnet werden
Fußgängerwarnsystem	1	
Intelligente Innenraumbeleuchtung	1	
Vorkonditionierung per App	1	Klimaanlage
Abfahrtszeiten lernen	1	
Tutorials	1	
Gesichtserkennung zum Aufschließen	1	

Tabelle 2.5.: Kategorien

Kategorie	Quintessenzen	Vorkommen
Entertainment		41
Anbindung an das Smartphones	8	
Sprachsteuerung im Auto	8	
Bedienungsfreundlichkeit	7	
Einbindung von externen Services	4	
Eigenes Entertainmentsystem für Mitfahrer	4	
Internetverbindung im Auto	3	
Film gucken	3	
Buch lesen	2	
Radio über Touch steuerbar	1	
Intelligente Innenraumbeleuchtung	1	
Infotainment		23
Bedienungsfreundlichkeit	7	
Funktionstüchtigkeit vor Fahrtantritt prüfen	5	
Verkehrsinformationen	5	
Internetverbindung im Auto	3	
Wetterinformationen	1	
Radarwarnsystem	1	
Tutorials	1	
Profileinstellungen		23
Sitzprofil	9	
Temperaturprofil	6	
Allgemein	3	
Musik und Radio Profil	3	
Lautstärkeprofil	2	
Navigation		18
Parkplatzsuche	6	
Verkehrsinformationen	5	
Intelligente Navi Einstellung	3	

2.4. Auswertung

Auswertung von Fahrdaten	2
Automatische Reservierungsvorschläge	1
Inkognito Einstellung	1
Vorkonditionierung	11
Vorklimatisierung	8
Vorkonditionierung per <i>Ophelia-App</i>	1
Abfahrtszeiten lernen	1
Standorterkennung des Nutzers	1
Weitere Quintessenzen	7
Müdigkeitserkennung	5
Fußgängerwarnsystem	1
Gesichtserkennung zum Aufschließen	1

Tabelle 2.6.: Priorisierungsergebnisse

Begriff 1	1	2	3	4	5	Begriff 2	Mittelwert
Komfort	7	1	6	7	1	Umweltfreundlichkeit	2,73
Komfort	9	6	4	2	1	Vermeidung von Datenerfassung	2,09

3. Related Work

Die Projektgruppe hat basierend auf dem Rahmen der Projektgruppe PG CHILL 2.0 Recherchen zu unterschiedlichen Systemen verschiedener Automobilhersteller durchgeführt, um die Aufgabenstellung der Projektgruppe gegenüber aktuellen Systemen am Markt abgrenzen zu können. Dabei wurden im Besonderen Systeme ermittelt, die den Komfort von Nutzern im Fahrzeug erhöhen und den Nutzer im Bezug auf den Komfort im Fahrzeug vor oder während der Fahrt unterstützen. Dazu wurden die Automobilhersteller Audi, BMW, Ford, Mercedes, Opel, Porsche, Tesla, Toyota und Volkswagen näher untersucht.

3.1. Audi

Audi bietet das *MMI touch response-System* mit intuitiver Bedienung und unterschiedlichen Personalisierungsmöglichkeiten. Mit diesem ist es beispielsweise möglich, die Funktionen auf dem Infotainment-Display per *drag and drop* anzupassen. Ebenfalls ist es möglich, für maximal sieben Personen, ca. 400 Parameter zu speichern [7].

Die MMI Naviagation lernt auf Basis der gefahrenen Strecken, was in intelligente Suchvorschläge resultiert. Hierbei wird die aktuelle Verkehrslage miteinbezogen. Zudem wird die Funktion *On-Street Parking* bereitgestellt, welche die Parkplatzsuche erleichtert [7].

Stimmenerkennung, Sprachsteuerung sowie ein HUD bieten bei Audi gute Möglichkeiten zur kontaktlosen Interaktion zwischen Auto und Mensch. Für das Smartphone gibt es eine direkte USB-Schnittstelle sowie eine Bluetooth Verbindung, durch welche Anrufe direkt über das Auto getätigt werden können. Bei Touch-Eingaben gibt das System dem Nutzer haptisches Feedback [8] [9].

Das Ver- und Entriegeln des Autos sowie der Motorstart lassen sich, von maximal fünf Personen, über ein Android-Smartphone steuern. Der Datenaustausch erfolgt hierbei über *Near Field Communication* [7].

Auf der Technik-Messe CES in Las Vegas wurde 2015 ein Forschungsfahrzeug vorgestellt. In diesem Fahrzeug hat der Beifahrer ein integriertes Display, welches in die Oberfläche des Armaturenbrett eingearbeitet ist und sich über Gesteuerung bedienen lässt. Der Beifahrer kann dem Fahrer Vorschläge über dieses Display

3. Related Work

senden; zum Beispiel ist es möglich über eine Wischbewegung eine vorkonfigurierte Route vorzuschlagen. Der Fahrer kann diese dann akzeptieren oder ablehnen [10].

Über die *Connect App* ist es möglich die Termine des Kalenders übertragen, sodass die Ziele in der Navigation übernommen werden können. Zudem kann man die lokal gespeicherte Musik auf dem Smartphone abspielen [7].

3.2. BMW

BMW bietet seinen Kunde die Möglichkeit mit Hilfe von Gesten das Teile des Fahrzeugs zu steuern. Mit dem Heads-Up Display von BMW kann der Fahrer ein Vielzahl von Informationen erlangen ohne den Blick von der Fahrbahn abzuwenden. Die Nachtsicht erkennt automatisch bei Dunkelheit Personen auf der Fahrbahn. Außerdem haben einige Fahrzeugmodelle automatisch ausfahrbare Sonnenrollen und einen Garagentoröffner im Spiegel integriert.

BMW bietet außerdem einen Komfortzugang mit dem das Auto aufgeschlossen und gestartet werden kann ohne die Fernbedienung zu benutzen. Ein weiteres Feature ist die Massagefunktion, die auf verschiedenen Stufen regulierbar ist. In dem neuen BMW 7er ist es möglich Eingaben durch zeichnen von Buchstaben auf dem Touchpad zu erledigen, statt auf einen Buchstaben auf der virtuellen Tastatur zu drücken.[11].

3.3. Ford

Ford SYNC ist ein Kommunikations- und Entertainment System, welches es dem Nutzer erlaubt, per Sprachsteuerung Anrufe anzunehmen und zu tätigen, das Autoradio sowie die Klimaanlage zu bedienen und über eine Bluetooth-Verbindung bestimmte Mobilapplikationen, u.a. n-tv, wetter.de und Glympse [12], zu nutzen [13, 14].

Das System enthält die folgenden Funktionalitäten: Abrufen von Wetter- und Verkehrsdaten, Assistenz für Notrufe, Statusüberwachung der Funktionalität im Fahrzeug, Navigation, Ambientebeleuchtung, Remote-Verbindung auf PCs, Verbindung zu Apple CarPlay [15] und Android Auto [16, 13, 17, 18].

Mittels AppLink [19] erlaubt das System dem Nutzer, kompatible Applikation auf seinem Smartphone durch die Sprachsteuerung im Fahrzeug zu bedienen. Entwickler können Gebrauch vom Software Development Kit von Ford machen, um Kompatibilität mit ihren Applikationen zu gewährleisten [12].

Als weiteres System wird FordPass Connect angeboten [20]. Damit wird ein WLAN-Hotspot im Fahrzeug über LTE von Vodafone angeboten. Mittels der zugehörigen

mobilien Applikation ist es möglich, das Fahrzeug auch über größere Entfernungen, z.B. für Familienmitglieder, auf- und abzuschließen. Durch die Ortungsfunktion der App kann man den Standpunkt des Fahrzeugs überwachen. Somit kann ein Nutzer sein Fahrzeug auch auf unübersichtlichen Parkplätzen schnell wiederfinden [20].

3.4. Mercedes

Mercedes Benz bietet für ihre Fahrzeuge die Möglichkeit der Smartphone-Integration [21]. Damit kann ein Smartphone mit dem Media-Display des Fahrzeugs gekoppelt werden. Die Dienste Apple CarPlay [15] und Android Auto [16] werden unterstützt. Das Media Display wird über Sprachsteuerung gesteuert, zudem kann auf Funktionen, wie Navigation, Telefonie und Radio zugegriffen werden. Die Darstellung des Smartphone-Bildschirms ist für das Media-Display optimiert. Um Smartphone-Integration benutzen zu können ist ein Smartphone erforderlich, das mit Apple CarPlay oder Android Auto kompatibel ist.

Ebenfalls sind Mercedes-Benz Apps verfügbar [22]. Diese bieten u.a. die folgenden Funktionalitäten:

- Betriebsanleitungen ansehen (Mercedes-Benz Guides)
- Technische Fahrzeugdetails prüfen, wie z.B. den Reifendruck oder den Tankstand (Mercedes me)
- Nachrichten über Neuigkeiten rund um Mercedes Benz erhalten (Mercedes-Benz Magazin)
- Das Speichern von gefahrenen Strecken sowie schnelles Finden vom geparkten Fahrzeug (Mercedes me Adapter)

3.5. Opel

Das Infotainment System von Opel, IntelliLink [23], bietet die Funktion, bis zu fünf Nutzerprofile anzulegen. In den Profilen ist es möglich, Playlists, favorisierte Kontakte und Zielorte sowie Klima- und Toneinstellungen festzulegen [24]. Dadurch ist es möglich für unterschiedliche Nutzer des gleichen Autos Einstellungen zu speichern.

Das Infotainment System IntelliLink ist über Touch sowie über eine Sprachfunktion steuerbar [23]. Die Sprachsteuerung wird über einen Knopf am Lenkrad gestartet und kann über diesen auch wieder abgebrochen werden. Dadurch ist es möglich, während der Fahrt Musik- und Navigationseinstellungen zu ändern oder Telefonate zu starten. Unterstützend zu dem Infotainmentsystem gibt es zusätzlich ein Head-Up Display mit einer Navigationsanzeige [24].

3. Related Work

Ebenfalls bietet IntelliLink die Möglichkeit sein Smartphone anzukoppeln. Hierzu wird entweder die App Android Auto [16] oder Apple CarPlay [15] benötigt. Über die Anbindung sind ausgewählte Services des Smartphones ansprechbar. Hierzu gehören unter anderem Musikdienste, Nachrichtendienste und Siri oder die Google Sprachsteuerung.

Für ein hohes Komfortgefühl während der Fahrt bietet Opel Ambientebeleuchtung der Mittelkonsole und der vorderen Türen [25]. Zudem gibt es vorne und hinten Innenraumbeleuchtung inklusive Leseleuchten [25]. Zugang zum Auto erlangt man über das schlüssellose Schließsystem „Keyless Open“, dadurch bleibt es dem Nutzer erspart den Schlüssel zu suchen.

3.6. Porsche

Porsche bietet mit *My Porsche* eine Plattform, auf der viele Dienste und Angebote bereitgestellt werden [26]. Mittels einer Porsche ID können Accounts angelegt werden, welche in Fahrzeugen oder in Applikationen, wie der *Porsche Connect App* [27], genutzt werden können. Diese ID gewährleistet die Synchronisation zwischen allen Endgeräten und Fahrzeugen. *My Porsche* bietet dabei u.a. die Möglichkeit, Porsche Connect Dienste zu nutzen, die den Fahrkomfort erhöhen sollen [28].

Porsche Connect und die dazugehörige *Porsche Connect App* ermöglichen dabei schon vor Fahrtantritt Routen und Ziele zu erstellen, die dann an Fahrzeuge gesendet werden können. Weiterhin kann der Standort des Fahrzeugs ermittelt und eine Navigation zum Fahrzeug hin gestartet werden. Zudem können fahrzeugbezogene Daten, wie Laufleistung, Durchschnittsverbrauch und Restreichweite sowie bei Hybridmodellen der Ladezustand und die Restladezeit der Batterie, abgefragt werden. Voreinstellungen, wie das Starten eines Ladevorgangs oder die Vorklimatisierung des Fahrzeugs, können ebenfalls initiiert werden. Im persönlichen Porsche ID Profil können externe Dienste wie z.B. Musikstreamingdienste verwaltet und zugehörige Login-Daten hinterlegt werden. Außerdem kann mittels Apple Car Play [28] auf iPhones zugegriffen und so Apple-Dienste wie z.B. Siri genutzt werden.

Neben Standardfunktionen des *Porsche Connect* Systems können weitere kostenpflichtig dazugebucht werden. Das Navigation- und Infotainmentpaket [29] erweitert dabei das Offline-System Porsche Communication Management u.a. um Online-Informationen zu Strecken, Echtzeit-Verkehrssituationen und Satellitenbildern. Außerdem werden häufig gefahrene Routen erkannt, gespeichert und gegebenenfalls alternative, kürzere Routen vorgeschlagen. Darüber hinaus können Wetterdaten abgefragt, Nachrichten oder Social Media Feed vorgelesen oder Tankstellen in der Nähe oder Parkplätze gesucht werden. Das zugehörige Online-Spracherkennungssystem Voice Pilot ermöglicht dabei u.a. die Steuerung der Navigation (Eingabe von Zielen ohne Adressdetails), der Musik, der Klimatisierung und der Sitzheizung.

3.7. Tesla

Das Unternehmen Tesla hat eine Vielzahl von technischen Komponenten entwickelt, die dem Fahrer ein angenehmeres Fahrerlebnis erlauben sollen.

Tesla beinhaltet für jeden Fahrer des Autos ein separates Profil, welches durch den Schlüssel erkannt wird. Neben der Möglichkeit, die Türen des Autos über sein eigenes Smartphone zu öffnen, gibt es die Möglichkeit, eine Kindersicherung zu verwenden. Mithilfe dieser kann der Fahrzeughalter festlegen, welcher Nutzer des Autos welche Funktionalitäten verwenden kann und welche nicht [30].

Tesla bietet des Weiteren die Möglichkeit die Batterie vorzuheizen, um die Effizienz bei kaltem Wetter zu verbessern [31]. Die Fahrzeuge von Tesla sind zur besseren Kommunikation untereinander vernetzt und in der Lage Over-The-Air-Updates zu erhalten. Das Fahrzeug kann vom Käufer im Voraus online individuell design werden [32].

Weitere Funktionalitäten sind manuelle Sitz- und Lenkeinstellungen sowie Satellitenbildkarten mit Live-Verkehrsvisualisierung und zusätzlicher Navigation. Tesla bietet dem Fahrer eine umfassende Menge an Funktionalitäten, die individuell genutzt werden können. Streamingdienste und Browser sind ebenfalls integriert. Bei Fehrtenden steht dem Fahrer die Möglichkeit zur Verfügung das Garagentor automatisch öffnen zu lassen. Nutzerdefinierte Fahrerprofile stellen ein weiteres Merkmal der Funktionsvielfalt von Tesla dar [33].

3.8. Toyota

Toyota bietet durch Infotainment-Technologien verschiedene Komfortsystem-Services, die die Nutzerkonnektivität erhöhen. Eine dieser Infotainment-Technologien ist das Entune 3.0-System, das auch über eine verfügbare App Suite verfügt, mit der jeder Benutzer im Fahrzeug Verkehrsanwendungen in Echtzeit nutzen kann. Diese Infotainment-Suite enthält auch das Apple CarPlay-System [15], mit dem Benutzer ihr kompatibles iPhone mit ihrem Fahrzeug verwenden und auf die Anwendungen zugreifen können. Außerdem enthält es eine Amazon Alexa-Kompatibilität, mit der Benutzer Musik abspielen, Nachrichten lesen und Witze erzählen lassen können. Hier wird speziell eine Sprachsteuerung benutzt [34].

Scout GPS Link liefert eine dynamische Navigation und einen Zielassistenten. Mit diesem ist es möglich, GPS-Daten von einer anderen Person zu erhalten und automatisch vom Navigationssystem als Route einzustellen zu lassen [35].

3. Related Work

3.9. Volkswagen

Aus dem VW Technik-Lexikon: [36, 37, 38, 39, 40]

VW bietet mit dem *Active Info Display*, dem *Head-up-Display* und der *Multifunktionsanzeige* verschiedene Möglichkeiten zum Anzeigen von Informationen im Fahrzeug für den Fahrer. Die Displays lassen sich zum Teil an die Bedürfnisse der Nutzer anpassen. Neben der Bedienung des Fahrzeugs über einen berührungsempfindlichen Bildschirm oder verschiedene Tasten, biete VW die Möglichkeit Teile des Fahrzeugs mit Hilfe von Gesten oder der gesprochenen Sprache zu steuern. Zusätzlich zu den genannten Aus- und Eingabegeräten im Fahrzeug, bietet VW ein Entertainment System für die hinteren Sitze und eine Applikation für mobile Endgeräte, mit der sich im Fahrzeug zum Beispiel das Musikprogramm ändern oder auch Navigationsdaten übertragen lassen.

Verschiedene Applikationen für die Fahrzeuge werden von VW angeboten. Beispielsweise *Car-Net App-Connect*, *Car-Net Cam-Connect*, *Car-Net Guide & Inform*, *Car-Net Security & Service*. Die verschiedenen Applikationen bieten verschiedene Möglichkeiten zum Zugriff auf unterschiedliche Informationen, die das Fahrzeug selbst, ein verbundenes mobiles Endgerät oder externe Services betreffen.

Fahrzeuge von VW bieten verschiedene Möglichkeiten der Beleuchtung im Fahrzeug und außerhalb des Fahrzeugs. Dazu zählen illuminierte Einstiegs- sowie Dekorleisten, Leselampen vorn und hinten und die Beleuchtung des Fußraums (*Ambientebeleuchtung*). Des Weiteren leuchtet für eine bestimmte Zeit das Abblendlicht, die Umfeldleuchten in den Außenspiegelgehäusen, das Schlusslicht der Heckleuchten und die Kennzeichenbeleuchtung, wenn das Fahrzeug aufgeschlossen (*Leaving Home-Funktion*) wird oder wenn das Fahrzeug abgestellt, verlassen und abgeschlossen wird (*Coming Home-Funktion*).

Verschiedene Sensoren, wie zum Beispiel der *Feuchtesensor*, der *Luftgütesensor* oder der *Regensensor*, sorgen in Kombination mit der *Klimaanlage* oder auch *Climatronic* und der *Standheizung und -lüftung* für eine angenehme Luftfeuchtigkeit und Temperatur im Fahrzeug auch schon vor Fahrtantritt.

Mit der *Memory-Funktion* lassen sich individuelle Sitzeinstellungen von verschiedenen Personen speichern. Zusätzlich lassen sich Fahrersitz- und Außenspiegeleinstellungen einem Fahrzeugschlüssel zuordnen.

4. Aufgabenstellung

Im Rahmen der Lehrveranstaltung Projektgruppe unter dem Titel *Configurating & Helping, Independent Learning Limousine* soll ein Komfortsystem für ein Serienfahrzeug entwickelt werden.

Um mögliche Funktionen für das System zu ermitteln, wurden Interviews nach einem vorab erarbeiteten Leitfaden durchgeführt (siehe Kapitel 2 Leitfadeninterview). Außerdem wurden Recherchen zu Systemen von etablierten Automobilherstellern durchgeführt, um zu erfahren, wie die aktuelle Situation des Marktes aussieht und welche Funktionen, die einen Bezug zu der Projektgruppe haben, von anderen Herstellern entwickelt wurden (siehe Kapitel 3 Related Work).

Zunächst werden die Probleme der auf dem Markt befindlichen Komfortsysteme identifiziert und daraus die Motivation abgeleitet (siehe Abschnitt 4.1 Motivation). Auf Basis der Informationen aus dem Leitfadeninterview (siehe Kapitel 2 Leitfadeninterview), der Motivation und der persönlichen Vorlieben der Gruppenmitglieder ist die, im Abschnitt 4.2 Vision beschriebene, Vision entstanden. Auf die zugrunde liegende Aufgabenstellung, welche die Vision konkretisiert, wird im Abschnitt 4.3 Lösungsansatz näher eingegangen.

4.1. Motivation

Viele Automobilhersteller haben eine große Anzahl von Assistenz- und Komfortsystemen entwickelt, die auf die Steigerung des Nutzerkomforts ausgerichtet sind. Beispiele hierfür sind das Anlegen und Arbeiten mit Nutzerprofilen, um Spiegel- und Sitzeinstellungen, Wunschtemperatur und Musikplaylists bei Fahrtantritt eingestellt zu haben sowie diverse Navigations- und Assistenzsysteme.

Auf künstliche Intelligenz (KI) basierende Systeme sind in der Lage Nutzerpräferenzen zu lernen und auf Basis gesammelter Informationen eigenständig Entscheidungen zu treffen. Dies bietet Nutzern den Komfort diverse Funktionen nutzen zu können, ohne diese aktiv ansteuern zu müssen. Im Folgenden werden die Probleme der aktuellen Komfortsysteme beschrieben:

Erstes Problem Viele dieser Systeme sind so komplex und umfangreich, dass manche Nutzer mit der Einrichtung und Nutzung überfordert sind und dadurch das volle Potenzial dieser Systeme nicht ausschöpfen können [1, 2, 3].

4. Aufgabenstellung

Zweites Problem Zudem kann die hohe Anzahl der Bedienelemente und Systeme, die dem Fahrer während der Fahrt zur Verfügung stehen, auch zu einem Sicherheitsrisiko werden. Vor allem bei Touchscreens ist die Gefahr groß, dass der Fahrer während der Bedienung abgelenkt wird, da der Fahrer auf den Bildschirm sehen muss, um ein Element auf dem Bildschirm auszuwählen [4].

Drittes Problem Die Systeme der etablierten Automobilhersteller verfügen zwar über umfangreiche und durchdachte Komfortsysteme, diese sind jedoch an die jeweilige Automarke gebunden und sind meistens nicht lernfähig (siehe Kapitel 3 Related Work).

Viertes Problem Wird ein Fahrzeug von mehreren Nutzern gefahren, kann es sehr zeitaufwendig sein, bis alle Einstellungen für die jeweiligen Personen vorgenommen sind. Dieses Problem ist für Nutzer von Car-Sharing oder Firmenflotten umso größer, da sie sich auf immer andere Fahrzeuge mit anderen Bedienungen einstellen müssen.

4.2. Vision

Die Vision wird als Richtungsangabe für die weitere Arbeit im Projekt verwendet und gibt der Projektgruppe eine greifbare Idee und ein grobes Ziel vor. Dabei definiert die Vision die Kerneigenschaften des Systems. Dieses System wird im folgenden *Ophelia* (die Helfende) genannt und wie folgt beschrieben:

Das System *Ophelia* ist ein Metasystem zur Einstellung verschiedener Komfortsysteme im Fahrzeug, welches die Interaktion mit diesen vereinfacht und somit dem Nutzer Zeit und Mühe erspart. *Ophelia* verarbeitet und lernt Nutzereingaben und konfiguriert darauf basierend Komforteinstellungen vor oder während der Fahrt an. Ebenso unterbreitet *Ophelia* dem Nutzer u.a. Vorschläge für das Navigationssystem. *Ophelia* bietet eine intuitive Nutzerschnittstelle für alle *Ophelia* unterstützenden Fahrzeugsysteme u.a. durch eine Sprach- oder Gesteuerung. Zukünftig soll es so möglich sein in jedes beliebige „*Ophelia*-Fahrzeug“ einzusteigen und ein, den persönlichen Bedürfnissen entsprechend, vorkonfiguriertes Fahrzeug vorzufinden.

4.3. Lösungsansatz

Ziel des Komfortsystems ist es, die im Abschnitt 4.1 Motivation herausgestellten Probleme, wie die Überforderung der Nutzer, das Gebundensein an eine Automarke

oder die beeinträchtigte Sicherheit durch Ablenkungen, zu adressieren und zu lösen.

Ophelia soll den Komfort des Nutzers erhöhen, indem sie Komforteinstellungen vor und während der Fahrt konfiguriert. Hierzu müssen zwei Arten von Einstellungen unterschieden werden:

Statische Einstellungen Statische Einstellungen umfassen alle Einstellungen, die von *Ophelia* ohne die Verwendung einer KI vor der Fahrt durchgeführt werden. Diese Einstellungen können über eine Schnittstelle zu einem Nutzerprofil gespeichert werden. Statische Einstellungen sind:

- Sitzposition
- Position der Innen- und Außenspiegel
- Lenkradposition

Dynamische Einstellungen Dynamische Einstellungen umfassen alle Einstellungen, die von *Ophelia* mit Verwendung einer KI vor der Fahrt oder während der Fahrt durchgeführt werden. *Ophelia* lernt mit der Zeit die Präferenzen des Nutzers mit Hilfe seiner Eingaben in Abhängigkeit der aktuellen Situation. Dynamische Einstellungen sind:

- Licht
- Musik
- Fenster
- Rollo
- Temperatur
- Navigationsziele

Szenarien sind eine Sammlung von dynamischen Einstellungen, die für eine bestimmte Situation gespeichert werden können. Szenarien können im Nutzerprofil angelegt und verwaltet werden.

4. Aufgabenstellung

4.3.1. Komponenten

Ophelia besteht aus den folgenden Komponenten, welche im Weiteren beschrieben werden:

- Nutzerprofil
- *Ophelia-App*
- *Ophelia-Modul*
- *Ophelia-Server*

Nutzerprofil Der Nutzer soll die Möglichkeit haben *Ophelia* in unterschiedlichen Fahrzeugen zu verwenden. Aus diesem Grund ist es notwendig, dass die Daten zum Nutzer in einem *Nutzerprofil* hinterlegt werden können. In diesem Nutzerprofil können Informationen zu statischen Einstellungen hinterlegt werden. Ebenso ist es möglich Szenarien im Nutzerprofil zu speichern. Der Nutzer soll die Möglichkeit haben die statischen Einstellungen in Abhängigkeit zur Rolle im Auto (Fahrer/Mitfahrer) zu definieren.

Ophelia-App Zur Verwaltung des Nutzerprofils und für weitere Nutzereingaben wird eine App benötigt. Über die *Ophelia-App* kann der Nutzer sein Nutzerprofil erstellen und statische Einstellungen sowie Szenarien erstellen und verwalten. Ebenso kann der Nutzer ein Tagesprogramm angeben.

Zur Erstellung des Nutzerprofils muss sich der Nutzer registrieren. Die Registrierung erfolgt über eine Internet-Verbindung zum *Server*. Hierbei wird ein Nutzerprofil in der *Datenbank* gespeichert, welche auf dem Server liegt. Die App hat des Weiteren eine Verbindung zum *Ophelia-Modul*, welches sich im Fahrzeug befindet. Durch diese Verbindung ist es zum Beispiel möglich, geplante Fahrten an das Ophelia-Modul zu übertragen. Kann keine Verbindung hergestellt werden, so ist die App nur eingeschränkt nutzbar und die vorgenommenen Einstellungen werden bis zur nächsten Verbindung über das Internet gespeichert. Diese Einstellungen werden bei der nächsten Verbindung mit dem Server synchronisiert.

Über die *Ophelia-App* erhält der Nutzer ebenfalls Anfragen der *KI*, welche akzeptiert oder abgelehnt werden können. Ein Beispiel hierfür ist, ob eine im Vorfeld gelernte Strecke gefahren wird und ob das Fahrzeug hierfür vorkonfiguriert werden soll.

Ophelia-Modul Das *Ophelia-Modul* ist mit den *Sensoren* und *Aktuatoren* des Fahrzeugs verbunden, welche es jeweils über eine Schnittstelle observiert und bedient. Durch Mikrofone oder Kameras im Innenraum des Fahrzeugs können vom Ophelia-Modul No-Touch-Interaktionen des Nutzers erkannt werden. Mit diesen Interaktionen kann der Nutzer *Ophelia* bedienen und so u.a. sein Nutzerprofil mit Informationen zu statischen Einstellungen befüllen. Dies geschieht, indem das Ophelia-Modul mittels einer schreibenden Verbindung über das Internet die initial durchgeführten Einstellungen an den Server weitergibt. Dort werden die empfangenen Einstellungen in der Datenbank zum Nutzerprofil gespeichert. Ist keine Internetverbindung vorhanden, so werden die Daten übermittelt, sobald die Verbindung hergestellt werden konnte.

Alle Nutzer können u.a. über die No-Touch-Interaktion Einstellungen ändern bzw. anpassen. Hierbei ist es wichtig, den Begriff des Nutzers in Fahrer und Mitfahrer zu unterscheiden. Der Fahrer behält immer die Kontrolle darüber, welche Einstellungen vorgenommen werden, da er für die Sicherheit der sich im und um das Fahrzeug befindlichen Personen zuständig ist. Ebenso sind die Bedürfnisse des Fahrers wichtiger, als die der Mitfahrer. Um dies zu gewährleisten wird eine Priorisierung verwendet: Einstellungen, welche ein Mitfahrer vornimmt, werden nicht sofort durchgeführt, sondern zur Überprüfung an den Fahrer weitergegeben. Dieser kann die Einstellungsanfragen dann annehmen oder ablehnen.

Ophelia-Server Der *Ophelia-Server* verwaltet die *Datenbank* und die *KI*.

Auf der *Datenbank* werden die Nutzerprofile gespeichert, sowie alle Daten, die die KI zum Lernen der dynamischen Einstellungen benötigt.

In der Zeit nach der initialen Profilerstellung lernt *Ophelia* mit der *KI*, welche sich auf dem Server befindet, die Präferenzen des Nutzers in Bezug zu dynamischen Komforteinstellungen. Durch externe Webservices können weitere Informationen, wie z.B. Termine eines Kalenders des Nutzers, von der KI bezogen und verwertet werden. Auf Basis dieser Daten können dem Nutzer relevante Vorschläge gemacht werden, wie z.B. ob das Fahrzeug sich auf eine Fahrt zu einem im Kalender zu diesem Zeitpunkt eingetragenen Ort vorbereiten soll. Diese Anfragen können zusätzlich im Fahrzeug dem Nutzer direkt präsentiert werden, wo der Nutzer sie annehmen oder ablehnen kann.

Der Komfort der Nutzer soll durch die automatische Vornahme von Einstellungen, noch bevor der Nutzer das Fahrzeug betreten hat, erhöht werden. Durch die von der KI gelernten und in der Datenbank gespeicherten Einstellungen kann sich das Fahrzeug über den Server und das Ophelia-Modul selbsttätig den Wünschen und Bedürfnissen des Nutzers anpassen und alles für anstehende Fahrten vorbereiten.

4. Aufgabenstellung

4.3.2. Portabilität

Das zu entwickelnde Komfortsystem wird als Teilsystem eines Fahrzeugs entwickelt und soll in ein beliebiges Fahrzeug integriert werden können.

Es spezifiziert dabei unter anderem Schnittstellen, mittels derer auf Sensoren, Aktuatoren und Bedienelemente des Fahrzeugs zugegriffen werden können. Implementiert ein konkretes Fahrzeug diese Schnittstellen, können über die Sensoren und die Bedienelemente Eingaben gelesen, beziehungsweise über die Aktuatoren angesteuert werden. Der Anwendungsbereich von *Ophelia*, inklusive der Wahrnehmung der Umwelt, wird dabei auf den Fahrzeuginnenraum beschränkt.

Neben der Nutzung bereits vorhandener Sensoren, Bedienelemente und Aktuatoren können weitere Komponenten von *Ophelia* hinzugefügt werden.

Durch die Features und Funktionen von *Ophelia* werden die herausgearbeiteten Probleme und Nachteile von bestehenden Komfortsystemen adressiert und minimiert.

5. Projektmanagement

Projektmanagement ist die Anwendung von Wissen, Fähigkeiten, Werkzeugen und Techniken für Projektaktivitäten, um Projektanforderungen zu erfüllen. Projektmanagement ermöglicht es Organisationen Projekte effektiv und effizient durchzuführen.

Effektives Projektmanagement hilft Individuen, Gruppen und öffentlichen und privaten Organisationen unter anderem dabei Unternehmensziele zu erfüllen, Stakeholdererwartungen zu erfüllen und Erfolgsschancen für das Projekt zu erhöhen. Schlecht gemanagte Projekte, bzw. die Abwesenheit von Projektmanagement kann zu verpassten Deadlines, Kostenübersteigungen oder minderwertige Qualität führen.

In der Projektgruppe CHILL 2.0 werden alle Prozesse über das Projektmanagement definiert, um Aufgaben effektiv zu erarbeiten. Es sollen dadurch Strukturen geschaffen werden, die den Prozess der Zielerreichung unterstützen.

Im Abschnitt 5.1 Umfangsmanagement werden der Projektumfang, die Liefergegenstände definiert. Der Abschnitt 5.2 Anforderungsmanagement beschreibt, wie User Stories, Use-Cases und Anforderungen erhoben und dokumentiert werden. Im Abschnitt 5.3 Zeitmanagement wird das Vorgehensmodell des Projekts beschrieben sowie die Meilenstein- und Sprintplanung. Der Abschnitt 5.4 Qualitätsmanagement gliedert sich in das Test- und das Dokumentationsmanagement; Ebenso werden hier Verbesserungsmaßnahmen und Qualitätskriterien festgehalten. Im Abschnitt 5.5 Ressourcenmanagement werden die Expertenrollen und -teams der Projektgruppe beschrieben. Unter dem Abschnitt 5.6 Kommunikationsmanagement werden die Kommunikationsmethoden, -technologien und -regeln beschrieben sowie wie die Kommunikation zwischen den Stakeholdergruppen stattfinden soll. Im Abschnitt 5.7 Risikomanagement wird beschrieben, wie mit Unsicherheit im Projekt umzugehen ist; Hierzu gehören sowohl die Risiken als auch die sich ergebenen Chancen. Im Abschnitt 5.8 Tooldokumentation werden die verwendeten Werkzeuge der Projektgruppe erläutert.

Die im Kapitel 5 Projektmanagement betrachteten Inhalte basieren auf der sechsten Edition des PMBOK Guides vom Project Management Institute (PMI) [41]. Wenn andere Quellen hinzugezogen wurden, wurde dies gekennzeichnet.

5. Projektmanagement

5.1. Umfangsmanagement

Ein Teil des Umfangsmanagements ist die Erstellung des Umfangsmanagementplans. Der Umfangsmanagementplan ist Teil des Projektmanagementplans und beschreibt, wie der Projekt- und Produktumfang definiert, validiert und überwacht wird. Im agilen Projektmanagement wird der Umfang des Produkts über mehrere Iterationen definiert und bewilligt. Dieser Produktumfang wird in verschiedene Anforderungen (siehe Abschnitt 6.4 Anforderungskatalog) und zu leistende Arbeit heruntergebrochen.

In diesem Kapitel wird der Projektumfang (siehe Abschnitt 5.1.1 Projektumfang) und die dazugehörigen Liefergegenstände (siehe Abschnitt 5.1.2 Liefergegenstände) beschrieben.

5.1.1. Projektumfang

Der Projektumfang legt detailliert fest, was im Rahmen der Projektgruppe umgesetzt werden soll und was nicht. Eine Definition des Umfangs soll sicherstellen, dass alle Stakeholder die gleichen Erwartungen an das Projekt haben und die Zieldefinition kennen. Ein weiterer Vorteil eines definierten Projektumfangs ist, dass die Meilensteine leichter zu planen sind und das geprüft werden kann, ob das Projektziel zu erreichen ist.

Der Projektumfang wird von der Projektgruppe bestimmt und mit den Betreuern, der IAV und den Professoren abgesprochen. Der Projektumfang wird während des Projekts iterativ angepasst und definiert sowohl alle Aspekte und Funktionen von *Ophelia*, die umgesetzt werden sollen als auch die Grenzen von *Ophelia*. Die iterative Erweiterung wird durch das Vorgehensmodell (siehe Abschnitt 5.3.1 Vorgehensmodell) vorgesehen. Der Projektumfang wird im Folgenden durch die einzelnen Liefergegenstände definiert.

5.1.2. Liefergegenstände

Ein Liefergegenstand ist ein „eindeutiges und überprüfbares Produkt oder Ergebnis oder eine Dienstleistung, das/die hergestellt bzw. erbracht werden muss, um einen Prozess, eine Phase oder ein Projekt abschließen zu können“ [42]. Sie gliedern also den Projektumfang in kleinere überprüfbare Einheiten. Durch die definierten Liefergegenstände ist es möglich zu überprüfen, ob der gesamte Projektumfang oder nur Teile dessen erreicht wurden. Die Liefergegenstände müssen die jeweiligen Qualitätskriterien erfüllen. Die Liefergegenstände sind im Folgenden aufgelistet:

Zwischenbericht Ein Zwischenbericht ist ein Dokument, welches die Beschreibung des gesamten Inhalts der bisher von der Projektgruppe geleisteten Arbeit beinhaltet. Zu jedem Review muss der jeweilige Zwischenbericht über die Projektgruppenwebseite veröffentlicht werden. Die gesamte Dokumentation muss nach dem Dokumentationsmanagement im Abschnitt 5.4.1 Dokumentationsmanagement erfolgen.

Abschlussbericht Der Abschlussbericht beinhaltet den gesamten dokumentarischen Inhalt der Projektgruppenarbeit. Der Abschlussbericht entsteht über eine iterative Erweiterung der Zwischenberichte und ist zum Projektabschluss bei den Professoren, den Betreuern und der IAV einzureichen. Der Abschlussbericht muss ebenfalls über die Projektgruppenwebseite veröffentlicht werden. Die gesamte Dokumentation muss nach dem Dokumentationsmanagement im Abschnitt 5.4.1 Dokumentationsmanagement erfolgen.

Vision Die Vision definiert das Ziel von *Ophelia* und grenzt es von weiteren Systemen ab. Die Vision leitet sich aus den Problemen und Wünschen der Nutzer ab und stellt die Lösungsidee dar. Ebenso sollen die Teilkomponenten von *Ophelia* festgelegt werden. Die Vision muss mit den Betreuern und Professoren abgesprochen werden. Die Vision muss den Professoren beim 1. Review präsentiert werden und ist somit Teil des ersten Zwischenberichts.

User Stories Durch User Stories sollen Kernfunktionalitäten von *Ophelia* aus Sicht der Fahrer, Mitfahrer und Fahrzeughalter erfasst werden. Sie müssen aus der Vision und dem Leitfadeninterview abgeleitet werden. User Stories müssen nach den definierten Qualitätskriterien (siehe Abschnitt 5.2.2 Qualitätskriterien) erstellt werden. Die User Stories müssen den Professoren am 1. Review präsentiert werden und sind somit Teil des ersten Zwischenberichts. Durch das Vorgehensmodell der PG CHILL 2.0 können sie während des Projekts angepasst werden. Alle User Stories befinden sich im Abschnitt 6.2 User Stories Katalog.

Use Cases In einem Use Case Diagramm wird ein Teilsystem von *Ophelia* nach außen aus Sicht der Nutzer beschrieben. Sie stellen die Interaktion von *Ophelia* mit dem Nutzer dar. Use Cases werden aus den User Stories abgeleitet und müssen nach den definierten Qualitätskriterien (siehe Abschnitt 5.2.3 Qualitätskriterien) erstellt werden. Ebenso wie die User Stories sind sie Teil des ersten Zwischenberichts und müssen den Professoren am 1. Review vorgestellt werden. Alle User Stories befinden sich im Abschnitt 6.3 Use Case Katalog.

Anforderungskatalog Der Anforderungskatalog umfasst alle Anforderungen, die *Ophelia* definieren. In den Anforderungen muss gekennzeichnet werden, welche Anforderungen innerhalb des Projekts umgesetzt werden sollen und welche nicht. Anforderungen müssen nach den definierten Qualitätskriterien (siehe Abschnitt 5.2.4 Qualitätskriterien) und nach der Abb. 5.2 Schablone für Anforderungen nach Andreas Winter erstellt werden. Es muss deutlich werden,

5. Projektmanagement

woraus die jeweilige Anforderung abgeleitet wurde und wie diese umgesetzt ist. Zu jeder Anforderung müssen ebenfalls alle weiteren Informationen, welche in der Tabelle 5.2 Anforderungsvorlage definiert sind, dokumentiert werden. Durch das Vorgehensmodell ist es möglich die Anforderungen während des Projekts anzupassen.

Code Der Code von *Ophelia* ist am Ende des Projekts abzugeben. Der Code muss alle vorher definierten, umsetzbaren Anforderungen umsetzen und muss nach definierten Qualitätskriterien implementiert und dokumentiert werden.

5.2. Anforderungsmanagement

Im Anforderungsmanagement wird definiert, wie Anforderungen und die dazu relevanten Informationen erhoben und dokumentiert werden. Als Grundlage hierfür werden User Stories (siehe Abschnitt 5.2.2 User Stories) und Use Cases (siehe Abschnitt 5.2.3 Use Cases) genutzt. Ziel des Anforderungsmanagements ist es, dass sämtliche Projektgruppenmitglieder ein einheitliches Verständnis über das zu entwickelnde System und die Anforderungsdokumentation haben. Anforderungen werden im Abschnitt 6.4 Anforderungskatalog hinterlegt.

5.2.1. Workflow

Der erste Schritt ist die Identifikation aller Stakeholdergruppen durch eine Stakeholderanalyse (siehe Abschnitt 6.1 Stakeholderanalyse). Die Bedürfnisse der Nutzer wurden durch ein Leitfadeninterview (siehe Kapitel 2 Leitfadeninterview) identifiziert. Auf Basis der Stakeholderanalyse, des Leitfadeninterviews und der Vision werden User Stories erstellt, welche die Bedürfnisse der Stakeholder darstellen. Zur Erstellung der User Stories werden Nutzer von *Ophelia* betrachtet. Zur Verdeutlichung der Interaktion aller relevanter Akteure mit *Ophelia*, werden Use Cases erstellt. Zur Erhebung der Use Cases werden die User Stories in Kategorien unterteilt, sodass zu jeder Kategorie der User Stories mindestens ein Use Case erstellt wird. Auf Basis der User Stories und Use Cases werden Anforderungen abgeleitet.

Da User Stories und Use Cases nur Kernaspekte von *Ophelia* abbilden werden weitere Anforderungen zu den Teilfunktionalitäten von *Ophelia* erfasst, welche durch Brainstorming ermittelt werden. Diese Anforderungen werden als Soll-Anforderungen dokumentiert. Zunächst werden Top-Level Anforderungen definiert, welche die grundlegenden Funktionalitäten von *Ophelia* festhalten sollen. Davon können dann in den Teilfunktionalitäten Anforderungen anhand der User Stories und Use Cases abgeleitet werden. Für nicht-funktionale Anforderungen werden benötigte Normen und Standards herangezogen. Ein Beispiel zur Anforderungserhebung ist im Abschnitt 5.2.4 Beispiel einer Anforderungserhebung zu finden.

Anforderungen werden unabhängig von den Möglichkeiten der Projektgruppe erhoben. Die Arbeit des gesamten Prozesses wird in Arbeitsgruppen aufgeteilt, um diese zu parallelisieren.

5.2.2. User Stories

Durch User Stories sollen Funktionalitäten *Ophelia* aus Sicht der relevanten Stakeholder erfasst werden [5]. Diese dienen zur Ableitung von Use Cases und im Weiteren zur Anforderungserhebung. User Stories werden im Abschnitt 6.2 User Stories Katalog dokumentiert und in Kategorien unterteilt. Zur Erhebung von User Stories wurden innerhalb der Projektgruppe Qualitätskriterien definiert (siehe Abschnitt 5.2.2 Qualitätskriterien).

Schablone

User Stories werden in diesem Projekt in der folgenden Form erfasst:

Als [Nutzer] möchte ich [Funktionalität], um/damit [Nutzen] zu erreichen.

Qualitätskriterien

- Keine User Stories zum Speichern von Einstellungen erheben, sofern diese Einstellungen von *Ophelia* vorgenommen wurden
- Wenn *Ophelia* Einstellungen lernen soll, wird dies mit den folgenden Worten beschrieben: Als [Rolle] möchte ich, dass [zu lernendes Objekt] von *Ophelia* gelernt wird, damit/um [Nutzen]
- Zu den lernenden Objekten können beispielsweise Nutzerpräferenzen gehören.
- Einstellungen, welche in der *Ophelia-App* vorgenommen werden können, werden nicht der Kategorie App, sondern der zugehörigen funktionalen Kategorie zugeordnet
- User Stories enthalten keine Aufzählungen (wie z.B. und/oder)

5. Projektmanagement

5.2.3. Use Cases

Use Cases werden laut [43] wie folgt definiert: In einem Use Case Diagramm wird *Ophelia* nach außen aus Sicht der Nutzer beschrieben. Hierbei wird das grundsätzliche Verhalten bzw. die zusammenhängende Funktionalität von *Ophelia* definiert. Das Verhalten von *Ophelia* wird durch Anwendungsfälle beschrieben, die über Assoziationen mit der Umwelt interagieren.

Des Weiteren sind Use Cases aus Entwicklersicht konkreter als User Stories und sollen bei der Ableitung von Anforderungen helfen. Die Use Cases werden im Abschnitt 6.3 Use Case Katalog dokumentiert. Zur Erhebung von Use Cases wurden innerhalb der Projektgruppe Qualitätskriterien definiert (siehe Abschnitt 5.2.3 Qualitätskriterien).

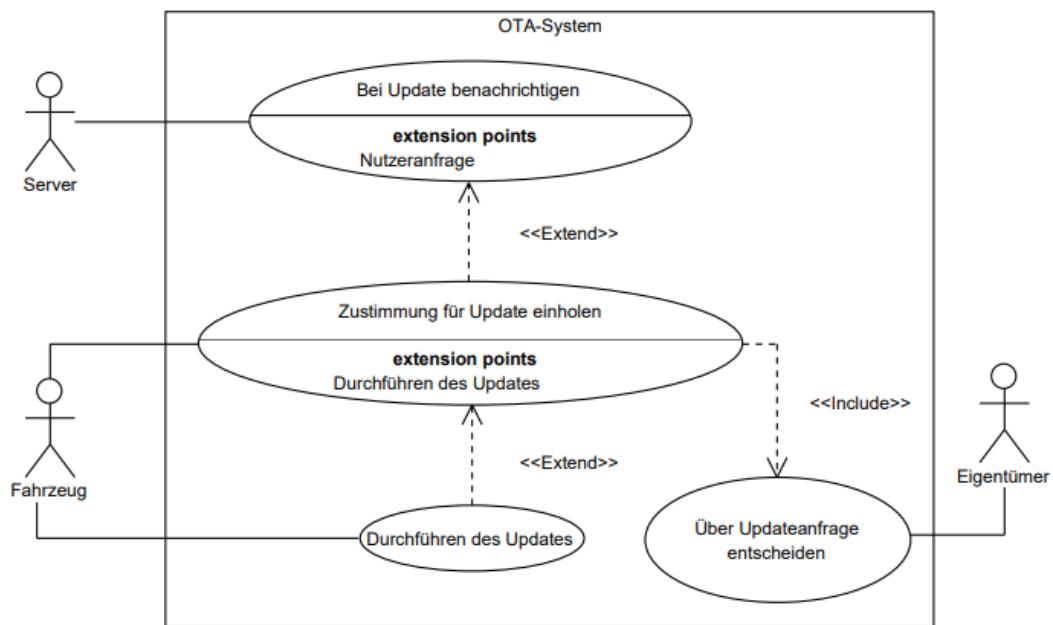


Abbildung 5.1.: Beispiel eines Use Case Diagramms aus dem Abschlussbericht der PG CHILL 1.0 [44]

Qualitätskriterien

- Use Cases sind eine Aktivität
- Innerhalb der Grenzen von *Ophelia* sollen nur Funktionalitäten stehen, die von ihr umgesetzt werden
- Als Extension Points sollen nur Nomen verwendet werden

- Erweiternde Use Cases sollen eine Notiz besitzen
- Die Notiz soll die Bedingung in der Form Condition: Bedingung enthalten, die zur Erweiterung des Use Cases führt
- Die Notiz enthält den Extension Point in der Form *Extension Point: Erweiterungspunkt*, durch welchen der Use Case erweitert wird
- Als Akteure werden Nutzer (Fahrer und Mitfahrer) und Fahrzeughalter (gewerblich und privat) aufgefasst. Des Weiteren werden, falls nötig, die Akteure *Lernender Agent* und *Nicht-lernender Agent* angegeben, um zu verdeutlichen in welchen Use Cases gelernt wird und in welchen nicht
- Für die Akteure Nutzer und Fahrzeughalter sind Fahrer und Mitfahrer bzw. gewerbliche und private Fahrzeughalter mitanzugeben
- Verbindungslien sind ausschließlich vertikal oder horizontal auszurichten

5.2.4. Anforderungen

Anforderungen erfassen die Bedürfnisse von Stakeholdern bzw. Eigenschaften eines Systems [45]. Sie dienen dem Zweck, das System genau zu definieren und sollen somit den Entwicklern verdeutlichen was umgesetzt werden soll. Zu den Eigenschaften eines Systems gehören neben den Funktionen u.a. ebenfalls Sicherheitsaspekte sowie Qualitätskriterien. Im Rahmen dieses Projektes werden Anforderungen in funktionale und nicht funktionale Anforderungen unterteilt. Funktionale Anforderungen werden aus den User Stories und den Use Case Diagrammen abgeleitet. Nicht-funktionale Anforderungen erfassen Sicherheitsaspekte sowie Qualitätskriterien von *Ophelia*. Diesbezüglich muss man sich bei der Anforderungserhebung an definierte Normen (z.B. ISO 26262) und Standards orientieren.

Eine Anforderung erfasst Spezifikationen von *Ophelia* auf verschiedenen Abstraktionsebenen. Dabei wird zwischen Top-LevelAnforderungen und Komponenten-Anforderungen unterschieden. Top-LevelAnforderungen enthalten übergeordnete Anforderungen von *Ophelia*. Dabei wird erfasst, welche Anforderungen an *Ophelia* gestellt werden.

KomponentenAnforderungen umfassen Anforderungen an die jeweiligen Teilkomponenten von *Ophelia*.

Die Erfüllung einer Anforderung wird durch Akzeptanztests nachgewiesen. Ein Akzeptanztest besteht aus drei Bereichen. Es werden Vorbedingungen definiert, sodass Aktionen durchgeführt werden können, wodurch die Anforderung überprüft werden kann. Schließlich wird überprüft, ob die Anforderung erfüllt ist.

5. Projektmanagement

Die Anforderungen werden im Abschnitt 6.4 Anforderungskatalog dokumentiert. Zur Erhebung von Anforderungen wurden innerhalb der Projektgruppe Qualitätskriterien definiert (siehe Abschnitt 5.2.4 Qualitätskriterien).

Funktionale Anforderungen

Funktionale Anforderungen nach [5]: Funktionale Anforderungen sind Anforderungen an die Funktionalität von *Ophelia*. Sie legen fest, was *Ophelia* leisten soll. Dazu gehören u.a. Aktionen, die ausgeführt werden und die Interaktion zwischen Benutzern und *Ophelia*. Beispielsweise werden Nutzereingaben und die entsprechenden Ausgaben von *Ophelia* aufgefasst.

Nicht-funktionale Anforderungen

Nicht-funktionale Anforderungen nach [5]: Nicht-funktionale Anforderungen sind Qualitätsanforderungen an *Ophelia*. Hier kann zwischen Anforderungen seitens der Entwickler und der Benutzer unterschieden werden. Als Entwickler fordert man ein gut dokumentiertes, verständliches Projekt, welches einfach validierbar ist. Als Benutzer hat man andere Anforderungen an *Ophelia*. Hier ist die Zuverlässigkeit bzw. die Genauigkeit sowie die Korrektheit und Sicherheit von *Ophelia* von Relevanz.

Die Tabelle 5.1 Aspekte für nicht-funktionale Anforderungen zeigt die im Rahmen dieses Projektes betrachteten Arten nicht-funktionaler Anforderungen.

Tabelle 5.1.: Aspekte für nicht-funktionale Anforderungen

Aspekt	Erklärung
Datenschutz	Datenschutzbestimmungen sollen eingehalten werden (welche Daten werden erhoben, wie werden Daten vor Außenstehenden gesichert)
Sicherheit	<i>Ophelia</i> soll der ISO 26262 bezüglich funktionaler Sicherheit genügen
Datenübertragung	Die Datenübertragung soll verschlüsselt erfolgen (welches Verfahren? Über welches Protokoll wird kommuniziert?)
Datenspeicherung	Wo sollen Daten gespeichert werden? Mit welchem Verfahren werden Daten verschlüsselt gespeichert?
Datenzugriff	Wer hat Zugriff auf Personenbezogene Daten? Wie wird der Zugriff auf diese Daten geregelt?

Antwortzeiten	In welchem Zeitrahmen (und unter welchen Bedingungen) sollen Antwortzeiten von <i>Ophelia</i> erfolgen?
Benutzerfreundlichkeit	<i>Ophelia</i> soll benutzerfreundlich sein
Genauigkeit und Präzision	Z.B. Temperaturgrenzen müssen eingehalten werden, in welchem Rahmen der Abweichung wird eine Einstellung als korrekt angesehen?
Randbedingungen für die Ausführung	Welche Laufzeitbedingungen müssen gelten, damit <i>Ophelia</i> funktioniert (Schnittstellen für Fahrzeuge auf Systemebene. Dies kann auch als Portabilität aufgefasst werden: Welche Schnittstellen muss ein Fahrzeug bereitstellen, damit <i>Ophelia</i> funktioniert?)

Identifikation von Anforderungen

Zur Identifikation von Anforderungen dienen die zuvor generierten User Stories sowie die Use Cases. Durch die User Stories wurden die Bedürfnisse der Stakeholder von *Ophelia* erfasst. Durch die Use Cases wurde die Interaktion des Nutzers mit *Ophelia* verdeutlicht. Somit kann auf diesen Grundlagen die Erhebung der Anforderungen erfolgen. Bei der Anforderungserhebung ist es wichtig, eine Top-Level-Systemanforderung bis auf die unterste Ebene herunter zu brechen.

Zunächst sollen Top-Level-Anforderungen erstellt werden, von denen weitere Anforderungen abgeleitet werden können. Top-Level-Anforderungen spezifizieren die Aufgaben von *Ophelia*. Diese wurden aus der Vision abgeleitet:

- Automatisch Voreinstellungen am Fahrzeug durchführen
- Vorschläge für Einstellungen machen
- Präferenzen des Nutzers lernen
- Nutzer erkennen
- Eine intuitive Nutzerschnittstelle bereitstellen
- Notwendige Touch-Eingaben und Tasten-Bedienungen des Nutzers minimieren
- Portabilität zwischen diversen Fahrzeugen ermöglichen

5. Projektmanagement

Die Top-Level-Ebene lässt sich in drei Komponenten unterteilen, die die Komponenten von *Ophelia* darstellen: das *Ophelia-Modul*, die *Ophelia-App* und der *Ophelia-Server* (siehe Kapitel 4 Aufgabenstellung). Die nächste Ebene bilden die Kategorien der User Stories.

Jede Anforderung der Komponenten soll aus den Top-Level-Anforderungen oder aus weiteren Quellen, wie zum Beispiel Normen, abgeleitet werden. Dies hat den Vorteil, dass die Anforderungen für die Implementierungsphase nach den Komponenten sortiert sind.

Dokumentation

Die generierten Anforderungen werden im Abschnitt 6.4 Anforderungskatalog hinterlegt. Für Anforderungen sind alle Felder der Vorlage (siehe Abschnitt 5.2.4 Vorlage) anzugeben. Mitglieder anderer Arbeitsgruppen überprüfen diese nochmals auf die Qualitätskriterien und auf die Formulierung. Der Verantwortliche für das Anforderungsmanagement muss dabei den Überblick behalten, dass Abhängigkeiten zwischen Anforderungen realisiert werden und dass Anforderungen nicht mehrfach entstehen.

Anforderungen müssen nach der Abb. 5.2 Schablone für Anforderungen nach Andreas Winter erstellt werden.

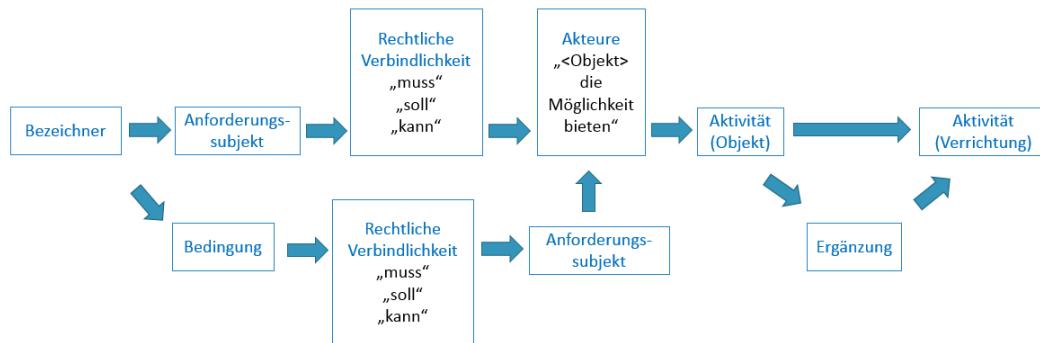


Abbildung 5.2.: Schablone für Anforderungen nach Andreas Winter [5]

Priorisierung Prioritäten sollen sinnvoll vergeben werden. Hierbei ist zu beachten, dass die Abhängigkeiten miteinbezogen werden müssen.

Die Priorisierung ist abhängig von den Faktoren:

- Abhängigkeiten zwischen Anforderungen (visuell darstellen, welche Anforderungen in Zusammenhang stehen)

- Was muss zunächst von der IAV umgesetzt werden, damit diese Anforderung realisierbar ist? (Absprache mit IAV/ Betreuer)

Die Priorisierung erfolgt über die rechtliche Verbindlichkeit, die über die Wörter *muss*, *soll* und *kann* festgelegt wird.

Vorlage Durch die Vorlage (siehe Tabelle 5.2 Anforderungsvorlage) werden die Punkte beschrieben, die bei allen Anforderungen definiert werden müssen.

Tabelle 5.2.: Anforderungsvorlage

Kategorie	Inhalt
ID	Eindeutige ID der Anforderung
Anforderungstyp	funktional oder nicht-funktional
Name	Name der Anforderung
Anforderungssatz	Beschreibung der Anforderung (nach Abb. 5.2 Schablone für Anforderungen nach Andreas Winter)
Nachgewiesen durch	Welcher Testfall beweist die Erfüllung der Anforderung
Akzeptanzkriterium	Vorbedingungen, Schritte (Was der Nutzer macht), Nachbedingungen
Abgeleitet von	Von welchen Anforderungen hängt diese Anforderung ab
Ursprung	User Story, Norm, von der die Anforderung eingebracht wurde
Referenzen	Verweis auf Dokumente bezüglich der Anforderung
Zuletzt geändert von	Name des Bearbeiters und Datum der Bearbeitung in Form: Name, Datum
Status	Status der Anforderung: offen, zu testen, erfüllt
Begründung	Begründung, warum diese Anforderung gebraucht wird
Wird umgesetzt	Wird diese Anforderung implementiert?

Eine ID wird im Format *Komponente-Anforderungstyp-Bezeichner* angegeben. Der Bezeichner soll ein Kürzel des Namens der Anforderung sein.

Qualitätskriterien

Die von der HOOD GmbH aufgestellten Qualitätskriterien für Anforderungen [46] wurden in dieser PG übernommen.

Inhaltlich

5. Projektmanagement

- **Identifizierbarkeit**

Eine Anforderung muss eindeutig als Anforderung erkennbar sein und einen Identifizierer (ID) besitzen.

- **Verständlichkeit**

Eine Anforderung muss für den adressierten Leserkreis verständlich sein.

- **Nachweisbarkeit**

Es gibt eine im Projektrahmen anwendbare Möglichkeit nachzuweisen, dass *Ophelia* die Anforderung erfüllt. Dies gilt nur für Anforderungen, die im Rahmen des Projektes auch implementiert werden.

- **Atomarität**

Eine Anforderung lässt sich nicht sinnvoll in weitere Anforderungen zerlegen.

- **Eindeutigkeit**

Eine Anforderung ist eindeutig formuliert, wenn sie genau eine Interpretation durch die erwarteten Leser zulässt.

- **Vollständigkeit**

Es fehlen keine relevanten Informationen innerhalb der Anforderung oder in einem definierten Anforderungsraum.

- **Korrekte Ableitung**

Die Anforderung ist in der Lage, das Bedürfnis eines Stakeholders oder eine übergeordnete Anforderung zu erfüllen.

- **Präzision**

Anforderungen sind mit möglichst kurzen Sätzen zu bilden. Lange, verschachtelte Sätze sollen vermieden werden.

- **Lösungsneutralität**

Anforderungen müssen so formuliert sein, dass (unter Berücksichtigung notwendiger, einschränkender Rahmenbedingungen) der maximale Lösungsraum genutzt werden kann.

- **Widerspruchsfreiheit**

Anforderungen stehen untereinander nicht im Widerspruch.

- **Notwendigkeit**

Der Grund für die Existenz der Anforderung ist nachvollziehbar (z.B. in Form eines Stakeholders oder einer übergeordneten Anforderung).

Formulierung Laut [5] und [46] sind die folgenden Punkte bei der Formulierung von Anforderungen zu beachten:

- Anforderungen werden im Aktiv formuliert
- Jeder Satz enthält genau eine Anforderung

- Bei Vergleichen soll der Bezugspunkt mit angegeben werden
- Bei Fallunterscheidungen sind alle möglichen Fälle anzugeben
- Spezifische Nomen sollen verwendet werden (keine generellen Terme wie z.B. die Daten. Dies gilt nicht für Top-Level Anforderungen)
- Redundanzen innerhalb einer Anforderung sind zu vermeiden

Beispiel einer Anforderungserhebung

Im Folgenden wird die Herleitung einer Anforderung zur Kategorie der User Stories *Musik* beschrieben. Hier soll die Top-Level-Anforderung „Ophelia muss automatische Voreinstellungen im Fahrzeug vornehmen“ betrachtet werden. Die betrachtete Komponente ist das *Ophelia-Modul*. Die zu betrachtende User Story lautet wie folgt: „**Als Nutzer möchte ich**, dass *Ophelia* einmalig meine Standardmusiklautstärke speichert, **damit** *Ophelia* meine Standardmusiklautstärke vor Fahrtantritt einstellt.“

Hierbei sind die folgenden Aspekte relevant:

- Die Lautstärke soll gespeichert werden. Da Anforderungen keine Interpretationen zulassen dürfen muss geklärt werden, **wo** die Lautstärke gespeichert werden soll.
- Wenn eine Lautstärke gespeichert werden muss, muss der Nutzer diese auch eingeben können. Also muss geklärt werden: Wie erfolgt die Eingabe der Lautstärke? Hier kann weiter unterteilt werden.
 - Die Eingabe kann über einen Sprachbefehl im Fahrzeug erfolgen
 - Die Eingabe kann über eine Geste im Fahrzeug erfolgen
 - Die Eingabe kann über die *Ophelia-App* erfolgen

Eventuell aufkommende Fragen zu den Inhalten/Begriffen müssen innerhalb der Gruppe bezüglich der Anforderungen geklärt werden. Aus den erfassten Aspekten zu jeder User Story und dem dazugehörigen Use Case können Anforderungen abgeleitet werden. Beziiglich der funktionalen Anforderungen wird aufgefasst, was genau *Ophelia* umsetzen muss, damit die Komponenten und somit die User Stories und Use Cases umgesetzt werden können.

Anforderungen werden so weit herunter gebrochen, bis für einen Entwickler aus einer Anforderung klar wird, was zu implementieren ist.

Wichtig bei der Anforderungserhebung ist die Priorisierung der Anforderungen durch die rechtlichen Verbindlichkeiten (muss, soll, kann).

5. Projektmanagement

5.3. Zeitmanagement

Ein Teil des Zeitmanagements ist die Erstellung des Zeitmanagementplans. Der Zeitmanagementplan ist Teil des Projektmanagementplans und beschreibt, wie und wann die Produkte, Services und Resultate, die im Projektumfang festgelegt wurden, geliefert werden. Zusätzlich fungiert der Zeitmanagementplan als Werkzeug zur Kommunikation, zur Definition von Stakeholdererwartungen und als Basis zur Performanceberichterstattung.

Ein aufstrebender Trend im klassischen Zeitplanmanagement ist die iterative Zeitplanung mit einem Backlog. Dies ist eine Form des *rolling wave planning*, basierend auf adaptiven Lebenszyklen, wie der agile Ansatz zur Produktentwicklung.

In diesem Kapitel werden im Abschnitt 5.3.1 Vorgehensmodell das Vorgehensmodell, im Abschnitt 5.3.2 Meilensteinplanung und -kontrolle die Meilensteinplanung und -kontrolle und im Abschnitt 5.3.3 Sprintplanung die Sprintplanung beschrieben. Im Punkt Vorgehensmodell werden die im Vorfeld betrachteten und das darauf basierend gewählte Vorgehensmodell vorgestellt. In der Meilensteinplanung wird beschrieben, wie Meilensteine erhoben und terminiert werden. In der Sprintplanung werden Punkte, wie die Sprintlänge, die prozedurale Zerlegung der Aufgaben, Aufgabenumfang sowie die Aufwandsabschätzung beschrieben.

5.3.1. Vorgehensmodell

Das Vorgehensmodell dient dazu, den Entwicklungsprozess zu strukturieren. Es gibt vor, wie die Zusammenarbeit im Team gelebt und welche Teilschritte in welcher Reihenfolge ausgeführt werden. Die richtige Vorgehensweise hängt dabei von diversen Faktoren wie z.B. der Art des zu entwickelnden Produktes, Zusammensetzung und Erfahrung des Teams, und dem Projektumfang ab. Im Rahmen dieser Projektgruppe wurden einige geläufige Vorgehensmodelle und Erfahrungen aus vorangegangen Projektgruppen, insbesondere der Vorgängergruppe CHILL 1.0, betrachtet. Aus diesen Erkenntnissen und Abschätzungen der Anwendbarkeit der einzelnen Modelle im Projektgruppenkontext wurde ein eigenes Modell abgeleitet.

Die Wahl des Vorgehensmodells hängt von diversen Projektumständen ab. Der Projektumfang von einem Jahr und die Gruppengröße von neun Teilnehmern zeigen die Notwendigkeit eines Vorgehensmodells, welches hilft, alle Arbeitsschritte zu koordinieren und strukturieren. Da keine Produktdefinition in Form eines Lastenheftes vorhanden ist und die Produktanforderungen von der Projektgruppe eigenständig erarbeitet werden müssen, muss die Möglichkeit zu mehreren Iterationen bei der Anforderungsdefinition bestehen. Zunächst entwickelte Konzepte und Ideen müssen anpassbar sein, falls sich im Laufe der Entwicklung herausstellt, dass sie in der Art und Weise nicht umsetzbar sind.

Da es sich um ein eingebettetes System in einem sicherheitskritischen Kontext handelt, gilt es, die Funktionen von *Ophelia* umfassend zu testen. Weiterhin zu beachten ist die geringe Vorerfahrung der Projektteilnehmer. Es benötigt Zeit, sich als Team in die Prozesse einzuleben. Außerdem ist der Projektumfang, aufgrund des anfänglich noch begrenzten Wissens in der Anwendungsdomäne in allen Teilschritten des Entwicklungsprozesses, schwer abschätzbar. Die Entwicklung eines Systems für ein beliebiges Serienfahrzeug macht es notwendig, dass zunächst allgemein definierte Anforderungen für den konkreten Prototypen noch präzisiert werden können.

Betrachtete Vorgehensmodelle

Es wurden verschiedene Vorgehensmodelle in Betracht gezogen und die Anwendbarkeit in unserem Kontext überprüft. Im Folgenden werden diese beschrieben:

Wasserfallmodell Das Wasserfallmodell ist ein sequenzielles Modell. An die Anforderungsanalyse schließt sich die Erstellung der Softwarearchitektur bzw. das Systemdesign, die Implementierung und die Modultests, die System- und Integrationstests und die Inbetriebnahme und Wartung an [47]. Iterationen sind nur zwischen aufeinanderfolgenden Schritten möglich.

Es ist in diesem Projektrahmen unbrauchbar, da es keine Möglichkeit bietet zu vorherigen Entwicklungsschritten zurückzukehren und so kaum Spielraum für Änderungen, Neukonzeptionierungen oder Erweiterungen im Laufe des Projektes bietet.

V-Modell Im V-Modell wird, ähnlich zum Wasserfallmodell, eine Ausarbeitung des groben Konzepts zur Implementierung verfolgt [48]. Dabei werden die Phasen der Anforderungsdefinition, des funktionalen Systementwurfs, der Erstellung der Systemarchitektur und der Komponentenspezifikation durchlaufen. Hieran schließt sich die Implementierung an. Dabei wird jedem dieser Schritte eine entsprechende Testphase gegenübergestellt. An die Implementierung angeschlossen werden die Phasen von der feingranularen Ebene der Modultests, über die Integrationstests und Systemtests bis hin zu den Akzeptanztests des fertigen Produkts durchlaufen.

Bei diesem Vorgehensmodell wird besonderer Wert darauf gelegt, dass während jeder Entwicklungsphase bereits Testfälle für die entsprechende Testphase erstellt werden. Dies führt zu einer guten Testabdeckung für die funktionalen Anforderungen an *Ophelia* auf allen Konzeptionsebenen. Daher ist dieses Modell für die Entwicklung sicherheitskritischer Systeme gut geeignet. Das V-Modell ist sinnvoll für einen definierten Testprozess. Nachteilig ist, dass es sich ebenfalls um ein sequenzielles Modell handelt, welches keine Iterationen zulässt.

5. Projektmanagement

Scrum Bei Scrum sowie im agilem Projektmanagement erfolgt die Entwicklung iterativ [49]. In Sprints von ein bis vier Wochen werden Teile der Anforderungen an *Ophelia* ausgearbeitet und implementiert. Das Ergebnis ist ein Inkrement, welches zusammen mit den Inkrementen vorheriger Sprints einen Teilprototypen von *Ophelia* darstellt. Scrum ist ein sehr flexibles Vorgehensmodell, welches die Möglichkeit bietet Änderungen von Anforderungen oder der Konzeptionierung miteinzubeziehen. Nachteilig ist, dass Scrum kein Vorgehen bietet wie eine Systemspezifikation schrittweise auf die zu implementierenden Module heruntergebrochen werden kann.

Vorgehensmodell nach CHILL 1.0 Die Vorgängergruppe CHILL 1.0 hat aus den oben genannten Gründen den V-Scrum-Ansatz entwickelt, welcher die Vorteile des V-Modells und Scrum vereint [44]. Dabei wurde jeder Entwicklungsschritt des V-Modells in Form von Scrum-Sprints realisiert. Im Nachhinein hat sich dabei gezeigt, dass durch den engen Zeitrahmen des Projektes für die Testphase zu wenig Zeit übrig geblieben ist. Außerdem konnte mit diesem Ansatz nicht so flexibel auf Änderungen reagiert werden, wie beim rein agilem Projektmanagement, sodass es am Ende nicht erfüllte Anforderungen gab.

Vorgehensmodell nach CHILL 2.0

Aus den Betrachtungen im Abschnitt 5.3.1 Betrachtete Vorgehensmodelle wurde ein eigenes Vorgehensmodell entwickelt. Dieses ist in der Abb. 5.3 Vorgehensmodell dargestellt.

Die Konzeptionierungsphase ist an das V-Modell angelehnt. Zunächst erfolgt die Anforderungserhebung. Daraufhin schließt sich die funktionale Spezifikation an. Schließlich werden die Systemkomponenten auf High Level bzw. Low Level Ebene spezifiziert. Dieser Ansatz wurde gewählt, um eine gut definierte Systemspezifikation und -architektur als Basis zu haben. Die Durchführung erfolgt innerhalb dieser einzelnen Phasen iterativ nach dem Scrum-Ansatz. Dies ist in der Abb. 5.3 Vorgehensmodell durch einen hellblauen Pfeil dargestellt.

Die Implementierungs- und Testphase umfasst mehrere Unterpunkte. Sie ist ebenfalls iterativ in Form von Sprints nach dem Scrum-Ansatz aufgebaut. Die Unterpunkte sind hierbei Implementierung, Unittest, Integrationstest und Systemtest. In der Abb. 5.3 Vorgehensmodell werden zur Verdeutlichung des Ansatzes die Unterpunkte von einem hellblauen Pfeil umkreist dargestellt.

Da im Kontext der sicherheitskritischen eingebetteten Systeme eine umfassende Testabdeckung angestrebt ist, sieht das Vorgehensmodell keine Verschiebung der Testphase an das Ende des Entwicklungsprozesses vor. Zu jeder Modulimplementierung sind entsprechende Testfälle zu definieren. Continuous Integration soll eingesetzt werden, um fortlaufend die Systemfunktionalität mittels Unit-, Integrations- und Systemtests zu gewährleisten. Dabei wird keine Richtung in Form von Pfeilen zwischen

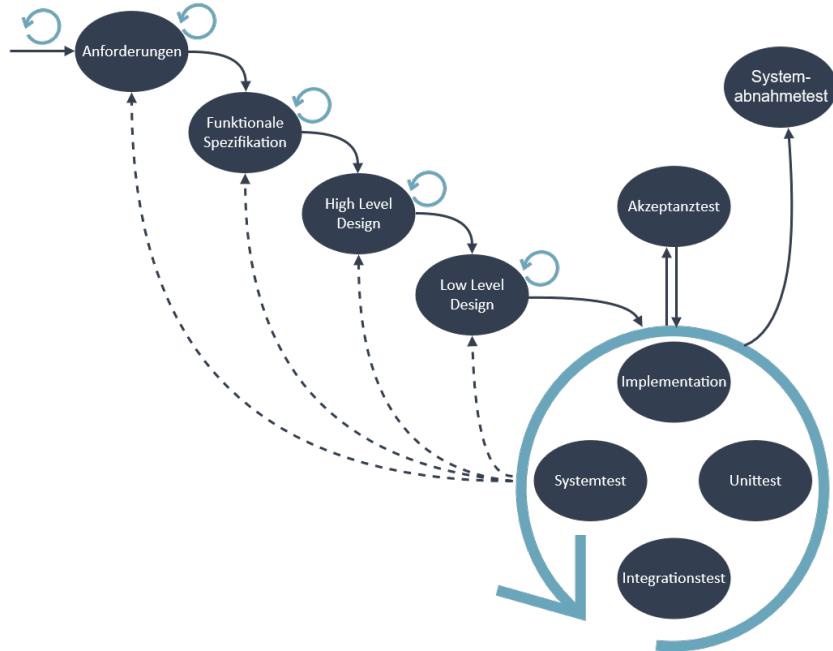


Abbildung 5.3.: Vorgehensmodell

Implementierung und Modultest vorgegeben, da auf niedriger Abstraktionsebene sowohl Test Driven Development als auch die Implementierung mit anschließender Testphase innerhalb des Vorgehensmodells möglich sein soll.

Für die Flexibilität während der Entwicklung die Anforderungen, die Systemspezifikation und/oder die Systemarchitektur anpassen zu können, besteht die Möglichkeit aus dieser Phase zu jeder Konzeptionierungsphase zurückzukehren. Dies ist durch die gestrichelten Pfeile dargestellt. Akzeptanztests sind dabei nicht in der Entwicklungs- und Testphase enthalten. Diese werden ausgeführt, um die Funktionalität des Prototypen beim Erreichen wichtiger Meilensteine zu verifizieren. Daher werden sie nicht in den kontinuierlichen Testprozess miteinbezogen, sondern bei Bedarf ausgeführt. Da vom Akzeptanztest wieder in die Entwicklungs-Testphase zurückkehrt wird, kann von dort auch zu jedem Konzeptionierungsschritt gegangen werden. Schließlich wird der gesamte Entwicklungsprozess durch den Systemabnahmetest am Ende der Projektlaufzeit abgeschlossen.

5.3.2. Meilensteinplanung und -kontrolle

Ein Meilenstein ist ein definiertes Ziel unter Berücksichtigung des SMART-Ansatzes (spezifisch, messbar, akzeptiert, realistisch, terminiert), welches zu einem definierten Zeitpunkt erreicht werden soll. Meilensteine werden im Zeitmanagement zur Engpasserkennung und Vermeidung einzelner Aufgabengebiete verwendet, welche

5. Projektmanagement

andernfalls den Gesamtzeitplan des Projektes gefährden könnten. Dies wird durch die Endgültigkeit eines Meilensteins gewährleistet, da dieser zur Deadline entweder erreicht oder nicht erreicht wurde. Die Deadline eines Meilensteins nach anfänglicher Festlegung zu ändern ist nicht möglich.

Die Meilensteinplanung liegt in der Verantwortung der Projektleitung. Der Projektumfang wird auf kleinere Ziele heruntergebrochen und terminiert. Bei diesem Prozess ist zu beachten, dass den jeweiligen Meilensteinen genügend Zeitpuffer eingeräumt werden, um diese auch bei unvorhergesehenen Ereignissen, im Rahmen der Möglichkeiten, noch erreichen zu können.

Zur Meilensteinkontrolle reflektiert die Projektleitung nach jedem abgeschlossenen Sprint, inwiefern die derzeitige Meilensteinplanung noch realisierbar, beziehungsweise ob der nächste Meilenstein bei aktuellem Kurs noch erreichbar ist. Hierbei wird der Aufwand grob abgeschätzt und mit den verfügbaren Ressourcen abgeglichen. Sollte der Meilenstein nicht mehr zu erfüllen sein, müssen Maßnahmen getroffen werden, um die Arbeitsweise dem Erreichen des Meilensteins anzupassen. Der Fall, dass ein Meilenstein nicht mehr erreicht werden kann und die daraufhin abgeleiteten Maßnahmen werden im Abschnitt 5.7 Risikomanagement beschrieben.

Ziele der Meilensteine sowie die jeweiligen Deadlines werden im Anhang (Kapitel C Meilensteinkatalog) dokumentiert.

5.3.3. Sprintplanung

Ein Sprint ist, wie im Abschnitt 5.3.1 Vorgehensmodell beschrieben, ein definierter Zeitraum, in welchem ein definiertes Ziel erreicht werden soll. Sprintziele werden von der Projektgruppe zu Sprintbeginn festgelegt und im Anhang Kapitel E Sprintkatalog dokumentiert. Diese werden innerhalb der Gruppe zunächst in kleinere Arbeitspakete (Epics) heruntergebrochen. Jedem Epic wird ein Epic-Beauftragter zugewiesen. Dieser ist dafür verantwortlich, dass die zugewiesenen Aufgaben von den jeweiligen Personen erledigt und in Jira (siehe Abschnitt 5.8.1 Jira) dokumentiert werden. Der Vorteil dabei liegt darin, dass keine Aufgaben vernachlässigt werden.

Im Weiteren werden diese in abschätzbare Aufgaben unterteilt. Die Arbeitspakete und die Aufgaben werden in Jira dokumentiert und gemanagt. Der Umfang dieser Aufgaben wird durch die Aufwandsabschätzung (siehe Abschnitt 5.3.3 Aufwandsabschätzung) definiert und soll den bestimmten Wert von acht Stunden in der Regel nicht überschreiten; vereinzelte, nicht logisch weiter herunterbrechbare Aufgaben können länger ausfallen.

Der Sitzungsleiter ist dafür verantwortlich, dass im Regelmeeting Weekly Scrum durchgeführt wird und Scrumtechniken, wie Sprintabschluss und Retrospektiven eingehalten werden. Da der jeweilige Sitzungsleiter diese Aufgaben übernimmt, ist hierfür kein separater Scrum Master nötig.

Der Sprintabschluss beinhaltet das Sprintreview und die Sprintretrospektive (siehe Abschnitt 5.4.2 Verbesserungsmaßnahmen). Der Abschluss dient dazu, den Betreuern und der IAV die erarbeiteten Ergebnisse vorzustellen und von ihnen Feedback zu erhalten. Durch den Sprintabschluss wird der Sprint geschlossen.

Aufwandsabschätzung

Unter Aufwandsabschätzung wird in der Softwaretechnik die Bewertung des Umfangs einer Aufgabe verstanden [50]. Im Rahmen der Projektgruppe wurde entschieden, dass Aufwandsabschätzungen durchgeführt werden. Im Folgenden werden drei der bekannteren Verfahren vorgestellt und gegeneinander abgewägt. Es existieren noch weitere Verfahren zur Aufwandsabschätzung, die Prinzipien in diesen unterscheiden sich jedoch nur in kleineren Aspekten, sodass die vorgestellten Verfahren für den Zweck des Projektes genügen.

Planning Poker Bei Planning Poker werden einzelne User Stories in Form von Story Points von jedem Mitglied bewertet. Jedes Mitglied hat dabei in der Regel eine Menge von Bewertungskarten mit Zahlen der Fibonacci-Folge, die die Story Points repräsentieren. Story Points sind hierbei keine fest definierte Einheit, sodass im Entwicklerteam ein gemeinsames Verständnis davon geschafft werden muss [51].

Pro

- Gut geeignet, wenn Fachwissen aus verschiedenen Bereichen aufeinandertrifft.
- Alle Mitglieder, aktive wie passive, erhalten gleiches Mitspracherecht.
- Auseinandersetzung mit der Aufgabenstellung wird erforderlich.
- Erfahrung wird spielerisch gesammelt.
- Teilnehmeridentifikation mit den Aufgaben wird gefördert.

Contra

- Bei einer gewissen Gruppengröße wird die Bewertung unübersichtlich und aufwendig.
- Ohne entsprechendes kontextbezogenes Fachwissen wird die Methode spekulativ.
- Wenn Einzelne nicht bei der Sache sind, wird das Ergebnis verzogen.
- Es kann teilweise sehr lange dauern und dadurch potenziell seinen Mehrwert verlieren.

5. Projektmanagement

T-shirt Sizing Bei T-Shirt Sizing handelt es sich um eine Alternative zum Planning-Poker, welche vor allem für unerfahrene Entwickler eine gute Einstiegsmethode zur Aufwandschätzung ist. Anstatt fixen Zahlen wie beim Planning Poker werden hierbei Aufgaben mit Größen wie Small, Medium, Large und Extra-Large bewertet [52].

Pro

- Die Aufwandsintervalle sind größer und damit ist die Wahrscheinlichkeit einer korrekten Einschätzung höher.
- Es ist weniger spezifische Fachkenntnis erforderlich.
- Erfordert weniger Zeit.
- Alle Mitglieder werden gleichermaßen beteiligt.

Contra

- Wenn einzelne nicht bei der Sache sind wird das Ergebnis spekulativ.
- Das Ergebnis ist nicht sehr spezifisch.

Affinity Mapping Beim Affinity Mapping sollen ähnliche Komponenten zusammen gruppiert werden. Es existiert eine Menge an Behältern mit verschiedenen Bewertungen. Dann wird die Aufgabe nach entsprechender Diskussion einem Eimer zugeordnet. Auf diese Weise werden die Aufgaben nach Umfang geordnet. [52].

Pro

- Förderung des Gemeinschaftsverhaltens.
- Intensive Auseinandersetzung mit jeder Aufgabe.
- Die Bewertung beruht auf Ähnlichkeiten, daher ist das Fachwissen weniger essentiell.
- Fachwissen des Einzelnen erhält mehr Aufmerksamkeit.

Contra

- Kann viel Zeit in Anspruch nehmen und dadurch seinen Mehrwert verlieren.
- Keine gleiche Berechtigung aller Gruppenmitglieder.
- Dominante Gruppenmitglieder setzen ihre Ansicht eher durch.

Fazit Es wurden drei verschiedene Verfahren zur Aufwandsabschätzung vorgestellt. Die Entscheidung darüber welche Methode im Rahmen dieses Projektes verwendet wird, wurde mit einer Mehrheitsentscheidung für Planning Poker gefällt. Planning Poker bietet die konkreteste zeitliche Bewertung der einzelnen Aufgaben. Das Konzept der Story Points ist für dieses Projekt durch Stunden ersetzt worden, sodass ein Story Point einer Stunde Aufwand entspricht. Die Anzahl der Story Points für eine Aufgabe wird ermittelt, indem der Median von allen Schätzungen ermittelt wird. Die Aufwandsabschätzung wird dabei für jede einzelne Aufgabe, welche Teil eines Sprints ist, durchgeführt. Die Projektgruppe ist sich bewusst darüber, dass die Bewertung anfangs einige Fehler enthalten und erst im Laufe des Projekts ihren vollständigen Mehrwert entfalten wird.

5.4. Qualitätsmanagement

Im Qualitätsmanagement wird das Planen, Managen und Überwachen von Qualitätsanforderungen beschrieben, die nötig sind, um Stakeholdererwartungen zu erfüllen. Im agilen Projektmanagement werden die Schritte zur Qualitätssicherung regelmäßig während des Projekts durchgeführt, anstelle von wenigen Qualitätssicherungsphasen am Ende eines klassischen Projekts. Ein Beispiel für wiederkehrende Qualitätssicherung ist der Prozess der Retrospektive am Ende eines Sprints, welcher eher auf die Verbesserung der Qualität der Arbeitsweise ausgelegt ist, als auf die Qualität der Ergebnisse.

In diesem Kapitel werden das Dokumentationsmanagement im Abschnitt 5.4.1 Dokumentationsmanagement und die Verbesserungsmaßnahmen im Abschnitt 5.4.2 Verbesserungsmaßnahmen beschrieben. Das Dokumentationsmanagement beschreibt die Art und Weise, wie gruppenintern dokumentiert werden soll und legt Richtlinien hierfür fest. Unter Verbesserungsmaßnahmen fallen sowohl kontinuierliche Verbesserungsmaßnahmen der angewandten Prozesse, als auch ein Leitfaden dafür, wie der Ablauf der (Dokumentations-)Reviews aussieht.

5.4.1. Dokumentationsmanagement

Das Kapitel Dokumentationsmanagement enthält den Reviewprozess für die Dokumentation (siehe Abschnitt 5.4.1 Dokumentationsreview). Außerdem wird anhand von Beispielen erklärt wie in L^AT_EX dokumentiert werden soll (siehe Abschnitt 5.4.1 Dokumentation in L^AT_EX), damit die Dokumentationsqualität gesichert ist.

5. Projektmanagement

Dokumentation in L^AT_EX

In diesem Abschnitt des Dokumentenmanagements werden Regel für die Formatisierung und Verwendung von Begriffen, Abkürzungen, Zitaten, Absätzen, Bildern, Tabellen, Labels sowie der Silbentrennung und der Versionshistorie beschrieben.

Begriffe & Abkürzungen Sämtliche Begriffe und Abkürzungen, die für das Projekt von Bedeutung sind oder mehr als einmal in einem von der Projektgruppe erstellten Dokument verwendet werden (auch übergreifend über verschiedene Dokumente), sind in der Datei Glossar.tex zu definieren. Außerdem sollen Begriffe und Abkürzungen schon während des Schreibens eines Textes direkt im Glossar definiert und verwendet werden. Weitere Hinweise zur Verwendung des Pakets Glossary unter Wikibooks: LaTeX/Glossary.

Um Begriffe in einem Text hervorzuheben, soll `\emph{}` verwendet werden.

Zitate Zitate werden mit dem Befehl `\enquote{}` gekennzeichnet und mit Hilfe von `\cite{QUELLE}` mit einer Quelle versehen.

Absätze Zur Erzeugung eines Absatzes wird eine leere Zeile in L^AT_EX verwendet und nicht `\\"`.

Silbentrennung Für den Fall, dass der L^AT_EX-Compiler Probleme mit der Silbentrennung eines Wortes hat, kann mit `"-` eine zusätzliche Stelle zur Trennung eines Wortes in einem Wort markiert werden. Beispiel: `er"-geb"-nis"-ori"-en"-tiert`. Der Compiler trennt das Wort dann an der am besten passenden Stelle. Außerdem soll `"=` statt eines normalen Bindestrichs `(-)` verwendet werden, damit die Trennung des Wortes auch an anderen Stellen möglich ist. Beispiel: `Touch"=Eingaben` erlaubt die Trennung Touch-Ein-ga-ben. Ohne die Nutzung von `"=` würde Touch-Eingaben nur an dem Bindestrich getrennt werden.

Weitere Hinweise zur Silbentrennung in L^AT_EX können unter Wikibooks: LaTeX-Wörterbuch: Silbentrennung gefunden werden.

Versionshistorie Die Versionshistorie soll für wichtige Veränderungen im Bericht, zu besonderen Terminen oder zu Meilensteinen einen neuen Eintag erhalten.



Abbildung 5.4.: Offizielles Logo der Projektgruppe Configurating & Helping, Independent Learning Limousine



Abbildung 5.5.: Beispiel für drei Bilder nebeneinander

Bilder Jedes Bild in der Dokumentation sollte eine Bildunterschrift und ein Label aufweisen.

```
\caption [Text im Abbildungsverzeichnis]{Text unter dem Bild}
\label{fig:label-des-Bildes}
```

Optional kann der Befehl *caption* einen kürzeren Text für das Abbildungsverzeichnis in eckigen Klammern enthalten. Label für Abbildungen beginnen immer mit *fig*:

Mehrere Bilder nebeneinander können beispielsweise mit *minipage* realisiert werden.

Tabellen Jede Tabelle sollte eine Überschrift und ein Label haben. Das Label für Tabellen beginnt mit *tab*:. Die Umgebung *tabularx* ermöglicht Tabellen mit dynamischer Breite und Tabellen, die bei Bedarf einen Seitenumbruch machen und dadurch länger als eine Seite sein können.

Tabelle 5.3.: Beispiel für eine Tabelle mit *tabularx*

Nr.	Test1	Text2	Text3
1	Text	Text	Hier könnte ein langer Text stehen, der viel oder sehr viel Platz braucht.
2	Text	Text	Hier könnte ein langer Text stehen, der viel oder sehr viel Platz braucht.

5. Projektmanagement

Mit dem Befehl `\keepXColumns` kann eine Tabelle, die die `tabularx` Umgebung verwendet, dazu gebracht werden die volle Textbreite zu nutzen, auch wenn der Text kürzer als eine Zeile ist. Mit `\convertXColumns` kann der Modus für die Spalten mit X wieder zurückgestellt werden.

Tabelle 5.4.: Tabelle mit `tabularx` und `keepXColumns`

Nr.	Text
1	kurzer Text
2	kurzer Text

Im Vergleich dazu die gleiche Tabelle mit `\convertXColumns`.

Tabelle 5.5.: Tabelle mit `tabularx` und `convertXColumns`

Nr.	Text
1	kurzer Text
2	kurzer Text

Die nachfolgende Tabelle enthält Zeilen und Spalten, die sich jeweils über mehrere Zeilen oder Spalten erstrecken. Dies wurde mit Hilfe der Befehle `\multirow{2}{}` und `\multicolumn{2}{c}{}` realisiert.

Tabelle 5.6.: Tabelle mit `tabularx`, `multirow` & `multicolumn`

Links	Zentriert	Rechts	Diese Spalte füllt den Rest der Zeile aus und wird automatisch umgebrochen, wenn zu wenig Platz ist.
Links	Zentriert	Rechts	Auch Text über mehrere Zeilen ist ohne Probleme möglich.
	Text über mehrere Spalten geht auch.		
Links	Zentriert	Rechts	Hier wieder normaler Text.
Links	Zentriert	Rechts	Hier wieder normaler Text.
	Auch ein Text über mehrere Spalten und Zeilen ist denkbar.		Hier wieder normaler Text.
			Hier wieder normaler Text.
			Hier wieder normaler Text.

Label Label für Tabellen und Bilder wurden bereits in dem Abschnitt 5.4.1 Bilder und dem Abschnitt 5.4.1 Tabellen beschrieben, werden jedoch der Vollständigkeit halber hier noch einmal mit aufgeführt. Die Label müssen wie folgt aufgebaut sein: *Typ:Kapitel-Unterkapitel-Abschnitt* beispielsweise: *subsec:projektmanagement-qualitaetsmanagement-dokumentationsmanagement*, damit die andere Autoren des

Dokuments erkennen können worauf mit dem dem Label referenziert werden kann. Umlaute und Unterstriche sind in einem Label nicht gestattet.

Tabelle 5.7.: Tabelle mit Labels für die Dokumentation

Art	Label
Kapitel	ch:
Abschnitt	sec:
Unterabschnitt	subsec:
Unterunterabschnitt	subsubsec:
Bilder	fig:
Tabellen	tab:
Gleichung	eq:
Codeliste	lst:
Listenelement (nummerierte Aufzählung)	itm:
Algorithmus	alg:
Anhang	app:
Paragraph	par:

Um auf Label zu referenzieren können unter anderem *nameref*, *autoref*, *cref* oder auch *ref* verwendet werden. Weitere Hinweise zu Labels und der Referenzierung unter Wikibooks: [LaTeX/Labels and Cross-referencing](#).

Dokumentationsreview

Dokumente, die in einen Zwischen- oder den Abschlussbericht einfließen, werden einem Reviewprozess (siehe Abb. 5.6 Workflow der Tasks in Jira) unterzogen. Dieser sieht vor, dass besagte Dokumente, von mindestens zwei Projektgruppenmitgliedern überprüft werden müssen, die nicht aktiv an der Dokumentenerstellung beteiligt waren. Das zweite Review wird von dem Review-Expertenteam durchgeführt. Das Expertenteam besteht aus Niklas Entrup, Marius Hubatschek, Noelle Rousselle und Tobias Schmitz.

Dokumente werden nach abgeschlossener Erstformulierung vom Autoren im Jiratask auf den Status *TOREVIEW1* gesetzt. Der erste Reviewer setzt den Status des Tasks auf *INREVIEW1*. Der Reviewer berichtet Grammatik- und Rechtschreibfehler direkt. Anmerkungen, Ideen, Inhaltliche Vorschläge oder Unklarheiten werden mit dem Befehl `\todo{}` in der tex-Datei dokumentiert. Abschließend setzt der Reviewer den Status des Tasks auf *TOCORRECT*. Der Autor überarbeitet den Text, kontaktiert bei Bedarf den Reviewer und setzt den Jiratask anschließend auf den *TOREVIEW2* Status. Der zweite Reviewer führt nun denselben Prozess, wie der erste Reviewer durch. Jedoch besteht die Möglichkeit den Task, wenn er akzeptiert werden kann, auf *DONE* zu setzen. Ansonsten wird der Task wieder zurück auf den Status

5. Projektmanagement

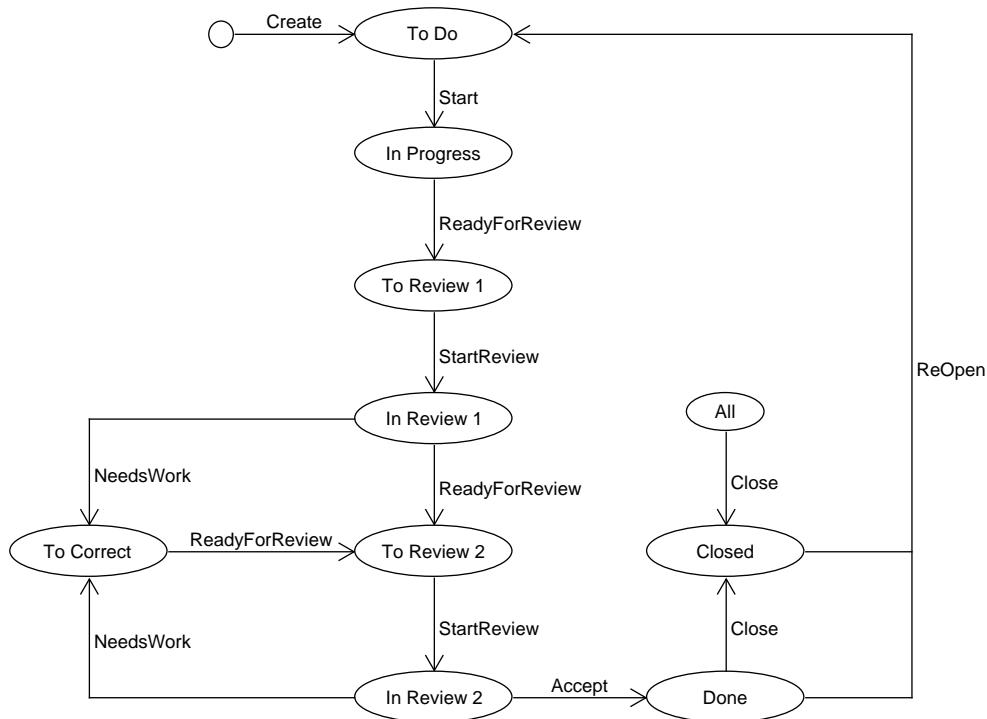


Abbildung 5.6.: Workflow der Tasks in Jira

TOCORRECT gesetzt und das zweite Review muss nach einer Bearbeitung des Inhalts durch den Autor wiederholt werden.

5.4.2. Verbesserungsmaßnahmen

Die Projektgruppe CHILL 2.0 besteht aus größtenteils unerfahrenen Entwicklern. Daher ist davon auszugehen, dass während des Projekts Arbeitsabläufe optimiert werden können. Dies geschieht durch die Iteration definierter Verbesserungsprozesse, welche im Folgenden beschrieben werden:

Sprintreview Das Sprintreview befasst sich mit der Erfüllung der im Sprint festgelegten Aufgaben und wird zum Ende eines Sprints durchgeführt. Hierbei wird sowohl die Qualität als auch die Quantität der bearbeiteten Aufgaben diskutiert. Darunter fällt die Anzahl der erledigten Aufgaben des Sprints, die gesamtheitliche Erfüllung der Epics und die Besprechung entstandener Probleme.

Bei der Durchführung des Sprintreviews werden die Aufgaben und Epics in Jira (siehe Abschnitt 5.8.1 Jira) betrachtet und besprochen. Für den Fall, dass

Aufgaben nicht erledigt worden sind, wird die Ursache hierfür identifiziert, dokumentiert und Maßnahmen abgeleitet und umgesetzt.

Der Vorteil des Sprintreviews ist, dass Probleme im Workflow frühzeitig identifiziert und behoben werden können. Zudem ermöglicht die Analyse der erledigten Aufgaben eine genauere Abschätzung des Umfangs des nächsten Sprints.

Sprintretrospektive Die Sprintretrospektive gewährt einen Einblick in die Arbeitsweise der Projektgruppe und wird nach dem Sprintreview durchgeführt. Der Fokus liegt hierbei auf der Verbesserung der Arbeitsweise und der Prozessoptimierung.

Hierzu werden in der Projektgruppe Verbesserungspotentiale aus dem Feedback der einzelnen Gruppenmitglieder abgeleitet. Dieses Feedback wird in die Bereiche *anfangen zu tun, aufhören zu tun, weitermachen, mehr von und weniger von* kategorisiert und im Anhang Kapitel E Sprintkatalog dokumentiert.

Der Vorteil der Sprintretrospektive ist die kontinuierliche Verbesserung der Arbeitsweise innerhalb der Projektgruppe. Ebenso dient sie dazu, die eigene Arbeitsleistung und die der Gruppe zu reflektieren.

Qualitätskriterien Qualitätskriterien bieten die Möglichkeit, Maßnahmen zur Qualitätsverbesserung zu schaffen und Ergebnisse zu überwachen und zu beurteilen. Dadurch können Probleme identifiziert und Gegenmaßnahmen eingeleitet werden. Langfristig kann durch die Umsetzung dieser Kriterien die Qualität der Ergebnisse gesichert werden. Die Qualitätskriterien werden in den jeweiligen Abschnitten, in denen sie Anwendung finden, beschrieben.

Qualitätsmanager-Expertenteam Das Qualitätsmanager-Expertenteam wurde eingeführt, um die Umsetzung der Qualitätskriterien in den einzelnen Artefakten zu gewährleisten (siehe Abschnitt 5.5.2 Expertenteams).

Review-Expertenteam Das Review-Expertenteam soll die Dokumentationsqualität sicherstellen und den Schreibstil vereinheitlichen (siehe Abschnitt 5.5.2 Expertenteams).

5.5. Ressourcenmanagement

Der Ressourcenmanagementplan ist Teil des Projektmanagementplans und beschreibt, wie Projektressourcen identifiziert, akquiriert und gemanagt werden. Es wird hierbei zwischen menschlichen und physischen Ressourcen unterschieden. Während sich menschliche Ressourcen auf die Teammitglieder, deren Kompetenzen und individuellen zeitlichen Einbringungsmöglichkeiten beziehen, betrachten physische Ressourcen die Verfügbarkeit von Materialien und Infrastruktur.

5. Projektmanagement

In dem Abschnitt 5.5.1 Experten werden die individuellen Aufgaben und Verantwortlichkeiten der einzelnen Projektteammitglieder beschrieben. Expertenteams sind Gruppen von Projektteammitgliedern, die sich auf einen bestimmten Bereich, wie beispielsweise der Umsetzung des Systems, spezialisiert haben (siehe Abschnitt 5.5.2 Expertenteams). Unter Weiterbildung fällt die projektspezifische Weiterbildung der Gruppenmitglieder (siehe Abschnitt 5.5.4 Weiterbildung). In dem Abschnitt 5.5.5 Kontrolle physischer Ressourcen ist die Sicherstellung von physischen Ressourcen beschrieben. Hierzu gehört zum Beispiel die Verfügbarkeit des Versuchsträgers und des Besprechungsraums sowie der Arbeitsmaterialien.

5.5.1. Experten

Die Entwicklung eines sicherheitskritischen Softwareprojekts ist vielfältig und umfasst unterschiedliche Aspekte und Aufgaben. Hierzu gehört unter anderem das Planen und Umsetzen des Projektmanagements oder das Erheben von Risiken. Es ist notwendig, dass diese Aufgaben in der Gruppe gemanagt und kontrolliert werden. Aus diesem Grund wurden in der Projektgruppe CHILL 2.0, für die Umsetzung notwendige, Expertenrollen definiert. Jeder der Gruppenmitglieder der Projektgruppe hat eine oder mehrere Expertenrolle/n zugeteilt bekommen. Ziel dieser Experten ist es, dass sichergestellt wird, dass sich mindestens ein Gruppenmitglied tiefergehend mit der jeweiligen Thematik auseinandersetzt und gegebenenfalls der Gruppe vorstellt. Zudem hat dies den Vorteil, dass jeder der Gruppenmitglieder bei Fragen oder Problemen einen konkreten Ansprechpartner hat. Durch diese Rollenverteilung ist es möglich während des Projekts Zeit zu sparen. Die Experten können während des Projekts dynamisch angepasst werden. Dies bedeutet, dass bei Bedarf neue Experten hinzugefügt oder gestrichen werden können. Im Folgenden (siehe Tabelle 5.8 Expertendefinition) werden alle bisher definierten Expertenrollen erläutert und die zugehörige/n Person/en genannt. Ebenfalls ist definiert mit welchen anderen Experten der jeweilige Experte aktiv in Kommunikation treten muss.

Tabelle 5.8.: Expertendefinition

Experten	Ansprechpartner	Definition und Abgrenzung	Kommunikation mit
Projektleitung	Niklas Entrup, Noelle Rousselle	Erstellung eines Projektplans. Durchsetzung von getroffenen Entscheidungen. Kontrolle und Einhaltung des Vorgehensmodells und von relevanten Terminen, Meilensteinen und sonstigen Zwischenzielen. Kontrolle anderer Beauftragter.	Allen
Öffentlichkeitsarbeit	Alexander Camu	Verwaltung der Webseite. Gestaltung des Logos. Organisation der Teilnahme an Veranstaltungen und Präsentationen des Projektes.	Projektleitung, Risikomanager
Anforderungsmanager	Daniel Ott	Erstellung eines Anforderungsplans. Erzeugung eines gemeinsamen Verständnisses über das zu entwickelnde System und die Anforderungsdokumentation.	Porsche Experte, Testbeauftragter, Sicherheitsbeauftragter, CHILL 1.0 Experte
Dokumentenbeauftragter	Tobias Schmitz	Erstellung eines Dokumentationsplans. Erstellung von Vorlagen für Dokumente und Präsentationen und die Pflege dieser Vorlagen. Verantwortung für die Umsetzung der Dokumentationsregeln.	CHILL 1.0 Experte

5. Projektmanagement

CHILL 1.0 Experte	Daniel Ott	Auseinandersetzung mit den Projektergebnissen der vorherigen Projektgruppe und Ansprechpartner für bereits konzeptionierte Schnittstellen von dieser. Kommunikation mit PG CHILL 1.0.	
	Mohammadjavad Ranji	Ansprechpartner für die hardware- und softwaretechnisch relevanten Komponenten des Versuchsträgers. Schnittstelle zur IAV.	Admin/Tool Beauftragter, CHILL 1.0 Experte
Testbeauftragter	Marius Hubatschek	Festlegung des Testumfangs und Sicherstellung eines gemeinsamen Verständnisses von diesem. Beaufsichtigung des Testumfeldes und Verantwortung für ein im Rahmen der Möglichkeiten vollständig getestetes System.	Projektleitung, Anforderungsmanager, Porsche Experte, Admin/Tool Beauftragter, Sicherheitsbeauftragter, Systemarchitekt, CHILL 1.0 Experte
Admin- & Toolbeauftragter	Madalin-Ioan Branea	Hilfestellung bei der Verwendung gemeinsam benannter Softwarewerkzeuge und in diesem Kontext auftretender Probleme. Verwaltung des CarPCs und Servers sowie Verantwortung für diese.	Projektleitung, Anforderungsmanager, Sicherheitsbeauftragter, Testmanager
Sicherheitsbeauftragter	Alexander Camu	Zuständigkeit für die Wahrung der funktionalen Sicherheit nach ISO 26262 im Rahmen der Möglichkeiten dieser Projektgruppe. Sicherheitstechnische Abstimmung mit der IAV.	Porsche Experte

Systemarchitekt	Madalin-Ioan Branea	Ansprechpartner für die Erstellung von Architekturen. Interface zwischen Entwickler und Außenstehenden. Überblick über die gesamte Architektur.	Anforderungsmanager, Porsche Experte, CHILL 1.0 Experte
Risikomanager	Marius Hubatschek, Tobias Schmitz	Abschätzung und Erstellung eines Plans für Risiken und Chancen. Verantwortung für die Umsetzung des Plans. Abstimmung mit der IAV über Lieferstatus von Komponenten und Standort des Porsche.	Projektleitung, Anforderungsmanager, Porsche Experte

Expertenrollenänderungen

Tabelle 5.9.: Expertenrollenänderungen

Rolle	Ehemaliger Experte	Aktueller Experte	Datum	Begründung
Risikomanager	Artem Jarovoj	Marius Hubatschek, Tobias Schmitz	12.06.2019	Den Mitgliedern der Projektgruppe war der Umfang der Rolle am Anfang des Projekts nicht bewusst. Zudem hatte die Projektgruppe keine Erfahrung mit der Erstellung eines Risikomanagements. Aufgrund des hohen Aufwands wurde die Rolle auf zwei Personen aufgeteilt.
Öffentlichkeitsarbeit	Alexander Camu, Noelle Rousselle	Alexander Camu	24.06.2019	Entlastung von Noelle Rousselle durch Abgabe einer Expertenrolle
Systemarchitekt	Arne Hues	Madalin-Ioan Branea	01.07.2019	Austritt von Arne aus der Projektgruppe

5. Projektmanagement

5.5.2. Expertenteams

Die Entwicklung einer Software kann sehr komplex werden. Dies stellt Entwickler vor die Herausforderung diverse Wissensgebiete in einem Softwareentwicklungsprojekt zu vereinen. Eine Möglichkeit dies zu realisieren ist die Erstellung von Expertenteams, welche sich auf bestimmte Entwicklungsbereiche fokussieren.

Im Folgenden werden die in der Projektgruppe vertretenen Expertenteams vorgestellt:

Review-Experten Das Review-Expertenteam ist für die Durchführung des zweiten Reviews jeglicher Dokumentation verantwortlich. Der Vorteil hierbei ist, dass dies in einem einheitlichen Dokumentationsstil resultiert. Ebenso wird hierdurch die Einhaltung des Reviewprozesses sichergestellt.

Qualitätsmanagement-Experten Die Qualitätsmanagement-Experten sind dafür zuständig, definierte Qualitätskriterien durchzusetzen und den Projektgruppenmitgliedern ein einheitliches Verständnis von diesen zu vermitteln.

5.5.3. Personalplanung

Bei der Ressourcenplanung ist ebenfalls die Personalplanung bzw. die Mitgliederplanung zu beachten. Hierzu gehört die zeitliche und örtliche Verfügbarkeit der Gruppenmitglieder.

Jedes Gruppenmitglied der Projektgruppe CHILL 2.0 hat ein Anrecht auf 28 Tage Urlaub. Der Urlaub muss mindestens 30 Tage im Voraus eingereicht bzw. angemeldet werden. Dadurch ist es möglich, die Sprintplanung auf die Verfügbarkeit der Gruppenmitglieder anzupassen. Der März 2020 ist der letzte Monat vor der Abgabe des gesamten Projekts. In diesem Monat darf kein Urlaub mehr genommen werden. Dadurch soll gewährleistet werden, dass in diesem Monat alle Mitgliederressourcen vollumfänglich genutzt werden können, um eventuell noch nicht erledigte Aufgaben fertigzustellen.

Die wöchentliche Arbeitszeit richtet sich nach dem Leitfaden der Projektgruppen [53]. Jedoch ist Gleitzeit erlaubt. Dies bedeutet, dass es möglich ist, Arbeit vor- oder nachzuarbeiten. Jedes Gruppenmitglied ist demnach für die Organisation seiner Arbeitsstunden selbst verantwortlich. Dieses Konzept basiert auf Vertrauen innerhalb der Gruppe. Aus diesem Grund müssen keine Stunden erfasst werden. Besteht der Verdacht, dass ein Gruppenmitglied zu wenig Arbeit leistet, müssen, auf Anfrage der Projektmanager, Arbeitsstunden erfasst werden. Wird nach dieser Maßnahme die Arbeitsleistung nicht erhöht, wird ein Gespräch mit den Betreuern eingeleitet.

Während der gesamten Zeit der Projektgruppe gibt es einen wöchentlichen Termin zur Besprechung aller wichtigen Inhalte, der weiteren Planung und zum Weekly Scrum. Dieser Termin ist ein Pflichttermin für alle Gruppenmitglieder. An diesem

Termin sind ebenfalls die Betreuer und die IAV beteiligt. Falls ein Gruppenmitglied nicht an diesem Termin teilnehmen kann, muss dieses mindestens eine Stunde im Voraus gemeldet werden. Bei unentschuldigtem Fehlen, müssen 50 Euro in die Gruppenkasse eingezahlt werden. Ebenfalls müssen fünf Euro beim Zuspätkommen gezahlt werden. Für jedes Zuspätkommen erhöht sich dieser Wert um fünf Euro für die jeweilige Person.

Des Weiteren wurden im ersten Semester während der Vorlesungszeit zwei weitere Arbeitstermine zur Bearbeitung der Aufgaben festgelegt (Mittwoch 12 bis 20 Uhr, Freitag 12 bis 20 Uhr). Alle Gruppenmitglieder müssen mindestens an einem dieser beiden Termine teilnehmen. Die Bearbeitung der Aufgaben in der Gruppe hat den Vorteil, dass Fragen und Probleme schneller geklärt werden können.

5.5.4. Weiterbildung

Innerhalb der Projektgruppe haben die Gruppenmitglieder unterschiedliche Vorkenntnisse und Erfahrungen. Dies liegt daran, dass unter anderem an unterschiedlichen Universitäten der Bachelor absolviert wurde, an unterschiedlichen Studiengängen oder an unterschiedlichen Schwerpunkten im Studium. Damit alle Gruppenmitglieder erfolgreich an der Projektgruppe teilnehmen können und ein hinreichendes Wissen über das Projekt haben, müssen sich die einzelnen Mitglieder fortbilden. Diese Fortbildung kann über verschiedene Wege geschehen:

Experten Wie im Kapitel 5.5.1 beschrieben, haben Gruppenmitglieder unterschiedliche Expertenrollen und damit auch unterschiedliche Kenntnisse. Kenntnisse die wichtig für alle Mitglieder sind, müssen vorgestellt werden. Hierzu gehört zum Beispiel die Vorstellung des Anforderungsmanagements.

Präsentationen Jedes Gruppenmitglied hat die Aufgabe ein Seminarthema zu präsentieren. Diese Themen sind auf das Projekt und den Projektstand zugeschnitten und liefern den Mitgliedern wichtige Inhalte.

Recherche Neben der Erweiterung der Kenntnisse durch andere Gruppenmitglieder, hat jeder die Aufgabe sich, wenn nötig, weiterzubilden. Dies kann über Fragen oder Recherchen im Internet oder durch Bücher geschehen.

5.5.5. Kontrolle physischer Ressourcen

Neben den personellen Ressourcen, bestehen, wie in der Einleitung beschrieben, auch physische Ressourcen. Im Folgenden werden diese beschrieben und deren Kontrollprozess definiert:

5. Projektmanagement

Versuchsträger Als Versuchsträger fungiert der *Porsche Panamera 4 E-Hybrid Sport Turismo*, welcher von der IAV für die Dauer des Projekts gestellt wird. Dieser ist im Regelfall auf dem Parkplatz des OFFIS Gebäudes lokalisiert und steht der Projektgruppe zur Verfügung. Sollte dies nicht der Fall sein, ist die Projektgruppe im Voraus in Kenntnis zu setzen.

Arbeitsraum Der Projektgruppe steht der Arbeitsraum U104 für die gesamte Projektlaufzeit zur Verfügung. Diesen teilt sich die Projektgruppe durch definierte Zeitslots mit einer weiteren Projektgruppe. Sollte der Raum nicht zur Verfügung stehen, ist die Projektgruppe im Voraus in Kenntnis zu setzen.

Arbeitsmaterial Unter Arbeitsmaterial fallen jegliche materielle Gegenstände, welche ein Projektgruppenmitglied benötigt, um die erwartete Arbeit leisten zu können. Alle benötigten Arbeitsmaterialien sind zu den Gruppentreffen mitzubringen. Projektgruppenmitglieder managen deren Arbeitsmaterialien eigenverantwortlich.

5.6. Kommunikationsmanagement

Im Kommunikationsmanagement wird beschrieben, wie die Kommunikation innerhalb des Projektes stattfinden soll. Kommunikation beschreibt die Möglichkeit, Informationen zu senden und zu erhalten. Dies kann sowohl durch Kommunikationsaktivitäten, wie Meetings und Präsentationen, als auch durch Kommunikationsartefakte, wie E-Mails, Soziale Medien, Projektberichte oder Projektdokumentation, geschehen.

Erfolgreiche Kommunikation wird in zwei Teile unterteilt: Die Erstellung einer angemessenen Kommunikationsstrategie, basierend auf den Bedürfnissen des Projekts und der Projektstakeholder und der darauf basierende Kommunikationsplan. Dieser regelt, dass die nötigen Informationen an die jeweiligen Stakeholder in unterschiedlichen Formaten und über unterschiedliche Wege zugestellt werden können.

Kommunikationsinhalte sind unter anderem Ideen, Annahmen und Probleme (siehe Abschnitt 5.6.1 Kommunikationsinhalte). Methoden und Technologien helfen bei der Umsetzung des Kommunikationsvorhabens (siehe Abschnitt 5.6.2 Methoden und Technologien). Die vorgestellten Kommunikationsregeln gliedern sich in das Feedbackmodell und die interkulturelle Kommunikation (siehe Abschnitt 5.6.3 Kommunikationsregeln). Im Abschnitt 5.6.4 Kommunikation mit Stakeholdern wird die Kommunikation mit den einzelnen Stakeholdergruppen beschrieben.

5.6.1. Kommunikationsinhalte

Zu den zu kommunizierenden Inhalten gehören folgende Punkte:

Ideen sind während der Gruppentreffen mündlich anzubringen, im Protokoll aufzunehmen und im Wiki unter *Konzepte, Ideen und Vorschläge* festzuhalten. Gruppeninterne Ideen können im Vorfeld über WhatsApp besprochen werden.

Herausforderungen sind gruppenintern zu besprechen. Zugehörige Lösungsmöglichkeiten sind zu identifizieren und anschließend im Anhang Kapitel B Herausforderungskatalog festzuhalten.

Feedback ist regelmäßig zu geben und wird nach den im Abschnitt 5.6.3 Feedback vorgestellten Regeln durchgeführt.

Fehler sind transparent im Anhang Kapitel B Herausforderungskatalog festzuhalten und für die IAV und die Betreuer zugänglich zu machen.

Aufgaben werden im Projektteam während der Sprintplanung bestimmt und in Jira festgehalten. Projektteammitglieder sollen sich eigenverantwortlich über zu erledigende Aufgaben durch Jira informieren und zu bearbeitende Aufgaben als solche kennzeichnen, um redundante Arbeit zu minimieren.

Organisatorisches ist in der Projektgruppe und mit der IAV und den Betreuern zu besprechen und im Wiki unter *Bisher getroffene Entscheidungen* festzuhalten.

5.6.2. Methoden und Technologien

Bestimmte Methoden und Technologien werden eingesetzt, um die Kommunikation gruppenintern und stakeholderübergreifend effizienter zu gestalten und um sicherzustellen, dass die Gruppenmitglieder, die IAV und die Betreuer die nötigen Informationen zur Verfügung haben. Bei der Bestimmung und bei dem Einsatz der Methoden und Technologien wird besonderen Wert auf die Dringlichkeit der zu überliefernden Informationen, der Verfügbarkeit und Zuverlässigkeit der Technologien, der Nutzungseinfachheit und der Sensibilität und Geheimhaltung der Informationen gelegt.

Meetings sind einmal wöchentlich verpflichtend gruppenintern wahrzunehmen. Bei diesen Meetings nehmen ebenso Vertreter der Stakeholdergruppen der Betreuer und der IAV teil. Vertreter der IAV nehmen an diesen Treffen in der Regel über Skype teil. Neben diesen Treffen sind wöchentlich zwei weitere Arbeitsmeetings etabliert, die bei Bedarf wahrgenommen werden sollten. Für die Meetings wird wochenweise ein Protokollant bestimmt, der für die Protokollierung der Inhalte im Meeting verantwortlich ist. Dieser bereitet in der darauffolgenden Woche die weiteren Meetings vor und leitet diese.

Präsentationen werden als Werkzeug zur Informationsübertragung genutzt. Diese können sich von gruppeninternen Präsentationen, zum nahebringen bestimmter Modelle oder Technologien, bis hin zu Abschlusspräsentationen, die die Gesamtheit des Projektes umfassen, beziehen.

5. Projektmanagement

E-Mail wird zur formellen Informationsübertragung genutzt. E-Mails dienen, neben den Regelmeetings, als Hauptkanal für die stakeholdergruppenübergreifende Kommunikation. Es gibt einen E-Mail Verteiler, der die Stakeholdergruppen IAV, Betreuer und Projektgruppe und einen E-Mail Verteiler, der die Stakeholdergruppen Betreuer und Projektgruppe beinhaltet.

WhatsApp wird zur informellen, projektgruppeninternen Kommunikation genutzt.

Skype wird neben dem oben genannten Einsatz der IAV zur Teilnahme an den Meetings auch als Werkzeug der Kommunikation außerhalb der Meetings genutzt. Ebenso bietet Skype und auch Discord physisch abwesenden Projektgruppenmitgliedern die Möglichkeit sich in den Arbeitsmeetings miteinzubringen.

PowerPoint wird zur Erstellung und zum Abhalten von Präsentationen genutzt.

Jira wird zur projektgruppeninternen Aufgabendefinition und -verteilung genutzt.

5.6.3. Kommunikationsregeln

Kommunikation ist in einem Projekt unerlässlich und muss bestimmten Regeln und Modellen folgen. Durch diese Regeln soll die Qualität der Kommunikation im Projekt gesteigert werden.

Generell soll in der Projektgruppe offen kommuniziert werden, damit keine Missverständnisse entstehen. Dazu gehört, dass man seine Intention verdeutlicht und nachfragt, wenn etwas unklar ist. Ebenso soll die Kommunikation untereinander respektvoll sein. Jedes Gruppenmitglied in der Gruppe ist gleichwertig und wird mit Respekt behandelt.

Im Weiteren werden genaue Regeln für das Geben und Nehmen von Feedback (siehe Abschnitt 5.6.3 Feedback) definiert, sowie Regeln zur Kommunikation (siehe Abschnitt 5.6.3 Interkulturelle Kommunikation).

Feedback

Feedback ist wichtig, um sich weiterzuentwickeln und um sich selbst aus einer objektiven Sicht zu reflektieren. Ebenso ist es durch offenes Feedback möglich, Probleme und Missverständnisse anzusprechen und konstruktiv zu lösen. Dadurch wird das gegenseitige Verständnis in der Gruppe gestärkt. Damit Feedback nicht verletzend für den Feedbackempfänger ist, soll das Feedback mit den folgenden Regeln [54] professionell umgesetzt werden:

- Beschreibend statt bewertend
- Klar und genau formuliert
- Sachlich richtig

- Ohne moralische Verurteilung
- Konkret im Gegensatz zu allgemein
- Bezug auf Beobachtungen im Gegensatz zu Vermutungen, Phantasien oder Interpretationen.
- Veränderbare Verhaltensweisen ansprechen und nicht Unzulänglichkeiten, auf die der Betreffende relativ wenig oder gar keinen Einfluss nehmen kann.
- Erbieten im Gegensatz zu aufzwingen
- Berücksichtigung der Bedürfnisse des Empfängers
- Zur rechten Zeit oder möglichst bald

Interkulturelle Kommunikation

Interkulturelle Kommunikation ist ein relevantes Thema, wenn in einem Projekt kulturelle oder sprachliche Barrieren bestehen. Die Projektgruppe ist eine definiert deutschsprachige Lehrveranstaltung. Das Verständnis der Sprache ist notwendig, um an dieser teilzunehmen. Bei auftretenden Problemen ist das Projektgruppenmitglied in der Verpflichtung diese eigenverantwortlich anzusprechen.

5.6.4. Kommunikation mit Stakeholdern

Art und Inhalt der Kommunikation kann, je nach Stakeholder, an welchen die Kommunikation gerichtet ist, variieren. Im Folgenden werden die Kommunikationsinhalte und -wege stakeholderspezifisch dargestellt.

Gruppenintern wird die Kommunikation informell gehalten. Methoden, wie Gruppenmeetings und Technologien, wie WhatsApp, stehen im Vordergrund. Gruppenintern werden alle projektrelevanten Informationen besprochen, um so eine möglichst effiziente Kommunikation zu realisieren.

Betreuer werden projektphasenübergreifend in die Konzeption und Entwicklung miteinbezogen. Sie geben Feedback zum Projekt als Ganzes und zur Arbeitsweise und dienen als Schnittstelle zu den Professoren.

IAV Vertreter werden projektphasenübergreifend in die Konzeption und Entwicklung miteinbezogen. Sie geben Feedback und dienen als Ansprechpartner für Modifizierungen, die am Versuchsträger durchgeführt werden sollen. Anfragen zur Modifizierung werden von der Projektgruppe über das Redmine Ticketsystem eingereicht.

Professoren werden an definierten Terminen über den jeweiligen Projektstand informiert und beurteilen diesen.

5. Projektmanagement

Nutzer werden in Planungs- und Kontrollprozessen miteinbezogen, um zu gewährleisten, dass das zu entwickelnde Produkt den Nutzerwünschen entspricht.

5.7. Risikomanagement

Viele Faktoren wirken auf ein umfangreiches Projekt und sein Team ein. Diese sind jedoch im Allgemeinen nicht klar definiert und haben des Weiteren eine zeitliche Facette, weshalb es in Projekten häufig zu Rückschlägen durch nicht vorhergesehene, unterschätzte oder nicht für möglich gehaltene Einflüsse kommt.

Das Risikomanagement setzt sich damit auseinander, diese Einflüsse und Faktoren zu identifizieren, in das Projekt einzuordnen, präventive und korrigierende Maßnahmen festzulegen und durchzusetzen. Dazu orientiert sich die Projektgruppe an [55], wo ein Kreislauf zum Risikomanagement in agilen Projekten bestehend aus Risikoidentifikation, Risikoklassifikation, Risikoquantifizierung, Risikoplanung und Handeln vorgestellt wird. Dieses Vorgehen wird im Folgenden näher erläutert.

5.7.1. Risikoidentifikation

Die Risikoidentifikation findet in der Form von Brainstorming während der Sprintplanung statt. In ihr geht es darum, Stärken und Schwächen sowie Chancen und Risiken für das Team und das Projekt zu identifizieren. Dazu werden zwei Dimensionen von Faktoren betrachtet: Die Dimension der Quelle und die Dimension des Einflusses auf das Projekt. Dabei kann die Quelle eines Faktors projektintern oder projektextern liegen, während für den Einfluss eines Faktors unterschieden wird, ob dieser hilfreich oder nachteilig für das Projekt ist.

Durch das Kombinieren dieser Dimensionen wird die Basis einer SWOT-Analyse gebildet. In einer SWOT-Matrix werden die Kombinationen der Dimensionswerte aufgefasst.

Tabelle 5.10.: SWOT-Matrix

	Hilfreich	Schädlich
Intern	Stärken	Schwächen
Extern	Chancen	Risiken

Aufgrund dieser Matrix können Faktoren als Schwächen und Risiken identifiziert werden, auf welche das Risikomanagement das Hauptaugenmerk legt.

5.7.2. Risikoklassifikation

Bei der Risikoklassifikation werden Faktoren basierend auf den Bereich, den sie betreffen, mit einer Klasse versehen. Dies dient der Organisation, Zusammenfassung und der Berichterstattung an das Management und die Stakeholder. Für die Projektgruppe sind die folgenden Risikoklassen relevant:

Tabelle 5.11.: Risikoklassen

Risikoklasse	Beschreibung
Zeitplan	Betrifft das Risiko den Zeitplan des Projektes?
Ressourcen	Betrifft das Risiko die Projektressourcen?
Sicherheit	Betrifft das Risiko die Sicherheit von Personen?
Projektumfang	Betrifft das Risiko den Umfang des Projektes?
Privatsphäre	Betrifft das Risiko die Privatsphäre von Nutzern?
Rechtliches	Betrifft das Risiko rechtliche Aspekte?

5.7.3. Risikoquantifizierung

Die Risikoquantifizierung befasst sich damit, den klassifizierten Risiken eine Priorität zuzuordnen. Das Ergebnis ist eine sogenannte Erstellen der Risikomatrix, in welcher die Risiken basierend auf ihren Eintrittswahrscheinlichkeiten und ihren Auswirkungen dargestellt sind.

Eintrittswahrscheinlichkeiten und Auswirkungen Für die Eintrittswahrscheinlichkeiten wird statt der kontinuierlichen Skala in Prozent eine diskretisierte Version mit fünf Bereichen verwendet.

Tabelle 5.12.: Diskretisierte Skala für Eintrittswahrscheinlichkeiten von Risiken

Stufe	Wahrscheinlichkeit	Beschreibung
5	91 – 100%	Tritt sehr wahrscheinlich auf
4	61 – 90%	Tritt wahrscheinlich auf
3	41 – 60%	Tritt in der Hälfte der Fälle auf
2	11 – 40%	Tritt wahrscheinlich nicht auf
1	0 – 10%	Tritt sehr wahrscheinlich nicht auf

Durch diese diskretisierte Skala wird die Diskussion von Wahrscheinlichkeiten einfacher. Die gleiche Skala wird für die Auswirkungen der Risiken verwendet.

5. Projektmanagement

Tabelle 5.13.: Skala der Auswirkungen von Risiken

Stufe	Größe der Auswirkungen	Beschreibung
5	Extrem hoch	Resultiert im Projektfehlschlag
4	Hoch	Erhebliche Verzögerungen, Qualität sinkt stark unter das Ziel
3	Moderat	Bemerkbare Verzögerungen, Sinken der Qualität auf untere Grenze
2	Gering	Geringe Beeinträchtigung des Projekts im akzeptablen Bereich
1	Minimal	Sehr wenig oder keinen Einfluss auf das Projekt

Da beide Skalen den gleichen Wertebereich besitzen, wird die Behandlung der Werte vereinfacht, was zum gemeinsamen Verständnis der Werte für die Gruppe beiträgt.

Zu einem gegebenen Faktor werden durch einen sogenannten Subject Matter Expert (SME) die Eintrittswahrscheinlichkeit sowie die Größe der Auswirkungen aufgrund von Erfahrung bzw. Daten aus der Literatur festgelegt. Hier gilt es zu beachten, dass eine Fehleinschätzung der Wahrscheinlichkeit oder des Ausmaßes eines Faktors zu erheblichen Problemen führen kann, weshalb die Zuweisung dieser Werte nicht durch andere Gruppenmitglieder als die Experten geschehen darf.

Erstellen der Risikomatrix Basierend auf diesen Skalen, welche jeweils als Vektor aufgefasst werden können, ist es möglich, eine Erstellen der Risikomatrix aufzustellen. Diese hat die in Tabelle 5.14 dargestellte Form.

Tabelle 5.14.: Risikomatrix						
Wahrscheinlichkeit p	Auswirkungen ι					
	5	5	10	15	20	25
	5	4	8	12	16	20
	4	3	6	9	12	15
	3	2	4	6	8	10
	2	1	2	3	4	5
	1	1	2	3	4	5

Für die Berechnung der Werte in den Kacheln der Matrix M , den sogenannten Risikowerten, wird die Formel

$$m_{ij} = p_i \cdot \iota_j$$

benutzt, wobei m_{ij} der Eintrag der Matrix an der Stelle ij , p_i die Eintrittswahrscheinlichkeit und ι_j die Auswirkungen sind.

Sämtliche gesammelte Faktoren werden anschließend von ihren jeweiligen SMEs einer geschätzten Eintrittswahrscheinlichkeit sowie einer geschätzten Größe ihrer Auswirkungen zugeteilt. Basierend auf der Risikomatrix wird ihnen daraufhin ein Risikowert zugeordnet, welcher schlussendlich im Planungsschritt des Vorgehens weiter verwendet wird.

5.7.4. Risikoplanung

Nachdem die Risiken identifiziert wurden, die den Erfolg des Projekts gefährden können, muss die Risikoplanung durchgeführt werden.

Risikobewertung

Jede der vier Risikokategorien stellt eine Risikoschwelle dar und dient als Auslöser, das Team, die Betreuer und andere Projektpartner zu mobilisieren und Maßnahmen zur Risikominimierung oder -eindämmung zu ergreifen. In der Tabelle 5.15 Risikobewertung werden die vier Risikokategorien, die Michael Lant in seinem Artikel (siehe [55]) vorgestellt hat, und die zu treffenden Maßnahmen für das Projekt beschrieben.

Tabelle 5.15.: Risikobewertung

25	Kritisch <ul style="list-style-type: none"> • Erfordert dringende Maßnahmen. • Muss täglich überwacht werden. • Erfordert die Benachrichtigung des verantwortlichen Ausführenden. • Erfordert eine sofortige Benachrichtigung der Betreuer und der Projektpartner von der IAV GmbH
15 - 20	Ernsthaft <ul style="list-style-type: none"> • Muss wöchentlich überwacht und überprüft werden. • Erfordert die Benachrichtigung der Projektleitung.
6 - 12	Moderat <ul style="list-style-type: none"> • Muss monatlich überwacht werden. • Muss mindestens einmal im Monat überprüft werden. • Erfordert die Benachrichtigung der Projektleitung.
1 - 5	Minimal <ul style="list-style-type: none"> • Sollte monatlich überprüft werden. • Keine explizite Aktion erforderlich. • Geringe oder keine Auswirkungen auf einen Aspekt des Projekts.

5. Projektmanagement

Risikoregister

Um Risiken in dem Projekt zu verfolgen und zu verwalten, wird ein Risikoregister verwendet. In jeder Sprint-Planungssitzung soll die Risikobewertung durchgeführt werden, dazu wird für jeden Sprint eine neue Tabelle erstellt. Dadurch lässt sich verfolgen, wie sich Risiken im Laufe des Projekts entwickeln und welche Risiken hinzugekommen sind oder entfernt wurden.

Im Laufe des Projekts sollten immer weniger neue Risiken identifiziert werden und der Schweregrad der bekannten Risiken sollte immer geringer werden. Wenn dies nicht geschieht und in den späten Phasen des Projekts noch ein ernsthaftes oder kritisches Risiko vorhanden ist, wurden die Risiken nicht angemessen gemanagt und Projekterfolg kann nicht garantiert werden. Die bis dahin investierte Zeit und der Aufwand laufen Gefahr, verloren zu gehen.

5.7.5. Strategien zur Risikominderung

In diesem Abschnitt werden Maßnahmen zur Umsetzung der Strategien zur Risikominderung definiert. Sehr wichtig ist, dass das Risikomanagement regelmäßig, vollständig und volumnfänglich durchgeführt wird.

Frühes Scheitern

Frühes Scheitern bedeutet, so früh wie möglich im Prozess herauszufinden, ob das, was die Projektgruppe gerade macht, erfolgreich sein wird oder nicht. Diese Ergebnisse sind im Wesentlichen die Grundlage für das Projekt. Wenn ein Erfolg nicht möglich ist, muss sofort mit den Projektverantwortlichen über eine Neuausrichtung des Projekts gesprochen werden.

Zuerst sollten schwierige und riskante Aufgaben erledigt werden, um ein Scheitern in späteren Verlauf des Projekts zu vermeiden. Dadurch lassen sich leichter die Grenzen von *Ophelia* definieren und Erwartungen darüber setzen, was möglich/realistisch und was unmöglich/unrealistisch ist. Es können dadurch unrealistische Erfolgskriterien und die Notwendigkeit zur Änderung der Definition des Projekterfolgs entdeckt werden.

In diesem Fall kann das Projekt noch weitergeführt werden, aber unter einer überarbeiteten und möglicherweise realistischeren Reihe von Erwartungen der Interessengruppen. Es kann auch Verpflichtungen, wie ein größeres Budget oder den Zugang zu Schlüsselpersonen fördern.

Wiederholen

Dieses Verfahren ist leicht und schnell durchführbar. Die frühzeitige Identifizierung von Risiken und die Umsetzung geeigneter Strategien zur Risikominderung im Projekt ist für den Erfolg des Projekts von entscheidender Bedeutung. Dies ist ein kontinuierlicher Kreislauf von Bewertungen und Aktionen, um Risiken ständig zu identifizieren, zu managen und zu minimieren.

Der Risikoplan sollte mindestens einmal pro Sprint überprüft werden. In den Sprint-Planungssitzungen kann das Team gemeinsam über Risiken im Projekt diskutieren. Es muss nicht jedes Mal eine ausführliche Überprüfung durchgeführt werden. Es sollte besonders auf die Risiken, die sich in dem Risikoregister befinden, geachtet werden.

Neue Risiken, die auftreten können, wenn das Team das Projekt durchläuft und mehr über die Herausforderungen erfährt, verdienen besondere Aufmerksamkeit. Wenn hohe Risiken entdeckt werden, sollte sich die Projektgruppe frühzeitig damit befassen.

5.8. Tooldokumentation

Die in diesem Projekt eingesetzten Tools werden in den folgenden Unterkapiteln näher beschrieben. Hierfür werden die Tools in sechs Abschnitte unterteilt. In dem Abschnitt 5.8.1 Projektmanagement Tools werden die verwendeten Projektmanagement Tools beschreiben und in dem Abschnitt 5.8.2 Anforderungsmanagement Tools werden Tools zum Anforderungsmanagement aufgeführt. Tools zum Erstellen von Grafiken werden im Abschnitt 5.8.3 Tools zur Grafikerstellung beschrieben. Im Abschnitt 5.8.4 Tools zur Softwareentwicklung werden Tools zur Softwareentwicklung aufgeführt. Tools zur Erstellung von Dokumenten und Präsentation, sowie zur Literaturverwaltung werden im Abschnitt 5.8.5 Tools für Dokumente, Präsentationen & Literatur beschrieben. Alle anderen Tools werden im Abschnitt 5.8.6 Sonstige Tools zusammengefasst. Die Tools dienen der Projektgruppe als Unterstützung in der Organisation des Projektes.

5.8.1. Projektmanagement Tools

Das Projektmanagement dient der Organisation und Strukturierung des Entwicklungsprozesses. Um dies zu gewährleisten werden verschiedene Tools verwendet, welche unter anderem zur Unterstützung der Sprintplanung oder dem Management von Aufgaben dienen. In den folgenden Abschnitten werden die einzelnen Tools und die Gründe für die Wahl dieser erläutert.

5. Projektmanagement

Redmine

Redmine ist eine flexible, plattform- und datenbankunabhängige Open Source Projektmanagement-Webanwendung und wird der Projektgruppe vom OFFIS zur Verfügung gestellt. Es beinhaltet unter anderem eine Ticketverwaltung, ein Wiki sowie eine Verwaltung für Dokumente [56]. Redmine ist das offizielle Tool zur Projektverwaltung mit Kommunikation zur IAV. Alle Tickets, welche die IAV betreffen, werden über das Redmine verwaltet. Ebenso befindet sich dort das Projektrepository.

Für das Redmine können verschiedene Systeme zur Versionsverwaltung genutzt werden. Im Wiki werden generelle Informationen über die Projektgruppe bereitgestellt. Es ist notwendig, dass Tickets, welche der Geheimhaltungserklärung unterliegen, nur im Redmine zu finden sind, da diese Informationen so nicht nach außen geliefert werden.

Ein Nachteil von Redmine ist, dass die Ticketverwaltung bei zunehmender Anzahl an Aufgaben unübersichtlich wird. Aus diesem Grund wird zur Verwaltung von Tickets, welche nicht der Geheimhaltung unterliegen, ein weiteres Tool verwendet.

Jira

Jira ist ein von Atlassian entwickeltes Projektmanagement-Tool und dient hauptsächlich zur Erstellung und Verwaltung von Tickets. Es wurde auf die agile Softwareentwicklung ausgerichtet und unterstützt Entwickler bei der Verwendung von Scrum und Kanban durch entsprechende Boards [57].

In der Projektgruppe wurde entschieden Jira für das Management des Entwicklungs- und Dokumentationsprozesses zu verwenden, da die Software übersichtlich und besser auf agile Prozesse ausgerichtet ist. Ein weiterer Grund für die Wahl dieses Tools sind mit einhergehenden Funktionalitäten wie beispielsweise die Möglichkeit Pull-Requests zu verwenden. Die Projektgruppe benutzt Jira für alle Aufgaben, welche nicht der Geheimhaltungserklärung unterliegen.

Confluence

Confluence ist eine kommerzielle, von Atlassian entwickelte, Software zum gemeinsamen er- und bearbeiten der Projektplanung, Besprechungsnotizen, Produktanforderungen, des Marketingplans und von Blogposts [58]. Über die Lizenz für Jira hat die Projektgruppe Zugang zu Confluence.

Aktuell verwendet die Projektgruppe den Kalender in Confluence zur Organisation von gemeinsamen Terminen und der Urlaubsplanung.

5.8.2. Anforderungsmanagement Tools

Open Source Requirements Management Tool (OSRMT) ist ein Tool zur Verwaltung von Anforderungen [59]. Es dient dazu Anforderungen übersichtlich zu erstellen und Rückverfolgbarkeit zu haben. Das Tool verwendet einen JBoss Server, der auf dem Server-PC im Projektgruppenraum aufgesetzt wurde. Die Verbindung wird über http aufgebaut.

Die Projektgruppe hat ursprünglich geplant das Tool zu verwenden, da es plattformunabhängig, Open Source und damit kostenlos ist. Dadurch kann das Tool von der Projektgruppe flexibel angepasst werden. Eine Alternative zu OSRMT ist rmToo - Requirements Management Tool, da dieses jedoch nur auf Linux-Systemen ausführbar ist wurde sich dagegen entschieden.

Nach ausführlichen Diskussionen hat die Projektgruppe die Entscheidung getroffen auf das Tool zu verzichten. Der Grund ist, dass das Tool in der aktuellen Version keine Spalten in den Anforderungstabellen hinzufügen oder löschen kann. Dieser Fehler ist bekannt und zum jetzigen Zeitpunkt offen. Der damit verbundene Aufwand und der zeitliche Engpass im Projektplan haben dazu geführt, dass auf die Verwendung dieses Tools verzichtet wird. Stattdessen findet die Anforderungsanalyse in Latex statt. Der Vorteil ist hierbei, dass kein zusätzlicher Parser für die Überführung der Anforderungen in Latex geschrieben werden muss.

5.8.3. Tools zur Grafikerstellung

Für die Erstellung von Diagrammen und Grafiken werden die Tools Draw IO, UMLet und Inkscape verwendet.

Draw IO

Draw IO ist ein web-basiertes Grafiktool, das dazu dient, Diagramme auf einfache Weise zu zeichnen. Das Tool wurde gewählt, weil es plattformunabhängig und intuitiv zu bedienen ist. Erstellte Diagramme können kostenlos in verschiedenen Formaten (JPEG, PNG, SVG und PDF) exportiert werden.

Die Projektgruppe verwendet das Tool zum Erstellen von Diagrammen, da es die Möglichkeit bietet unterschiedliche Diagrammtypen abzubilden. Das Tool kann kostenpflichtig über den Atlassian Marketplace in Jira und Confluence integriert werden, steht aber auch kostenlos auf der Herstellerwebseite zur Verfügung. Die erstellten Zeichnungen können direkt heruntergeladen werden [60].

5. Projektmanagement

UMLet

UML-Diagramme nehmen eine wichtige Rolle für die Kommunikation und das Verständnis unter den Gruppenmitgliedern und mit den Betreuern ein. Sie sind für die Systemarchitektur ebenso essentiell.

Das Tool UMLet ist auf das Erstellen von UML-Diagrammen ausgelegt und bietet die Möglichkeit mithilfe von Symbolen Abhängigkeiten zu verdeutlichen. die einzelnen Symbole können kategorisiert und über eine Syntax angepasst werden. Dieser Umstand spart Zeit bei der Erstellung von Diagrammen. Ein weiterer Vorteil des Tools ist, dass dieses in Java entwickelt wurde, wodurch Plattformunabhängigkeit erreicht wird.

Inkscape

Inkscape ist ein vektorbasiertes Tool mit dem SVG-Grafikdateien erstellt werden können [61]. In der Projektgruppe wurde es zum Erstellen des Logos verwendet. Inkscape nutzt Grafiktools für geometrische Figuren, die zu Figuren beliebiger Komplexität zusammengefügt werden können. Der Vorteil von Inkscape gegenüber anderen Tools wie Adobe Illustrator oder Affinity Designer ist, dass dieses Open Source und somit kostenlos ist.

5.8.4. Tools zur Softwareentwicklung

Dieser Abschnitt behandelt Tools für den Softwareentwicklungsprozess.

Jenkins

Jenkins ist eine quelloffene Software für Continious Integration. Jenkins charakterisiert sich durch einen Server, welcher verschiedene Aufgaben im Kontext der Softwareentwicklung automatisiert. Unter anderem beinhaltet dies das Erstellen, Testen und Einsetzen von Softwarecode. [62]

Die Projektgruppe nutzt Jenkins, um die kontinuierliche Integration von Komponenten in einem Entwicklungsprozess zu gewährleisten und zu testen. Jenkins kann durch verschiedene Plug-Ins erweitert werden. Das Tool wird auf dem Server-PC ausgeführt.

5.8.5. Tools für Dokumente, Präsentationen & Literatur

In diesem Unterabschnitt werden alle verwendeten Tools zur Erstellung von Dokumenten und Präsentationen, sowie Tools zur Verwaltung von Literatur, vorgestellt.

LaTeX

Die Benutzung des Textsatzsystems TeX wird durch das Softwarepaket L^AT_EX ermöglicht. Die Projektgruppe hat sich dazu entschlossen dieses Tool zur Erstellung von Dokumenten zu verwenden, da es plattformunabhängig ist und verschiedene kostenlose Editoren und Pakete zu Verfügung stehen. Des Weiteren lassen sich tex-Dateien mit einem zusätzlichen Tool versionieren. Die umfassenden Funktionalitäten und die Tatsache, dass es sich um ein Open Source Tool handelt, haben die Projektgruppe veranlasst sich gegen Alternativen wie Microsoft Word, OpenOffice Writer oder LibreOffice Writer zu entscheiden.

Microsoft PowerPoint

Die Projektgruppe hat sich für die Erstellung von Präsentationen für Microsoft PowerPoint entschlossen. Der Grund hierfür liegt in der Vorerfahrung von jedem einzelnen mit diesem Tool. Alternativ hätte die Projektgruppe L^AT_EX, OpenOffice Impress oder LibreOffice Impress verwenden können. Die Lizenz kann von jedem Projektmitglied über die Universität erworben werden (siehe Microsoft Office 365 ProPlus).

Mendeley

Mendeley ist eine Referenzmanagement-Software und ein Forschernetzwerk. Das Tool kann basierend auf den hinterlegten Literaturquellen automatisch BibTeX Einträge in einer bib-Datei für L^AT_EX generieren und diese mit dem Mendeley-Server synchronisieren. Das Tool wird in der Projektgruppe verwendet, da es die Zusammenarbeit bei der Literaturrecherche und der Verwaltung von Literaturquellen erleichtert. Es können mit Hilfe des Browser Plug-ins neue Literatureinträge erstellt werden [63]. Alternativen zu Mendeley sind Anwendungen wie EndNote und Qiqqa.

5.8.6. Sonstige Tools

Alle Tools, die bisher keiner Kategorie zugewiesen wurden, werden hier zusammengefasst.

Git

Git ist ein kostenloses und quelloffenes verteiltes Versionskontrollsystem, das entwickelt wurde, um sowohl kleine als auch sehr große Projekte schnell und effizient zu verwalten. Weiterhin ermöglicht es das gleichzeitige Arbeiten von verschiedenen Personen an demselben Dokument [64]. Da fast alle Projektgruppenteilnehmer

5. Projektmanagement

schon vor Projektanfang Erfahrung mit Git hatten, hat sich die Projektgruppe dazu entschlossen dieses Tool zu verwenden. Alternativen zu Git sind Subversion oder Mercury.

6. Anforderungserhebung

Die Anforderungserhebung befasst sich mit Prozessen, die als Grundlage für Anforderungen dienen sollen. Dabei muss man zunächst den Rahmen des Projekts festlegen. Hierbei werden am Projekt beteiligte Personengruppen und dessen Interessen am Projekt erfasst. Dies geschieht im Abschnitt 6.1 Stakeholderanalyse, wo zu den beteiligten Personengruppen die Erwartungen des Projekts, Ziele bzw. Interessen und Einfluss auf das Projekt dokumentiert werden.

Um nun die Bedürfnisse dieser Personengruppen in einfacher Form festzuhalten, wurden User Stories erstellt, da es zunächst schwierig ist aus den Erwartungen der Stakeholder direkt konkrete Anforderungen abzuleiten. Die User Stories im Abschnitt 6.2 User Stories Katalog sind zur besseren Übersicht kategorisiert.

Durch Use Cases sollen die Funktionalitäten von *Ophelia* konkretisiert werden, indem hier die Unterscheidung von statischen und dynamischen Einstellungen (statisch, dynamisch) erfolgt. Die Use Cases sind im Abschnitt 6.3 Use Case Katalog hinterlegt.

Die Stakeholderanalyse, User Stories und Use Cases bilden die Grundlage für Anforderungen. Im Abschnitt 6.4 Anforderungskatalog sind die Anforderungen aufgeteilt nach Top-Level Anforderungen, Anforderungen an das Ophelia-Modul (Ophelia-Modul), Anforderungen an die Ophelia-App und den Ophelia-Server.

6.1. Stakeholderanalyse

Die Stakeholderanalyse ist eine Methode, mit welcher die Stakeholder und deren Interessen identifiziert werden. Sie resultiert in einer Liste, welche Informationen über die einzelnen Stakeholder aufführt. Dazu gehört die Rolle im Projekt, Erwartungen, Involvierung im Projekt und Interesse am Projekt.

Im Weiteren werden die einzelnen Stakeholdergruppen definiert:

6. Anforderungserhebung

Tabelle 6.1.: Stakeholder des Projektes

Stakeholder	Erwartungen an das Ziel/Interessen Projekt	Einfluss, Macht und Einstellung zum Projekt
IAV als Auftraggeber	<p>Der Auftraggeber erwartet einem von der PG eigenständig entwickeltes Komfortsystem. Die PG ist selbstständig verantwortlich für die Wahl der zu implementierenden Funktionalitäten, da von den Auftraggebern diesbezüglich keine Forderungen gestellt werden. Der Rahmen des Projektes ist durch das Thema Configuration & Helping, Independent Learning Limousine vorgegeben und wird damit begrenzt.</p>	<p>Der Auftraggeber zielt auf ein Komfortsystem ab, welches sich seriennäßig in Fahrzeuge einbauen lässt und erweiterbar ist. Daher sind sie an einer vollumfänglichen Dokumentation interessiert.</p>
Professoren	<p>Im Rahmen der Lehrveranstaltung wird ein lauffähiger Prototyp und ein entsprechender Lerneffekt der Entwickler erwartet. Des Weiteren erwarten die Professoren motivierte und engagierte Entwickler.</p>	<p>Auf Wirtschaftsebene sind die Ziele der Professoren, ein vorzeigbares und repräsentatives Projekt in Kooperation mit einem Industriepartner durchzuführen.</p>

6.1. Stakeholderanalyse

Betreuer	Die Erwartungen der Betreuer sind denen der Professoren sehr ähnlich, da sie im Auftrag der Professoren handeln. Die Erwartungen weisen Ähnlichkeiten zu denen der IAV als Auftraggeber auf, da auch sie eine vollumfängliche Dokumentation erwarten.	Das Ziel der Betreuer ist ein qualitativ hochwertiges Endprodukt sowie der Lerneffekt, der für die Entwickler entsteht, indem diese einen Industrieprozess leben.	Aufgrund der unmittelbaren Beaufsichtigung des Entwicklungsprozesses ist der Einfluss der Betreuer, neben dem der Entwickler, am größten. Das liegt einerseits an der Erfahrung durch vorangegangene Projekte und andererseits daran, dass die Betreuer die direkte Schnittstelle zwischen Professoren und den Entwicklern darstellen.
IAV als Entwickler	Die IAV als Entwickler erwartet eine vollumfängliche Dokumentation des Entwicklungsprozesses und eine gut dokumentierte und erweiterbare Programmiercodebasis.	Das Ziel der IAV ist es, den Entwicklern bei der Arbeit am Versuchsträger zu unterstützen.	Um die genannten Erwartungen und Ziele zu erfüllen, besitzt die IAV vollständigen Zugang zum Projekt. Die Arbeit der IAV geht der Arbeit der Entwickler voraus und kann diese bei Verzögerungen blockieren.

6. Anforderungserhebung

Entwickler	Die Entwickler erwarten zum Ende des Projektes einen auf das Projekt zugeschnittenen Entwicklungsprozess, aus welchem ein erweiterbares und wartbares Komfortsystem hervorgeht.	Das Ziel der Entwickler ist es, die selbst definierten Anforderungen und durch die Auftraggeber vorgegebenen Aufgaben soweit möglich zu erfüllen. Dieses Ziel umfasst eigene Grundlagen, wie beispielsweise Qualitätsstandards und gemeinsam getroffene Entscheidungen.	Die Entwickler haben aufgrund der Freiheit in der Auswahl der Funktionalitäten den größten Einfluss auf das Endprodukt und somit auch die entsprechende Verantwortung für den Entwicklungsprozess. Die Einstellung zum Gesamtprojekt ist positiv, da die Entwickler sich aktiv für das Projekt entschieden haben und damit eine intrinsische Motivationsgrundlage geschaffen haben.
Fahrer	Der Fahrer erwartet bei Fahrtantritt ein vorkonfiguriertes System, welches an seine individuellen Bedürfnisse angepasst ist.	Das Ziel für den Fahrer ist es, Einstellungen vor und während der Fahrt nicht mehr selber tätigen zu müssen.	Der Fahrer entscheidet aus subjektiver Perspektive inwiefern die Funktionalitäten zu seinem persönlichen Komfort beitragen. Sein Feedback ist ausschlaggebend für Änderungen und Anpassungen im Gesamtsystem.
Mitfahrer	Der Mitfahrer erwartet, ähnlich wie der Fahrer, ein vorkonfiguriertes System, welches an seine Bedürfnisse angepasst ist.	Das Ziel des Mitfahrers ist es, eine Umgebung zu haben, die trotz der Privilegien des Fahrers ein optimales Komforterlebnis schafft.	Da der Mitfahrer ebenfalls zu den Nutzern des Komfortsystems zählt, ist sein persönliches Empfinden des Komfortsystems fast genauso wichtig, wie das des Fahrers. Der Grund dafür ist, dass viele Funktionalitäten des Fahrers und Mitfahrers identisch sind.

Privater Fahrzeughalter	Er erwartet, dass seine persönlichen Einstellungen gespeichert werden und bei jeder Fahrt abrufbar sind. Dies soll auch dann gewährleistet sein, wenn jemand anderes das Fahrzeug benutzt.	Das Ziel des privaten Fahrzeughalters deckt sich mit denen des Fahrers mit dem Zusatz, dass das Komforterlebnis erhalten bleibt, unabhängig davon, wer das Fahrzeug benutzt.	Der Einfluss des privaten Fahrzeughalters ist, abgesehen von dem Aspekt der Profileinstellungen, verhältnismäßig gering, da er nicht zu den aktiven Nutzern des Komfortsystems zählt, solange er nicht auch Fahrer ist.
Gewerblicher Fahrzeughalter	Er erwartet, dass jeder Nutzer des Fahrzeugs seine persönlichen Einstellungen speichern und abrufen kann. Dies soll unabhängig davon sein, wer das Fahrzeug wie häufig benutzt.	Die Profileinstellungen sind für ihn das zentrale Element des zu entwickelnden Systems, da diese sicherstellen, dass jeder Nutzer des Fahrzeugs seine individuellen Komfortbedürfnisse bei jeder Fahrt zur Verfügung hat.	Der Einfluss des gewerblichen Fahrzeughalters ist größer, als der des privaten Fahrzeughalters. Ursache dafür ist, dass er der entscheidende Stakeholder für die Einsetzbarkeit der Profile ist. Erst wenn seine Erwartungen erfüllt sind, können die Profileinstellungen als nutzbar angesehen werden.

6.2. User Stories Katalog

Umfang User Stories enthalten ausschließlich Kernaspekte von *Ophelia*, die als Muss-Anforderungen im System aufgefasst werden.

- Musik
- Navigation
- Klimatisierung
- Fenster
- Rollos
- Sitzprofil (Sitz-, Spiegel-, Lenkradposition)
- Innenraumbeleuchtung
- KI

6. Anforderungserhebung

- Nutzereingaben
- App
- Priorisierung
- Datenverwaltung
- Systementwicklung
- Szenarien

Ophelia

6.2.1. Allgemein

User Story Allgemein 1: **Als Nutzer möchte ich**, dass *Ophelia* meine persönlichen Präferenzen lernt, **damit** es der aktuellen Situation entsprechend automatisiert Einstellungen vornimmt.

User Story Allgemein 2: **Als Nutzer möchte ich**, dass *Ophelia* mich als Nutzer erkennt, **um** Einstellungen nach meinen persönlichen Präferenzen vornehmen zu können.

6.2.2. Musik

User Story Musik 1: **Als Nutzer möchte ich**, dass *Ophelia* die Wiedergabequellen meiner Musik priorisiert, **damit** *Ophelia* eine Wiedergabequelle passend zu der aktuellen Situation auswählt.

User Story Musik 2: **Als Nutzer möchte ich**, dass *Ophelia* meine Standardmusiklautstärke speichert, **damit** *Ophelia* meine Standardmusiklautstärke vor Fahrtantritt einstellt.

User Story Musik 3: **Als Nutzer möchte ich**, dass *Ophelia* meine Standardequalizereinstellungen speichert, **damit** *Ophelia* meine Standardequalizereinstellungen vor Fahrtantritt einstellt.

User Story Musik 4: **Als Nutzer möchte ich**, dass *Ophelia* die Musiklautstärke entsprechend meiner persönlichen Präferenzen passend zu der aktuellen Situation einstellt, **damit** ich die Einstellung nicht selbst vornehmen muss.

User Story Musik 5: **Als Nutzer möchte ich**, dass *Ophelia* vor Fahrtantritt meine Standardmusiklautstärke einstellt, **um** dies nicht selbst machen zu müssen.

User Story Musik 6: **Als Nutzer möchte ich**, dass *Ophelia* vor Fahrtantritt meine Standardequalizereinstellungen einstellt, **um** dies nicht selbst machen zu müssen.

User Story Musik 7: **Als Nutzer möchte ich**, dass *Ophelia* meinen Radiosender

entsprechend meiner persönlichen Präferenzen passend zu der aktuellen Situation einstellt, **damit** ich die Einstellung nicht selbst vornehmen muss.

User Story Musik 8: **Als** Nutzer **möchte ich**, dass *Ophelia* den Equalizer der Musikanlage entsprechend meiner persönlichen Präferenzen passend zu der aktuellen Situation einstellt, **damit** ich die Einstellung nicht selbst vornehmen muss.

User Story Musik 9: **Als** Nutzer **möchte ich**, dass *Ophelia* eine situationsabhängige Wiedergabeliste beim Einschalten der Musikanlage passend zu der aktuellen Situation fortsetzt, **damit** ich die Wiedergabeliste nicht selbst auswählen muss.

User Story Musik 10: **Als** Nutzer **möchte ich** unterschiedliche Musikdienste einbinden können, **um** meine Musik über diese abspielen zu können.

6.2.3. Navigation

User Story Navigation 1: **Als** Nutzer **möchte ich** meine Standardnavigationslautstärke speichern können, **um** diese nicht vor jeder Fahrt neu einstellen zu müssen.

User Story Navigation 2: **Als** Nutzer **möchte ich**, dass *Ophelia* vor Fahrtantritt meine Standardnavigationslautstärke einstellt, **um** dies nicht selbst machen zu müssen.

User Story Navigation 3: **Als** Nutzer **möchte ich**, dass *Ophelia* entsprechend meiner persönlichen Präferenzen passend zur aktuellen Situation die Navigationslautstärke einstellt, **um** dies nicht selbst machen zu müssen.

User Story Navigation 4: **Als** Nutzer **möchte ich**, dass *Ophelia* mich bei Fahrtantritt an häufig verwendete Zwischenziele meiner Routen erinnert, **um** die Fahrt besser planen zu können.

User Story Navigation 5: **Als** Nutzer **möchte ich** *Ophelia* ein Tagesprogramm angeben, **damit** mehrere Ziele optimiert angefahren werden können.

User Story Navigation 6: **Als** Nutzer **möchte ich** *Ophelia* Abfahrtszeiten für das Tagesprogramm angeben können, **damit** es mich an diese erinnern kann.

User Story Navigation 7: **Als** Nutzer **möchte ich**, dass *Ophelia* meine Route optimiert, **um** ökonomisch fahren zu können.

User Story Navigation 8: **Als** Nutzer **möchte ich** die Möglichkeit haben bezüglich eines Ziels Gegenstände anzugeben, **damit** das Fahrzeug mich an diese erinnern kann.

User Story Navigation 9: **Als** Fahrer **möchte ich** dass das Fahrzeug meine Abfahrtszeiten erlernt, **damit** ich vom Fahrzeug Vorschläge bezüglich von Zielen erhalten

6. Anforderungserhebung

kann.

User Story Navigation 10: **Als** Fahrer **möchte ich** dass das Fahrzeug meine Fahrtziele erlernt , **damit** ich vom Fahrzeug Vorschläge bezüglich von Zielen erhalten kann.

User Story Navigation 11: **Als** Fahrer **möchte ich**, dass das Fahrzeug mich an für die Fahrt relevante Zwischenziele erinnert, **um** die Fahrt besser überblicken zu können.

6.2.4. Nutzereingaben

User Story Benutzereingaben 1: **Als** Nutzer **möchte ich** dem Fahrzeug ohne Touch-Eingaben Befehle erteilen können, **um** beim Fahren nicht den Blick von der Straße abzuwenden.

User Story Benutzereingaben 2: **Als** Nutzer **möchte ich** dem Fahrzeug ohne das Drücken von Tasten oder Knöpfen Befehle erteilen können, **um** beim Fahren nicht den Blick von der Straße abzuwenden.

User Story Benutzereingaben 3: **Als** Fahrer **möchte ich** die Möglichkeit haben die Befehlsfunktion zu aktivieren, **um** diese bei Bedarf nutzen zu können.

User Story Benutzereingaben 4: **Als** Fahrer **möchte ich** die Möglichkeit haben die Befehlsfunktion zu deaktivieren, **um** diese bei Bedarf ausschalten zu können.

6.2.5. Rollos

User Story Rollo 1: **Als** Mitfahrer **möchte ich** die Standardpositionen der Rollos speichern können, **damit** *Ophelia* diese vor Fahrtantritt einstellt.

User Story Rollo 2: **Als** Mitfahrer **möchte ich**, dass *Ophelia* vor Fahrtantritt meine Standardpositionen der Rollos einstellt, **um** dies nicht selbst machen zu müssen.

User Story Rollo 3: **Als** Mitfahrer **möchte ich**, dass *Ophelia* die Rollopositionen entsprechend meiner persönlichen Präferenzen passend zu der aktuellen Situation einstellt, **damit** ich die Einstellung nicht selbst vornehmen muss.

User Story Rollo 4: **Als** Mitfahrer **möchte ich** die Position der Rollos schon vor Fahrtantritt einstellen können, **um** nach dem Einstieg in das Auto nicht geblendet zu werden.

6.2.6. Sitzprofil

User Story Sitzprofil 1: **Als Nutzer möchte ich**, dass *Ophelia* die Sitzposition meines Sitzes im Profil speichert, **damit** *Ophelia* diese vor Fahrtantritt einstellt.

User Story Sitzprofil 2: **Als Fahrer möchte ich**, dass *Ophelia* die Lenkradposition im Profil speichert, **damit** *Ophelia* diese vor Fahrtantritt einstellt.

User Story Sitzprofil 3: **Als Fahrer möchte ich**, dass *Ophelia* die Position des Rückspiegels im Profil speichert, **damit** *Ophelia* diese vor Fahrtantritt einstellt.

User Story Sitzprofil 4: **Als Fahrer möchte ich**, dass *Ophelia* die Position der Außenspiegel im Profil speichert, **damit** *Ophelia* diese vor Fahrtantritt einstellt.

6.2.7. Klimatisierung

User Story Klimatisierung 1: **Als Nutzer möchte ich** die Standardinnenraumtemperatur speichern können, **um** diese nicht vor jeder Fahrt neu einstellen zu müssen.

User Story Klimatisierung 2: **Als Nutzer möchte ich** meine Standardsitztemperatur speichern können, **um** diese nicht vor jeder Fahrt neu einstellen zu müssen.

User Story Klimatisierung 3: **Als Fahrer möchte ich** meine Standardlenkradttemperatur speichern können, **um** diese nicht vor jeder Fahrt neu einstellen zu müssen.

User Story Klimatisierung 4: **Als Nutzer möchte ich** die Standardgebläsestärke der Klimaanlage speichern können, **um** diese nicht vor jeder Fahrt neu einstellen zu müssen.

User Story Klimatisierung 5: **Als Nutzer möchte ich**, dass *Ophelia* vor Fahrtantritt meine Standardinnenraumtemperatur einstellt, **um** dies nicht selbst machen zu müssen.

User Story Klimatisierung 6: **Als Nutzer möchte ich**, dass *Ophelia* vor Fahrtantritt meine Standardsitztemperatur einstellt, **um** dies nicht selbst machen zu müssen.

User Story Klimatisierung 7: **Als Fahrer möchte ich**, dass *Ophelia* vor Fahrtantritt meine Standardlenkradttemperatur einstellt, **um** dies nicht selbst machen zu müssen.

User Story Klimatisierung 8: **Als Nutzer möchte ich**, dass *Ophelia* vor Fahrtantritt meine Standardgebläsestärke der Klimaanlage einstellt, **um** dies nicht selbst machen zu müssen.

User Story Klimatisierung 9: **Als Nutzer möchte ich**, dass das Fahrzeug beim Betreten nach meinen Bedürfnissen klimatisiert ist, **damit** ich nicht in ein unangenehm

6. Anforderungserhebung

temperiertes Auto steigen muss.

User Story Klimatisierung 10: **Als Nutzer möchte ich**, dass mein Sitz beim Betreten des Fahrzeugs nach meinen Bedürfnissen klimatisiert ist, **damit** der Sitz nicht unangenehm temperiert ist.

User Story Klimatisierung 11: **Als Fahrer möchte ich**, dass mein Lenkrad beim Betreten des Fahrzeugs nach meinen Bedürfnissen klimatisiert ist, **damit** ich kein unangenehm temperiertes Lenkrad habe.

User Story Klimatisierung 12: **Als Nutzer möchte ich**, dass die Stärke des Gebläses beim Betreten des Fahrzeugs nach meinen Bedürfnissen eingestellt ist, **damit** ich dies nicht selbst einstellen muss.

User Story Klimatisierung 13: **Als Nutzer möchte ich**, dass meine Klimazone beim Betreten nach meinen Bedürfnissen klimatisiert ist, **um** meine persönlichen Vorlieben nicht der gesamten Innenraumklimatisierung des Fahrzeugs unterordnen zu müssen.[44]

6.2.8. Innenraumbeleuchtung

User Story Innenraumbeleuchtung 1: **Als Nutzer möchte ich**, dass *Ophelia* meine Standardinnenraumbeleuchtungsstärke speichert, **um** diese nicht vor jeder Fahrt neu einstellen zu müssen.

User Story Innenraumbeleuchtung 2: **Als Nutzer möchte ich**, dass *Ophelia* vor Fahrtantritt meine Standardinnenraumbeleuchtungsstärke einstellt, **um** dies nicht selbst machen zu müssen.

User Story Innenraumbeleuchtung 3: **Als Nutzer möchte ich**, dass *Ophelia* die Innenraumbeleuchtungsstärke entsprechend meiner persönlichen Präferenzen passend zur aktuellen Situation einstellt, **damit** ich die Einstellung nicht selbst vornehmen muss.

User Story Innenraumbeleuchtung 4: **Als Nutzer möchte ich**, dass *Ophelia* meine Standardinnenraumbeleuchtungsfarbe speichert, **um** diese nicht vor jeder Fahrt neu einstellen zu müssen.

User Story Innenraumbeleuchtung 5: **Als Nutzer möchte ich**, dass *Ophelia* vor Fahrtantritt meine Standardinnenraumbeleuchtungsfarbe einstellt, **um** dies nicht selbst machen zu müssen.

User Story Innenraumbeleuchtung 6: **Als Nutzer möchte ich**, dass *Ophelia* die Innenraumbeleuchtungsfarbe entsprechend meiner persönlichen Präferenzen passend zur aktuellen Situation einstellt, **damit** ich die Einstellung nicht selbst vornehmen muss.

6.2.9. Fenster

User Story Fenster 1: **Als** Nutzer **möchte ich**, dass *Ophelia* meine Standardfensterhöhe speichert, **um** diese nicht vor jeder Fahrt neu einstellen zu müssen.

User Story Fenster 2: **Als** Nutzer **möchte ich**, dass *Ophelia* vor Fahrtantritt meine Standardfensterhöhe einstellt, **um** dies nicht selbst machen zu müssen.

User Story Fenster 3: **Als** Nutzer **möchte ich**, dass *Ophelia* die Fensterhöhe entsprechend meiner persönlichen Präferenzen passend zu der aktuellen Situation einstellt, **damit** ich die Einstellung nicht selbst vornehmen muss.

User Story Fenster 4: **Als** Nutzer **möchte ich**, dass *Ophelia* die Fensterhöhe passend zum Klima einstellt, **um** das Fahrzeug passend zu klimatisieren.

6.2.10. App

User Story App 1: **Als** Nutzer **möchte ich** ein Nutzerprofil anlegen können, **um** meine Einstellungen speichern zu können.

User Story App 2: **Als** Nutzer **möchte ich** mich in der *Ophelia*-App anmelden können, **um** Zugriff auf mein Nutzerprofil zu bekommen.

User Story App 3: **Als** Nutzer **möchte ich** mich von der *Ophelia*-App abmelden können, **um** meine Daten sicher zu halten.

User Story App 4: **Als** Fahrer **möchte ich** über die *Ophelia*-App meine Abfahrtszeit eingeben können, **damit** die entsprechenden Voreinstellungen getätigter werden können.

User Story App 5: **Als** Fahrer **möchte ich** über die *Ophelia*-App an meine Abfahrtszeit erinnert werden, **damit** ich die Abfahrt nicht vergesse.

User Story App 6: **Als** Fahrer **möchte ich** über die *Ophelia*-App mein Tagesprogramm speichern können, **damit** die entsprechenden Voreinstellungen getätigter werden können.

User Story App 7: **Als** Nutzer **möchte ich** über die *Ophelia*-App Webservices einbinden können, **um** diese nutzen zu können.

User Story App 8: **Als** Fahrer **möchte ich** über die *Ophelia*-App Szenarien erstellen können, **um** diese im Fahrzeug nutzen zu können.

User Story App 9: **Als** Fahrer **möchte ich** über die *Ophelia*-App Gegenstände zu Zielen speichern können, **um** damit ich an diese erinnert werde.

User Story App 10: **Als** Fahrer **möchte ich**, dass das Fahrzeug mich frühzeitig über die *Ophelia*-App an Abfahrtzeiten erinnert, **um** nicht zu spät zu Terminen zu

6. Anforderungserhebung

erscheinen.

User Story App 11: **Als Nutzer möchte ich** in der *Ophelia*-App Standardfahrzeugeinstellungen hinterlegen können, **damit** diese vor Fahrtantritt automatisch ausgeführt werden.

User Story App 12: **Als Fahrer möchte ich** alle Einstellungen, die das Fahrzeug von selbst vornimmt, einzeln deaktivieren können, **um** die Kontrolle über alle Einstellungen zu behalten.

User Story App 13: **Als Fahrer möchte ich**, dass meine expliziten Einstellungen Vorrang vor denen von *Ophelia* haben, **um** die Kontrolle über alle Einstellungen zu behalten.

6.2.11. Priorisierung

User Story Rechte 1: **Als Mitfahrer möchte ich** dem Fahrer meine Navigationseinstellungen vorschlagen können, **damit** diese vom Fahrer akzeptiert werden können.

User Story Rechte 2: **Als Mitfahrer möchte ich** dem Fahrer meine Musikeinstellungen vorschlagen können, **damit** diese vom Fahrer akzeptiert werden können.

User Story Rechte 3: **Als Mitfahrer möchte ich** dem Fahrer meine Innenraumbeleuchtungseinstellungen vorschlagen können, **damit** diese vom Fahrer akzeptiert werden können.

User Story Rechte 4: **Als Fahrer möchte ich** gefragt werden, wenn ein anderer Nutzer Änderungen der Einstellungen vornimmt, **um** meine Einstellungen nicht ohne mein Einverständnis ändern zu lassen.

User Story Rechte 5: **Als Fahrer möchte ich** eine Änderungsanfrage für Einstellungen von anderen Personen ablehnen können, **um** meine Einstellungen beizubehalten.

User Story Rechte 6: **Als Fahrer möchte ich** eine Änderungsanfrage für Einstellungen von anderen Personen annehmen können, **um** die Einstellungen zu akzeptieren.

6.2.12. Datenverwaltung

User Story Datenverwaltung 1: **Als** Fahrzeughalter **möchte ich**, dass ich *Ophelia* zurücksetzen kann, **um** beim Verkauf keine Nutzerdaten herauszugeben.

User Story Datenverwaltung 2: **Als** Nutzer **möchte ich** Datenaufzeichnung ausschalten können, **um** Datenkontrolle zu bewahren.

User Story Datenverwaltung 3: **Als** Nutzer **möchte ich** mein Nutzerprofil löschen können, **damit** die Daten gelöscht sind.

User Story Datenverwaltung 4: **Als** Fahrzeughalter **möchte ich** OTA-Updates durchführen können, **um** nicht zur Werkstatt fahren zu müssen.

User Story Datenverwaltung 5: **Als** Fahrzeughalter **möchte ich** Updates vorher bestätigen, **um** mich über die Features vorher zu informieren.

User Story Datenverwaltung 6: **Als** Fahrzeughalter **möchte ich** Updates ablehnen können, **um** unerwünschte Features zu vermeiden.

User Story Datenverwaltung 7: **Als** Fahrzeughalter **möchte ich** entscheiden welche Webservices freigeschaltet sind, **um** Schadsoftware in *Ophelia* zu vermeiden.

User Story Datenverwaltung 8: **Als** Nutzer **möchte ich**, dass meine Daten verschlüsselt übertragen werden, **um** meine Daten zu schützen.

6.2.13. Szenarien

User Story Szenarien 1: **Als** Fahrer **möchte ich**, dass das Fahrzeug an verschiedene Szenarien angepasst werden kann, **um** eine komfortablere Fahrt zu haben.

User Story Szenarien 2: **Als** Fahrer **möchte ich**, dass ich verschiedene Szenarien für das Fahrzeug erstellen kann, **um** eine komfortablere Fahrt zu haben.

User Story Szenarien 3: **Als** Fahrer **möchte ich**, dass ich erstellte Szenarien speichern kann, **um** diese nicht immer wieder erstellen zu müssen.

User Story Szenarien 4: **Als** Fahrer **möchte ich**, dass ich Szenarien bearbeiten kann, **um** sie an meine sich ändernden Bedürfnisse anpassen zu können.

6. Anforderungserhebung

6.2.14. Systementwicklung

User Story Systementwicklung 1: **Als** Entwickler **möchte ich** die Möglichkeit haben Systeminformationen auszulesen, **um** *Ophelia* debuggen zu können.

User Story Systementwicklung 2: **Als** Entwickler **möchte ich** Routinen aufrufen können, die im Normalbetrieb nicht aufgerufen werden, **um** *Ophelia* zu Testzwecken in spezifische Zustände überführen zu können.

User Story Systementwicklung 3: **Als** Entwickler **möchte ich** Busdaten auslesen können, **um** *Ophelia* debuggen zu können.

6.3. Use Case Katalog

Einen Use Case definiert die Projektgruppe als eine Darstellung von Strukturen und dem Verhalten von Software und Systembestandteilen mit entsprechenden Akteuren. Use Cases werden aus User Stories abgeleitet und erweitern diese. Die Projektgruppe beschreibt Use Cases durch UML-Use-Case-Diagramme und erklärende Texte. Use Cases bilden mit der Vision und den User Stories die Grundlage für die Anforderungen.

Umfang Use Cases enthalten ausschließlich Kernaspekte des Systems, die als Muss-Anforderungen im System aufgefasst werden.

6.3.1. App Vorbereitung von Fahrten

Die Abb. 6.1 Use Case App Fahrtvorbereitung beschreibt die Vorbereitung von Fahrten in der App. Sowohl Fahrer als auch Mitfahrer können Fahrten planen. Die *Abfahrtszeiten* und *Ziele anzugeben* gehört zum Anwendungsfall *Vorbereitung von Fahrten*. Optional kann der Nutzer *Tagesprogramme* oder *Gegenstände angeben*.

Der nicht-lernende Agent kann den Nutzer vorab an *Fahrten erinnern*, dazu gehört die *Erinnerung an Abfahrtszeiten* und die *Erinnerung an Gegenstände*, wenn der Nutzer Gegenstände bei der Fahrtvorbereitung angegeben hat.

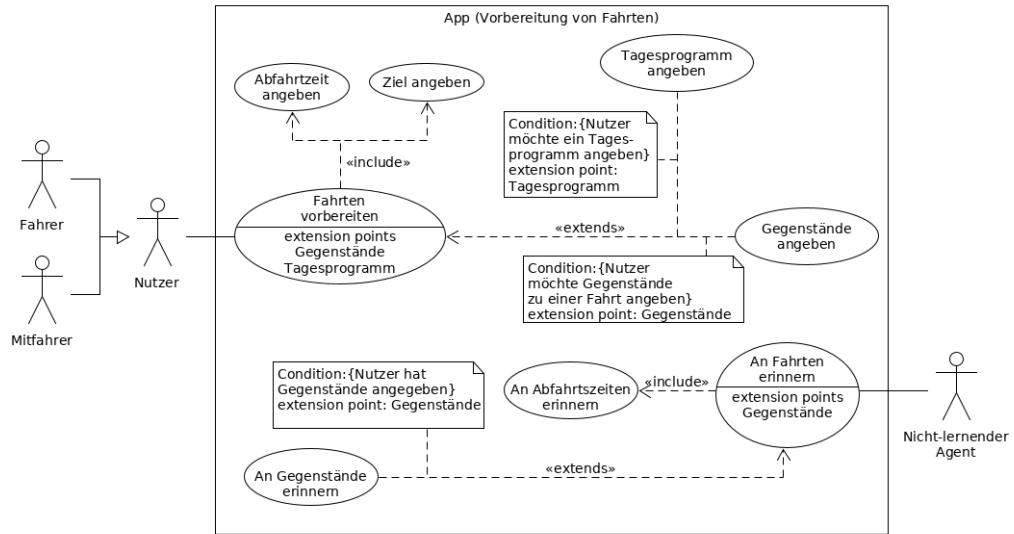


Abbildung 6.1.: Use Case App Fahrtvorbereitung

6.3.2. App Profilverwaltung

Die Abb. 6.2 Use Case App Profilverwaltung beschreibt die Verwaltung von Profilen in der App. Sowohl der Fahrer als auch der Mitfahrer können sich als Nutzer bei der App anmelden, abmelden und ein Profil erstellen oder löschen. Der Fahrzeughalter kann die App, wie die Nutzer, bedienen und zusätzlich die Datenverwaltung in der App durchführen. Die Nutzer haben im Gegensatz zum Fahrzeughalter noch die Optionen zum Webservices einbinden, Standardeinstellungen hinterlegen, Szenarien erstellen, spezifische automatisierte Einstellungen aktivieren oder deaktivieren.

6. Anforderungserhebung

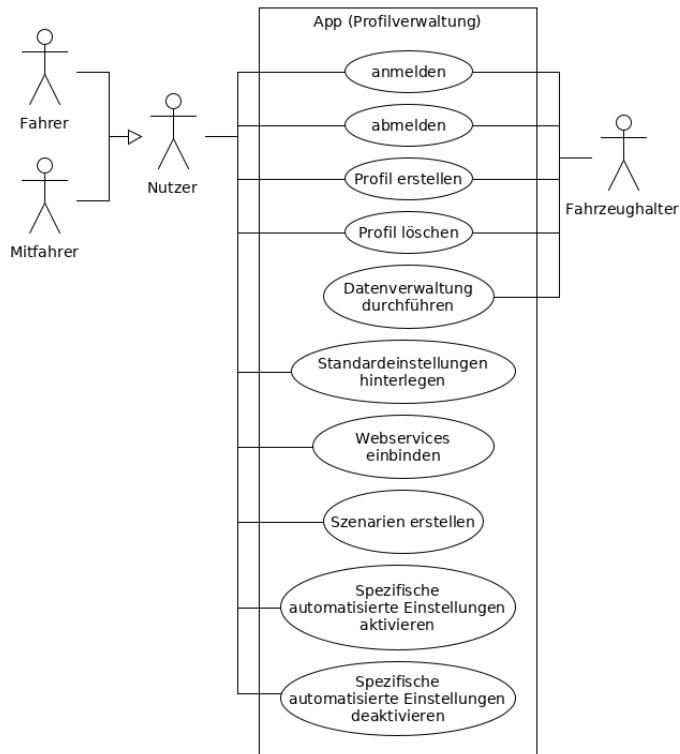


Abbildung 6.2.: Use Case App Profilverwaltung

6.3.3. Benutzereingaben

Die Abb. 6.3 Use Case Benutzereingaben beschreibt die Eingabemöglichkeiten, die Ophelia bietet. Die Mitfahrer und der Fahrer können als Nutzer *Befehle erteilen*. Dazu gehören *Gestenbefehle*, *Sprachbefehle* und die anschließende *Befehlsverarbeitung*, die der nicht-lernende Agent ausführt.

Der Fahrer kann die *Befehlsverarbeitung* von dem nicht-lernenden Agenten über die *Befehlsfunktion* *aktivieren* oder *deaktivieren*.

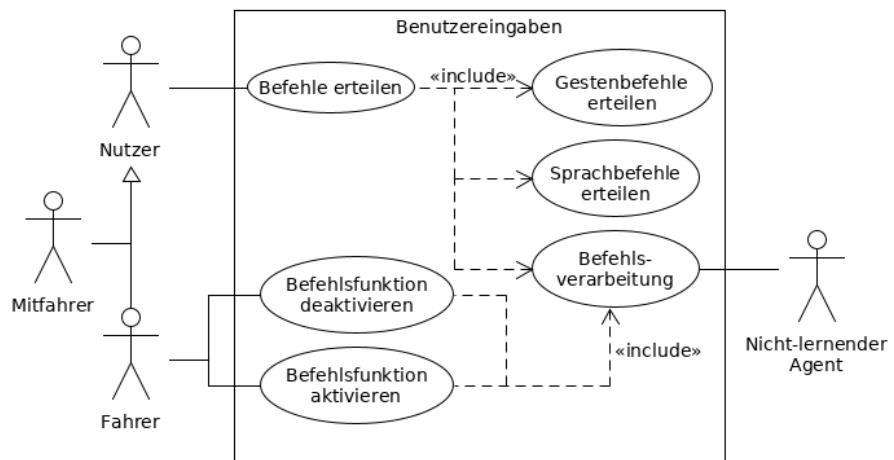


Abbildung 6.3.: Use Case Benutzereingaben

6.3.4. Datenverwaltung

Die Abb. 6.4 Use Case Datenverwaltung beschreibt die Datenverwaltung von *Ophelia*. Die Fahrer und Mitfahrer können als Nutzer die *Datenaufzeichnung ausschalten* oder ihr *Benutzerprofil löschen*. Die gewerblichen und privaten Fahrzeughalter können das *System zurücksetzen*, *Webservices freischalten*, *OTA-Updates ablehnen* und *OTA-Updates durchführen*, die bestätigt werden müssen.

6. Anforderungserhebung

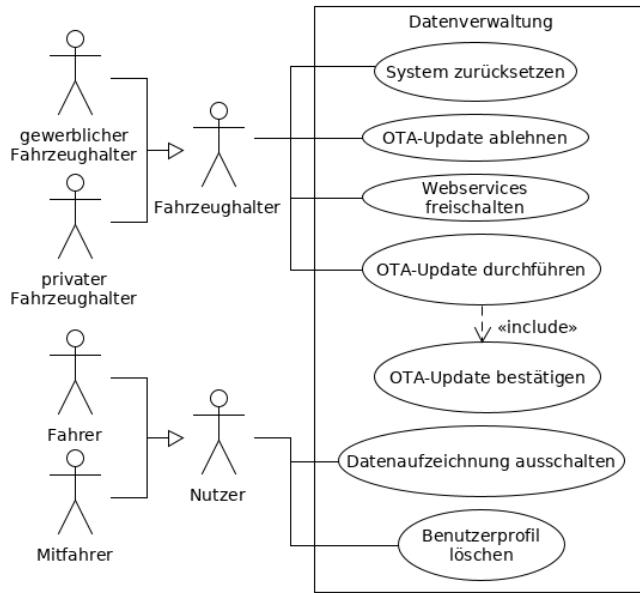


Abbildung 6.4.: Use Case Datenverwaltung

6.3.5. Innenraumbeleuchtung - Dynamische Einstellungen

Die Abb. 6.5 Use Case Innenraumbeleuchtung - Dynamische Einstellungen beschreibt die Funktionen der Innenraumbeleuchtung mit dem lernenden Agenten. Die Fahrer und Mitfahrer können als Nutzer die *Farbe* und die *Beleuchtungsstärke* der Innenraumbeleuchtung *einstellen*. Der lernende Agent lernt die *Farbe* und die *Beleuchtungsstärke* der Innenraumbeleuchtung *einzustellen*.

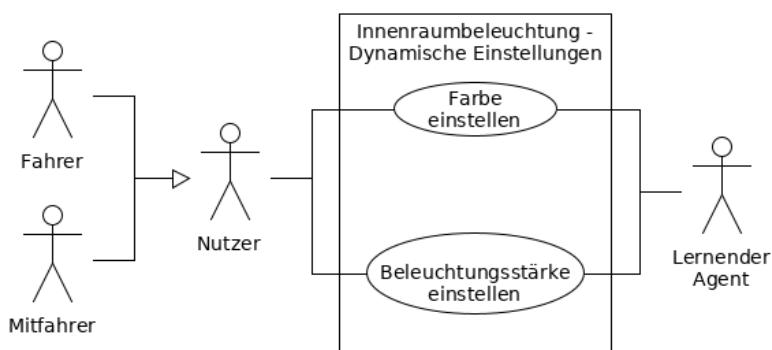


Abbildung 6.5.: Use Case Innenraumbeleuchtung - Dynamische Einstellungen

6.3.6. Innenraumbeleuchtung - Statische Einstellungen

Die Abb. 6.6 Use Case Innenraumbeleuchtung - Statische Einstellungen beschreibt die Funktionen der Innenraumbeleuchtung mit dem nicht-lernenden Agenten. Die Fahrer und Mitfahrer können als Nutzer die *Farbe* und die *Beleuchtungsstärke* der Innenraumbeleuchtung ändern. Der nicht-lernende Agent kann die *Standardfarbe* und die *Standardbeleuchtungsstärke* einstellen.

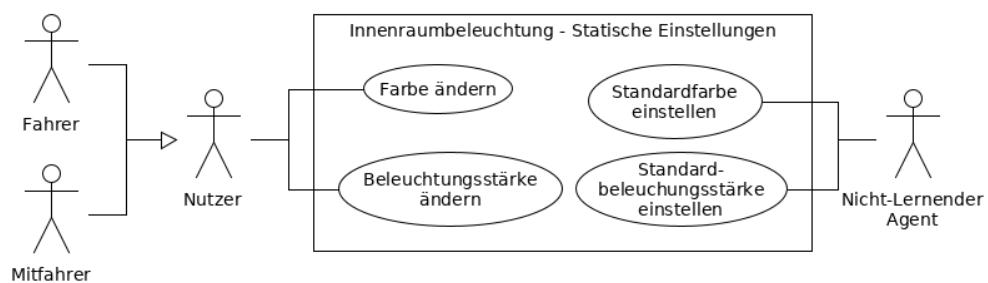


Abbildung 6.6.: Use Case Innenraumbeleuchtung - Statische Einstellungen

6.3.7. Klimatisierung - Dynamische Einstellungen

Die Abb. 6.7 Use Case Klimatisierung - Dynamische Einstellungen beschreibt die Funktionen der dynamischen Klimatisierung mit dem lernenden Agenten. Der Fahrer und die Mitfahrer können als Nutzer die *Sitze*, das *Lenkrad* und den *Innenraum klimatisieren* sowie die *Gebläsestärke einstellen*. Der lernende Agent kann ebenfalls die *Sitze*, das *Lenkrad* und den *Innenraum klimatisieren* sowie die *Gebläsestärke einstellen*. Zu der *Klimatisierung des Innenraumes* gehört es dazu *Klimazonen unabhängig voneinander zu klimatisieren*. Der lernende Agent kann selbstständig *Einstellungsänderungen lernen*.

6. Anforderungserhebung

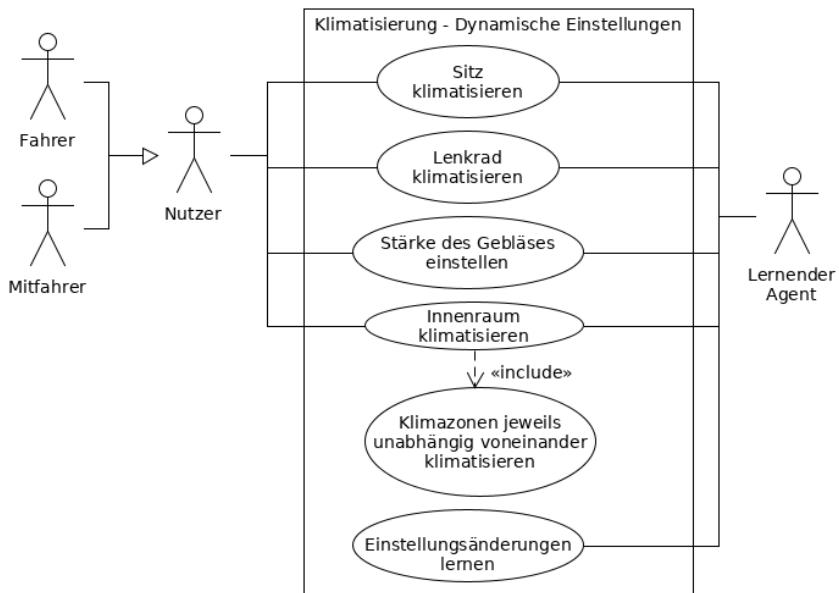


Abbildung 6.7.: Use Case Klimatisierung - Dynamische Einstellungen

6.3.8. Klimatisierung - Statische Einstellungen

Die Abb. 6.8 Use Case Klimatisierung - Statische Einstellungen beschreibt die Funktionen der statischen Klimatisierung mit dem nicht-lernenden Agenten. Der Fahrer und die Mitfahrer können als Nutzer die *Innenraumtemperatur*, die *Sitzklimatisierung*, die *Lenkradklimatisierung*, die *Gebläsestärke ändern* und die *Standardklimatisierungseinstellungen speichern*. Der nicht lernende Agent kann die *Standardklimatisierungseinstellungen vornehmen*.

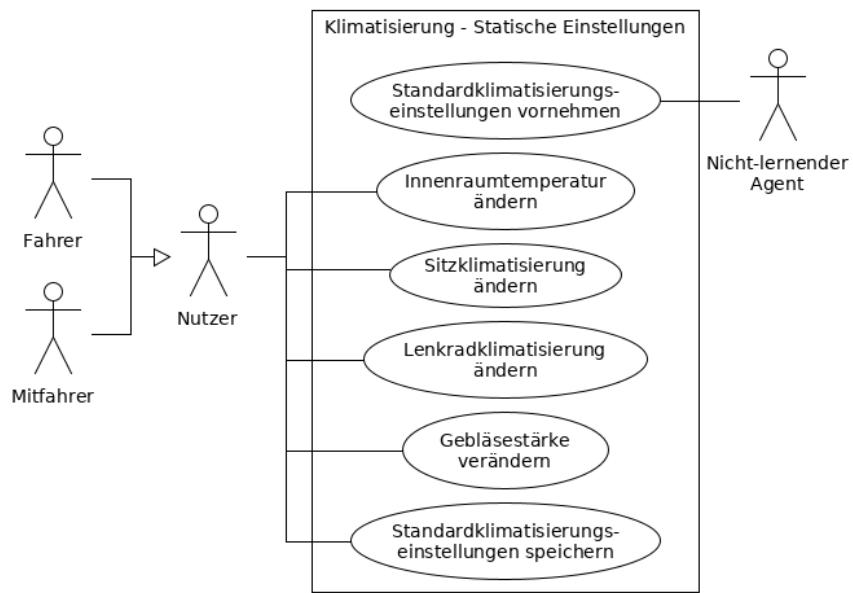


Abbildung 6.8.: Use Case Klimatisierung - Statische Einstellungen

6.3.9. Musik - Dynamische Einstellungen

Die Abb. 6.9 Use Case Musik - Dynamische Einstellungen beschreibt die Verwaltung von den dynamischen Musikeinstellungen mit dem lernenden Agenten. Die Fahrer und Mitfahrer können als Nutzer eine *Musikquelle auswählen*, dazu gehört das *Abspielen der Musik*. Optional kann der Nutzer den *Radiosender einstellen*, wenn die Musikquelle Radio ausgewählt wurde. Der Nutzer kann die *Musiklautstärke anpassen* oder den *Equalizer einstellen*. Der lernende Agent lernt die getätigten Einstellungen der Nutzer, dazu gehört das *Lernen der bevorzugten Musiklautstärke*, die *Einstellungen des Equalizers* und der *Musikquelle*, die der lernende Agent *priorisiert*.

6. Anforderungserhebung

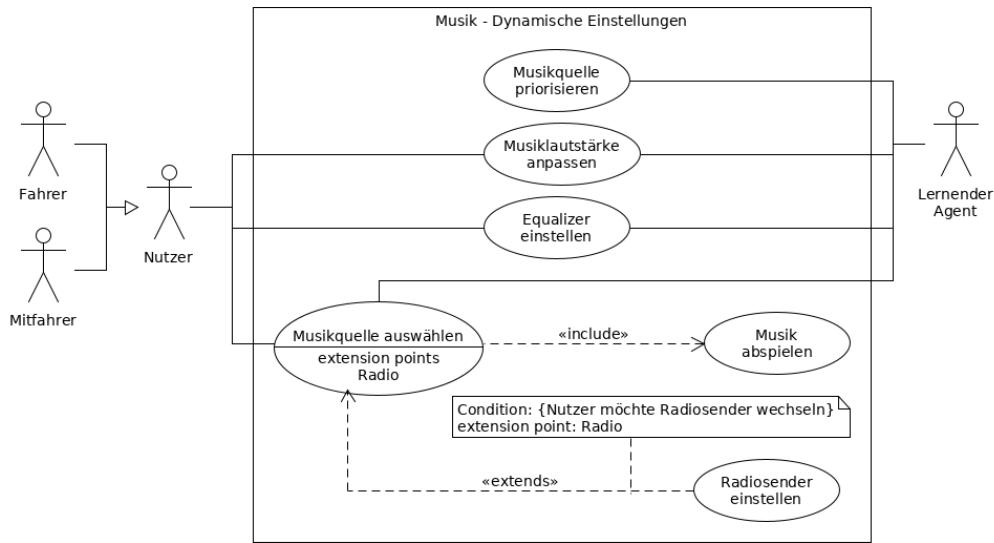


Abbildung 6.9.: Use Case Musik - Dynamische Einstellungen

6.3.10. Musik - Statische Einstellungen

Die Abb. 6.10 Use Case Musik - Statische Einstellungen beschreibt die Funktionen der statischen Musikeinstellungen. Der Fahrer und die Mitfahrer können als Nutzer die *Standardmusiklautstärke* und die *Standardequalizereinstellungen speichern*, während der nicht-lernende Agent die *Standardmusiklautstärke* und die *Standardequalizereinstellungen einstellen* und zusätzlich die *Wiedergabeliste fortsetzen* kann.

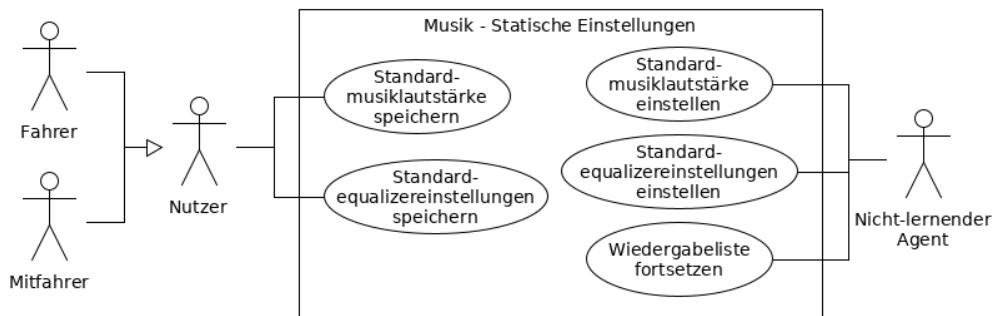


Abbildung 6.10.: Use Case Musik - Statische Einstellungen

6.3.11. Navigation - Dynamische Einstellungen

Die Abb. 6.11 Use Case Navigation - Dynamische Einstellungen beschreibt die Funktionen der Verwaltung der Navigation. Der Fahrer und die Mitfahrer können als Nutzer das *Tagesprogramm*, die *Gegenstände zu Zielen angeben* und die *Navigationslautstärke ändern*. Der lernende Agent kann selbstständig die *passende Navigationslautstärke* und die *Gegenstände zu Zielen angeben*, um *Ziele sowie Abfahrtszeiten zu lernen*. Zusätzlich zu dem *Lernen der Ziele und Abfahrtszeiten* kann der lernende Agent *Navigationsvorschläge geben*. Darüber hinaus kann der lernende Agent optional das *Navigationssystem voreinstellen* und an *Zwischenziele erinnern*, wenn der Nutzer den Navigationsvorschlag akzeptiert.

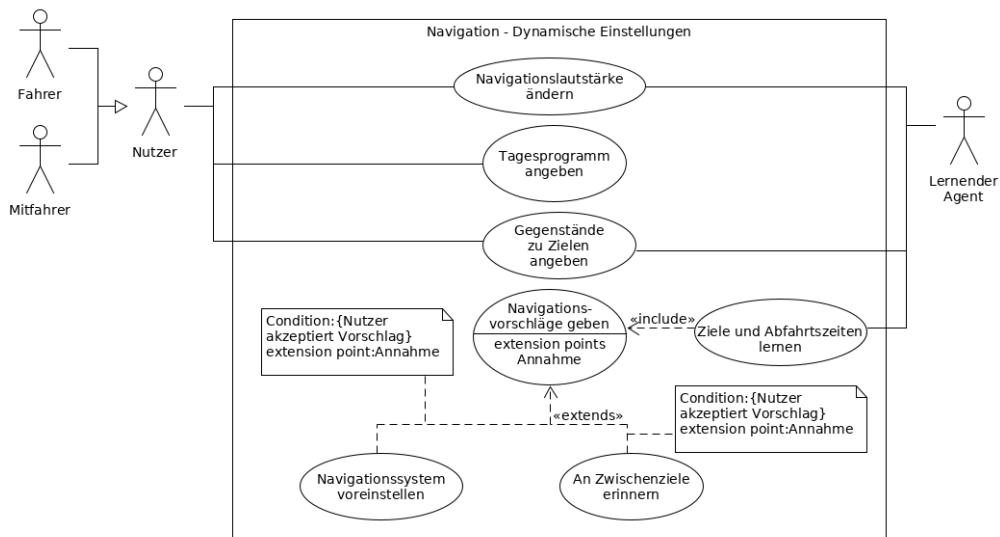


Abbildung 6.11.: Use Case Navigation - Dynamische Einstellungen

6.3.12. Navigation - Statische Einstellungen

Die Abb. 6.12 Use Case Navigation - Statische Einstellungen beschreibt die Funktionen der Verwaltung der Navigation. Der Fahrer und die Mitfahrer können als Nutzer das *Tagesprogramm*, die *Gegenstände zu Zielen angeben*, die *Standardnavigationslautstärke eingeben* und die *Navigationslautstärke ändern*. Der nicht-lernende Agent kann die *Routen optimieren*, die *Standardnavigationslautstärke einstellen*, das *Tagesprogramm einstellen*, *an das Tagesprogramm erinnern* und *an die Gegenstände erinnern*.

6. Anforderungserhebung

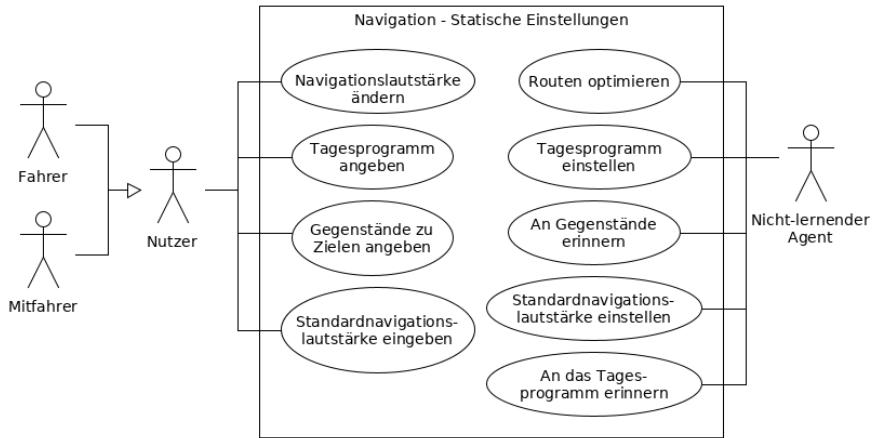


Abbildung 6.12.: Use Case Navigation - Statische Einstellungen

6.3.13. Priorisierung

Die Abb. 6.13 Use Case Priorisierung beschreibt die Verwaltung der Priorisierung. Die Mitfahrer können dem Fahrer *Einstellungen zu Musik, Navigation und der Innenraumbeleuchtung vorschlagen*. Der Fahrer kann den *Vorschlag* der Mitfahrer *ablehnen oder annehmen*, wenn der *Vorschlag angenommen* wird kann der lernende Agent die *Einstellung selbstständig vornehmen*. Der lernende Agent kann zusätzlich den Fahrer *selbstständig fragen*, ob der *Vorschlag* der Mitfahrer *akzeptiert und übernommen* werden kann.

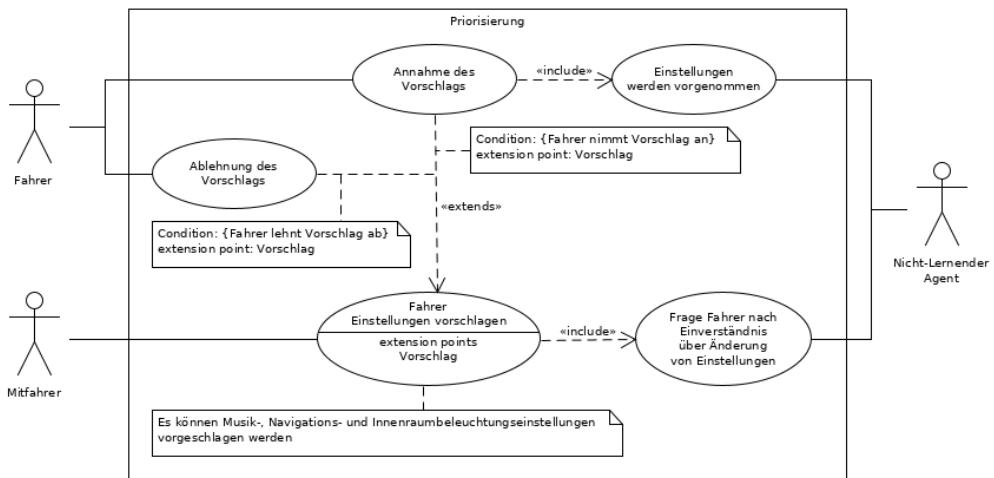


Abbildung 6.13.: Use Case Priorisierung

6.3.14. Rollos und Fenster - Dynamische Einstellungen

Die Abb. 6.14 Use Case Rollos Fenster - Dynamische Einstellungen beschreibt die Funktionen der Rollos und der Fenster. Der Fahrer und die Mitfahrer können als Nutzer die *Standardfensterhöhe speichern*, die *Fensterhöhe ändern*, die *Standardrollopositionen speichern* und die *Rollopositionen ändern*. Der lernende Agent lernt die *Rollopositionen* und die *Fensterhöhe* und kann diese selbstständig *einstellen*. Optional kann der lernende Agent lernen die *Fensterhöhe* und die *Rolloposition den klimatischen Bedingungen anzupassen*, wenn der Nutzer dies wünscht.

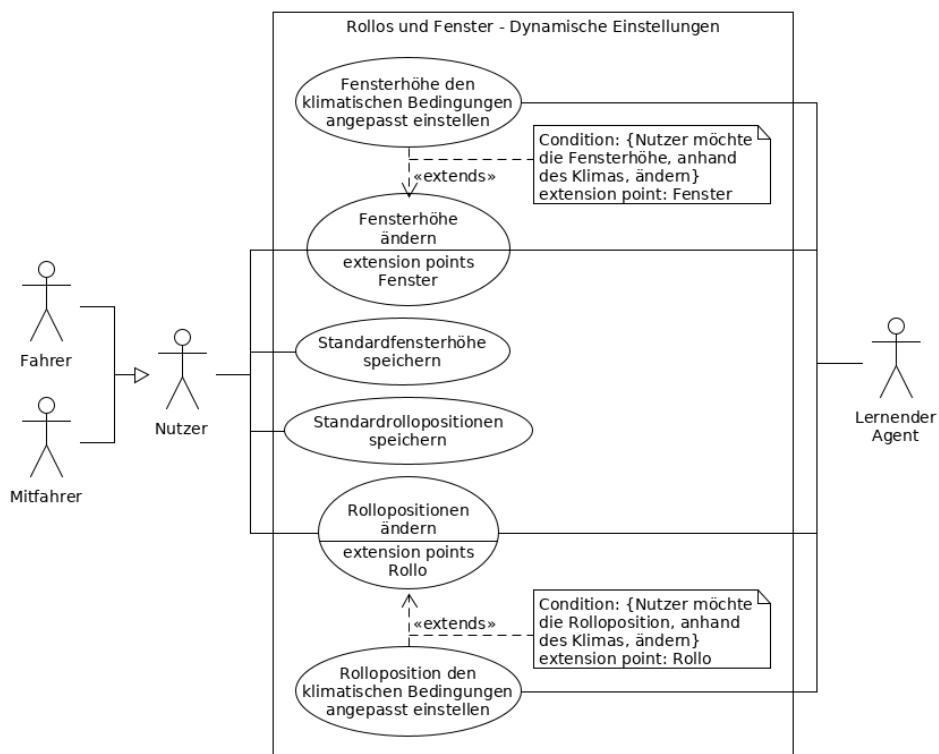


Abbildung 6.14.: Use Case Rollos Fenster - Dynamische Einstellungen

6.3.15. Rollos und Fenster - Statische Einstellungen

Die Abb. 6.15 Use Case Rollos Fenster - Statische Einstellungen beschreibt die Funktionen der Rollos und der Fenster. Der Fahrer und die Mitfahrer können als Nutzer die *Standardfensterhöhe speichern*, die *Fensterhöhe ändern*, die *Standardrollopositionen speichern* und die *Rollopositionen ändern*. Der nicht-lernende Agent kann die *Standardrolloposition* und die *Standardfensterhöhe einstellen*.

6. Anforderungserhebung

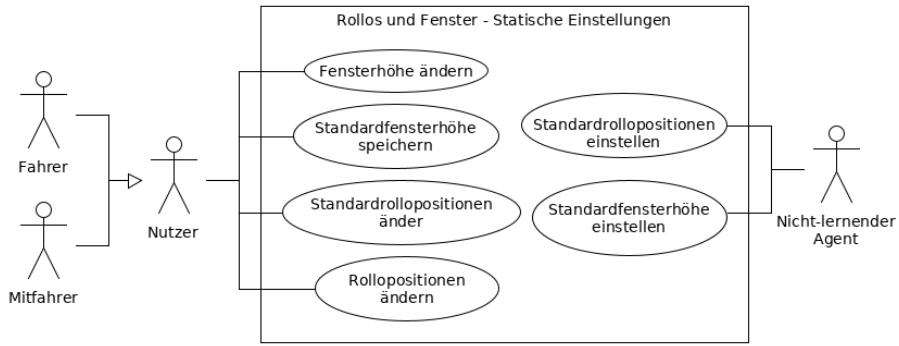


Abbildung 6.15.: Use Case Rollos Fenster - Statische Einstellungen

6.3.16. Sitzprofil

Die Abb. 6.16 Use Case Sitzprofil beschreibt die Einstellung der Spiegel, der Sitze und des Lenkrads. Der Fahrer und die Mitfahrer können als Nutzer die *Lenkradposition*, die *Sitzposition*, die *Rück- und Außenspiegelposition einstellen* und *speichern*. Der nicht-lernende Agent *stellt* die Einstellungen der *Lenkradposition*, *Sitzposition* und der *Rück- und Außenspiegelposition ein*.

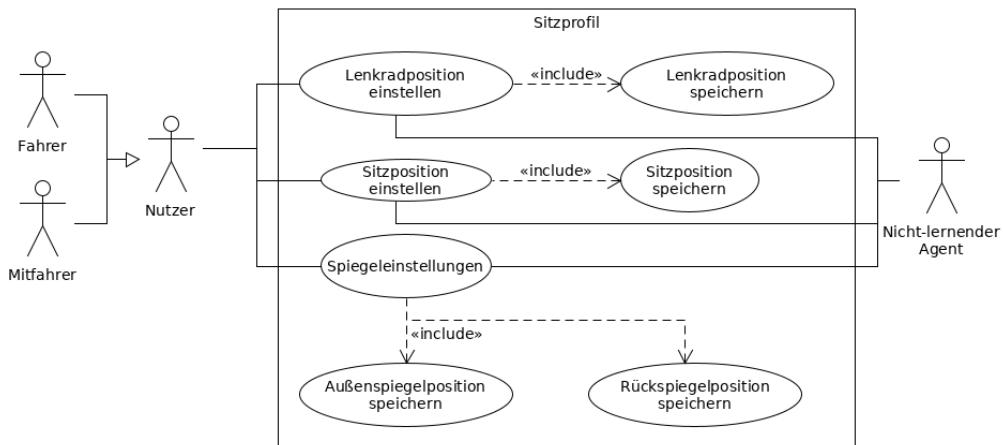


Abbildung 6.16.: Use Case Sitzprofil

6.3.17. Systementwicklung

Die Abb. 6.17 Use Case Systementwicklung beschreibt den Verlauf einer Systementwicklung von *Ophelia*. Der Entwickler kann *Ophelia* auf drei verschiedene Weisen

debuggen, er kann die *Busdaten auslesen*, die *Testroutinen aufrufen* oder die *Systeminformationen auslesen*.

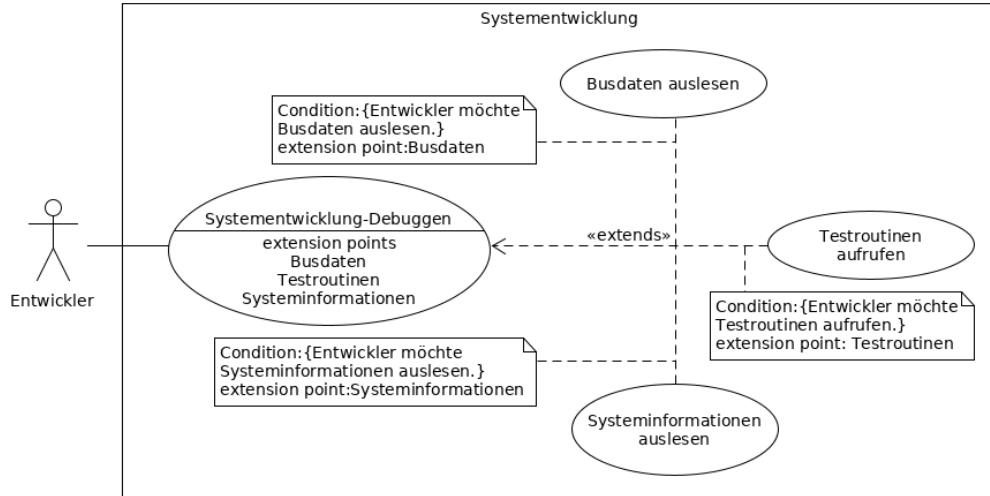


Abbildung 6.17.: Use Case Systementwicklung

6.3.18. Szenarien

Die Abb. 6.18 Use Case Szenarien beschreibt die Verwaltung von Szenarien. Der Fahrer und die Mitfahrer können als Nutzer *Szenarien verarbeiten*. Zusätzlich kann der Nutzer *Szenarien auswählen, speichern, erstellen oder bearbeiten*, wenn er dies wünscht.

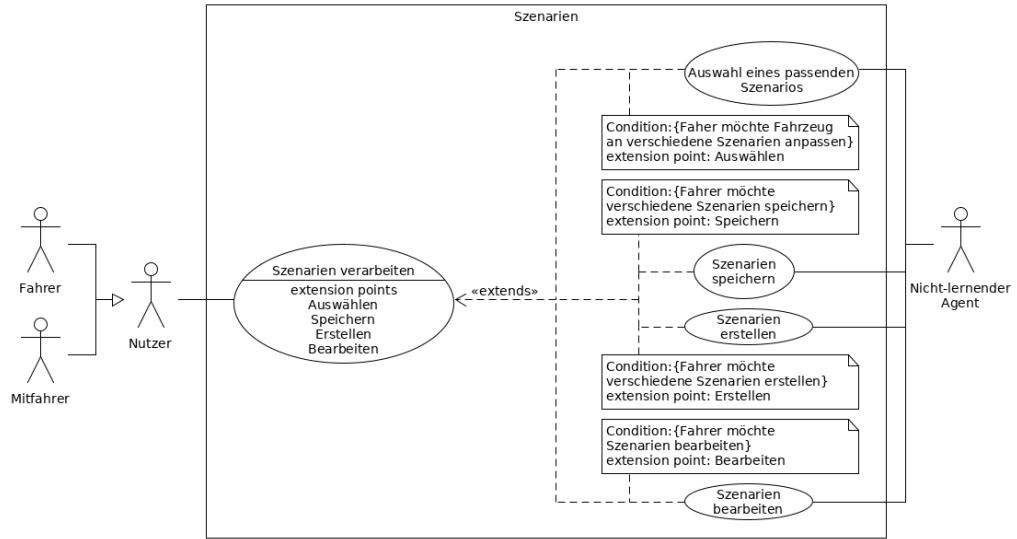


Abbildung 6.18.: Use Case Szenarien

6.4. Anforderungskatalog

Der Anforderungskatalog enthält alle Anforderungen des Projekts. Zu Beginn des Katalogs befindet sich eine Liste der funktionalen und eine Liste der nicht-funktionalen Anforderungen. Danach folgt der Abschnitt 6.4.1 Top-Level Anforderungen, der Abschnitt 6.4.2 System im Fahrzeug, der Abschnitt 6.4.3 App und zuletzt der Abschnitt 6.4.4 Server.

Liste der funktionalen Anforderungen

sys-fkt-vorein	106
sys-fkt-lern	107
sys-fkt-vorseins	107
sys-fkt-nutzerkenn	107
sys-fkt-erinnern	108
sys-fkt-touchmin	108
sys-fkt-tastenmin	109
sys-fkt-otaupdates	109
sys-fkt-internet	109

sys-fkt-navfunk	110
sys-fkt-musfunk	110
sys-fkt-klimafunk	111
sys-fkt-app	111
sys-fkt-server	111
sys-fkt-nutzschnitt	112
mus-fkt-prio	115
mus-fkt-prioreset	115
mus-fkt-laut	115
mus-fkt-lauterhoehen	116
mus-fkt-lautverringern	116
mus-fkt-lautort	117
mus-fkt-laupteing	117
mus-fkt-laupteings	118
mus-fkt-laupteingg	118
nav-fkt-statischeins	119
nav-fkt-slauteinstellen	119
nav-fkt-slauteing	120
nav-fkt-slauteinghmi	120
nav-fkt-slauteings	121
nav-fkt-slauteingg	122
nav-fkt-slautaendern	122
nav-fkt-slauterhoehenhmi	123
nav-fkt-slauterhoehensprache	124
nav-fkt-slauterhoehengeste	124
nav-fkt-slautverringernhmi	125
nav-fkt-slautverringernsprache	126
nav-fkt-lautstaerkeverringengeste	126
nav-fkt-lautwerte	127
nav-fkt-lautzuruecksetzen	127
nav-fkt-slautspeichern	128
nav-fkt-lautaendern	129
nav-fkt-lauterhoehenhmi	129
nav-fkt-lauterhoehensprache	130
nav-fkt-lauterhoehengeste	130
nav-fkt-lautverringernhmi	131
nav-fkt-lautstaerkeverringengeste	131
nav-fkt-lautstaerkeaktorik	132
nav-fkt-zielvor	132
nav-fkt-zieleingabe	133
nav-fkt-tpeinstellen	133
nav-fkt-zwischenzieleerinnern	134
nav-fkt-zwischenzieleerfassen	134
nav-fkt-zwischenzielespeichern	135

Liste der funktionalen Anforderungen

nav-fkt-zwischenzieleanserver	135
nav-fkt-routenoptimierung	136
nav-fkt-gegenstaendeerinnern	136
nav-fkt-tagesprogrammerinnern	137
nav-fkt-heimateingabe	137
nav-fkt-heimatsave	138
nav-fkt-heimatchange	138
nav-fkt-heimatdel	139
nav-fkt-heimatziel	139
nav-fkt-heimatroutedel	140
nav-fkt-routeverlaufwahl	140
nav-fkt-routezieleing	141
nav-fkt-routezielchange	141
nav-fkt-routezieldel	142
nav-fkt-routezielsave	142
nav-fkt-routezielchange	143
nav-fkt-favoritadresse	143
nav-fkt-favoritadresseerstellen	144
nav-fkt-favoritadressenspeichern	144
nav-fkt-zielalsfavoritwaehlen	144
nav-fkt-favoritadressealszielwaehlen	145
nav-fkt-favoritadresseloeschen	145
nav-fkt-parkplatzortspeichern	146
nav-fkt-routenoptimierungverkehrsmeldung	146
nav-fkt-routenoptimierungdynamischeanpassung	147
nav-fkt-lautstaerkeanpassenpraef	147
nav-fkt-lautstaerkeeinstellen	148
nav-fkt-lautstaerkeanpassensit	148
nav-fkt-lautstaerkeanpasseninnen	149
nav-fkt-lautstaerkeanpasseninnenmusik	149
nav-fkt-lautstaerkeanpasseninnenradio	150
nav-fkt-lautstaerkesensoren	150
nav-fkt-vorsziele	150
nav-fkt-vorsrouten	151
nav-fkt-vorsrouteeinstellen	151
nav-fkt-vorsroutezieleerinn	152
nav-fkt-vorsrouteablehn	152
nav-fkt-vorsannehm	152
nav-fkt-vorsannehmsprache	153
nav-fkt-vorsannehmgeste	153
nav-fkt-vorsablehn	153
nav-fkt-vorsablehnsprache	154
nav-fkt-vorsablehngeste	154
nav-fkt-tpeingabe	155

nav-fkt-tpeingabesprache	155
nav-fkt-tpeingabegeste	155
ben-fkt-touch	156
rol-fkt-ein	157
klima-fkt-autoeins	157
nav-fkt-abfahrzeitdaten	157
nav-fkt-abfahrzeitdaten	158
nav-fkt-app	158
nav-fkt-tagesprogrammeingabe	159
nav-fkt-tagesprogrammziele	159
nav-fkt-tagesprogrammspeichern	160
nav-fkt-tagesprogrammauswahl	160
nav-fkt-tagesprogrammaendern	161
nav-fkt-tagesprogrammzielhinzufuegen	161
nav-fkt-tagesprogrammzielentfernen	162
nav-fkt-tagesprogrammzielenfolge	162
nav-fkt-tagesprogrammaenderungspeichern	163
nav-fkt-tagesprogrammloeschen	163
nav-fkt-tagesprogrammsenden	164
nav-fkt-tagesprogrammabfahrtszeit	164
nav-fkt-tagesprogrammabfahrtszeitspeichern	165
nav-fkt-tagesprogrammabfahrtszeitaendern	165
nav-fkt-tagesprogrammabfahrtszeitaenderspeichern	166
nav-fkt-tagesprogrammabfahrtszeitloeschen	166
nav-fkt-tagesprogrammabfahrtszeitsendenfahrzeug	167
nav-fkt-tagesprogrammabfahrtszeitsendenserver	168
nav-fkt-lauteingabeapp	168
nav-fkt-lauterhoehenapp	169
nav-fkt-lautverringernapp	169
nav-fkt-gegenstaendeangabe	170
nav-fkt-gegenstaendeaendern	171
nav-fkt-gegenstaendloeschen	171
nav-fkt-gegenstaenddepushapp	172
nav-fkt-gegenstaendedepushappdeak	172
nav-fkt-parkplatzortabrufen	173
nav-fkt-lautspeichserver	173
nav-fkt-lautloeschserver	174
nav-fkt-routenspeichserver	174
nav-fkt-routenloeschserver	175
nav-fkt-zielespeichserver	175
nav-fkt-zieleloeschserver	176
nav-fkt-abfahrtszeitspeichserver	176
nav-fkt-tpspeichserver	177
nav-fkt-gegenstaendespeichserver	177

nav-fkt-gegenstaendeloeschserver	177
ki-fkt-lernkomponente	178
ki-fkt-abfahrzeitzielelern	178
ki-fkt-lautaenderlern	179
ki-fkt-lautlernausgeben	179
ki-fkt-zielelernausgeben	179
ki-fkt-abfahrtzeitlernausgeben	180

Liste der nicht-funktionalen Anforderungen

sys-nfkt-port	112
sys-nfkt-datenschutz	112
sys-nfkt-datenuebertragung	113
sys-nfkt-datenspeicherung	113
sys-nfkt-manipulationssicherheit	114
sys-nfkt-isosicherheit	114

6.4.1. Top-Level Anforderungen

ID	sys-fkt-vorein
Anforderungstyp	funktional
Name	Voreinstellungen
Anforderung	Ophelia muss Voreinstellungen im Fahrzeug vornehmen können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	Vision
Referenzen	Voreinstellung
Zuletzt geändert von	Daniel, 07.07.19
Status	offen
Begründung	Das System soll das Komfortgefühl der Nutzer erhöhen, indem Voreinstellungen vorgenommen werden. Voreinstellungen umfassen gelernte sowie nicht gelernte Einstellungen.
Wird umgesetzt	

Liste der nicht-funktionalen Anforderungen

ID	sys-fkt-lern
Anforderungstyp	funktional
Name	Präferenzen lernen
Anforderung	Ophelia muss Präferenzen des Nutzers lernen können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	Vision
Referenzen	Visionspräsentation
Zuletzt geändert von	Daniel, 13.06.19
Status	offen
Begründung	Durch das Lernen von Präferenzen sollen Einstellungen im Fahrzeug von einer KI vorgenommen werden. Damit muss der Nutzer diese Einstellungen nicht selbst tätigen.
Wird umgesetzt	

ID	vsys-fkt-vorseins
Anforderungstyp	funktional
Name	Vorschläge für Einstellungen
Anforderung	Ophelia muss dem Fahrer während der Fahrt Einstellungen vorschlagen können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	Vision
Referenzen	Visionspräsentation
Zuletzt geändert von	Daniel, 13.06.19
Status	offen
Begründung	Bevor Einstellungen von der KI während der Fahrt durchgeführt werden soll der Nutzer über einen Vorschlag nach der Einstellung gefragt werden.
Wird umgesetzt	

ID	sys-fkt-nutzerkenn
Anforderungstyp	funktional
Name	Nutzer erkennen
Anforderung	Ophelia muss den Nutzer erkennen können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	Vision

Liste der nicht-funktionalen Anforderungen

Referenzen	Visionspräsentation
Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	Durch eine Nutzererkennung können Voreinstellungen im Fahrzeug vorgenommen werden, wenn der Nutzer erkannt wird.
Wird umgesetzt	

ID	sys-fkt-erinnern
Anforderungstyp	funktional
Name	Erinnerungen
Anforderung	Ophelia muss den Nutzer eine Erinnerungsfunktion bereitstellen können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	User Story Navigation 4, User Story Navigation 6, User Story Navigation 8, User Story Navigation 11
Referenzen	Visionspräsentation
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	Erinnerungen machen es dem Nutzer einfacher, indem ihnen die Arbeit abgenommen wird, an Dinge zu denken.
Wird umgesetzt	

ID	sys-fkt-touchmin
Anforderungstyp	funktional
Name	Alternative zu Touch-Eingaben
Anforderung	Ophelia muss dem Nutzer eine Alternative zu Touch-Eingaben bereitstellen.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	
Referenzen	Visionspräsentation
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	Die Berührungseingaben im Fahrzeug sollen durch eine Alternative Bedienung im Fahrzeug verringert werden.
Wird umgesetzt	

Liste der nicht-funktionalen Anforderungen

ID	sys-fkt-tastenmin
Anforderungstyp	funktional
Name	Alternative zu Tasten Bedienungen
Anforderung	Ophelia muss dem Nutzer eine Alternative zu Tasten Bedienungen bereitstellen.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	Vision
Referenzen	Visionspräsentation
Zuletzt geändert von	Daniel, 13.06.19
Status	offen
Begründung	Die Tasteneingaben im Fahrzeug sollen durch eine Alternative Bedienung im Fahrzeug verringert werden.
Wird umgesetzt	

ID	sys-fkt-otaupdates
Anforderungstyp	funktional
Name	OTA Updates
Anforderung	Ophelia muss dem Nutzer die Möglichkeit bieten, die Systemsoftware im Fahrzeug über Over-the-Air Updates aktualisieren zu können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	Vision
Referenzen	Visionspräsentation
Zuletzt geändert von	Daniel, 07.07.19
Status	offen
Begründung	Ein Nutzer soll die Systemsoftware des Fahrzeugs aktualisieren können, ohne eine Werkstatt besuchen zu müssen.
Wird umgesetzt	

ID	sys-fkt-internet
Anforderungstyp	funktional
Name	Verbindung über Internet
Anforderung	Die Systemkomponenten von Ophelia müssen eine Verbindung mit dem Internet herstellen können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	

Liste der nicht-funktionalen Anforderungen

Ursprung	Vision
Referenzen	Visionspräsentation
Zuletzt geändert von	Daniel, 07.07.19
Status	offen
Begründung	Zur Kommunikation der Komponenten untereinander.
Wird umgesetzt	

ID	sys-fkt-navfunk
Anforderungstyp	funktional
Name	Navigationsfunktionen
Anforderung	Ophelia muss dem Nutzer die Möglichkeit bieten, Navigationsfunktionen verwenden zu können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	Vision
Referenzen	Visionspräsentation
Zuletzt geändert von	Daniel, 07.07.19
Status	offen
Begründung	Das Metasystem soll Navigationsfunktionen anbieten, damit Navigationsdaten gelernt werden können.
Wird umgesetzt	

ID	sys-fkt-musfunk
Anforderungstyp	funktional
Name	Musikfunktionen
Anforderung	Ophelia muss dem Nutzer die Möglichkeit bieten, Musikfunktionen verwenden zu können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	Vision
Referenzen	Visionspräsentation
Zuletzt geändert von	Daniel, 07.07.19
Status	offen
Begründung	
Wird umgesetzt	Das Metasystem soll eine Musikfunktionen anbieten, damit Musikdaten gelernt werden können.

Liste der nicht-funktionalen Anforderungen

ID	sys-fkt-klimafunk
Anforderungstyp	funktional
Name	Klimatisierungsfunktionen
Anforderung	Ophelia muss dem Nutzer die Möglichkeit bieten, Klimatisierungsfunktionen verwenden zu können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	Vision
Referenzen	Visionspräsentation
Zuletzt geändert von	Daniel, 07.07.19
Status	offen
Begründung	Das Metasystem soll Klimatisierungsfunktionen anbieten, damit Klimatisierungsdaten gelernt werden können.
Wird umgesetzt	

ID	sys-fkt-app
Anforderungstyp	funktional
Name	Mobile Applikation
Anforderung	Ophelia muss dem Nutzer die Möglichkeit bieten, eine mobile Applikation verwenden zu können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	Vision
Referenzen	Visionspräsentation
Zuletzt geändert von	Daniel, 07.07.19
Status	offen
Begründung	Durch eine mobile Applikation soll der Nutzer Einstellungen im Fahrzeug von allen Standorten durchführen können.
Wird umgesetzt	

ID	sys-fkt-server
Anforderungstyp	funktional
Name	Server
Anforderung	Ophelia muss einen Server beinhalten.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	Vision

Liste der nicht-funktionalen Anforderungen

Referenzen	Visionspräsentation
Zuletzt geändert von	Daniel, 07.07.19
Status	offen
Begründung	Um OtA Updates bereitzustellen.
Wird umgesetzt	

ID	sys-fkt-nutzschnitt
Anforderungstyp	nicht-funktional
Name	Intuitive Nutzerschnittstelle
Anforderung	Ophelia muss eine intuitive Nutzerschnittstelle haben.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	Vision
Referenzen	Visionspräsentation
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	Eine intuitive Nutzerschnittstelle soll dem Nutzer die Bedienung erleichtern.
Wird umgesetzt	

ID	sys-nfkt-port
Anforderungstyp	nicht-funktional
Name	Portabilität auf PKWs
Anforderung	Wenn Personenkraftwagen die notwendigen Schnittstellen für Ophelia bereitstellen, muss Ophelia Portabilität auf diese Personenkraftwagen ermöglichen.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	Vision
Referenzen	Visionspräsentation
Zuletzt geändert von	Daniel, 13.06.19
Status	offen
Begründung	Damit das System in Personenkraftwagen von verschiedenen Herstellern verwendet werden kann.
Wird umgesetzt	

ID	sys-nfkt-datenschutz
Anforderungstyp	nicht-funktional

Name	Datenschutz
Anforderung	Ophelia muss Daten gemäß Datenschutzbestimmungen behandeln.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel, 07.07.19
Status	offen
Begründung	Wenn personenbezogene Daten erhoben werden, müssen Datenschutzbestimmungen eingehalten werden.
Wird umgesetzt	

ID	sys-nfkt-datenuebertragung
Anforderungstyp	nicht-funktional
Name	Sichere Datenübertragung
Anforderung	Ophelia muss ein sicheres Verschlüsselungsverfahren zur Datenübertragung verwenden.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel, 07.07.19
Status	offen
Begründung	Um Daten bei der Übertragung vor unauthentifizierten Zugriff zu schützen.
Wird umgesetzt	

ID	sys-nfkt-datenspeicherung
Anforderungstyp	nicht-funktional
Name	Datenspeicherung
Anforderung	Ophelia muss ein sicheres Verschlüsselungsverfahren zur Datenspeicherung verwenden.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	Stakeholderanalyse
Referenzen	

Liste der nicht-funktionalen Anforderungen

Zuletzt geändert von	Daniel, 07.07.19
Status	offen
Begründung	Um Daten vor unauthentifizierten Zugriffen von Dritten zu schützen.
Wird umgesetzt	

ID	sys-nfkt-manipulationssicherheit
Anforderungstyp	nicht-funktional
Name	Absicherung vor Manipulation
Anforderung	Ophelias Systemsoftware muss vor außenstehender Manipulation abgesichert sein.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	Um die Systemsoftware vor Änderungen von Dritten zu schützen.
Wird umgesetzt	

ID	sys-nfkt-isosicherheit
Anforderungstyp	nicht-funktional
Name	Funktionale Sicherheit
Anforderung	Ophelias muss der ISO 26262 im Bezug auf funktionale Sicherheit genügen.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	Im Rahmen der Implementierung auf einen Personenwagen müssen sicherheitskritische Aspekte für das System herangezogen werden.
Wird umgesetzt	

6.4.2. System im Fahrzeug

Musik

ID	mus-fkt-prio
Anforderungstyp	funktional
Name	Priorisierung der Musikquelle
Anforderungssatz	Das Das <i>Ophelia</i> -Modul muss die Wiedergabequellen der Musik der Nutzer priorisieren.
Nachgewiesen durch	
Akzeptanzkriterium	Vorbedingungen, Schritte (Was der Nutzer macht), Nachbedingungen
Abgeleitet von	sys-fkt-lern
Ursprung	User Story Musik 1
Referenzen	
Zuletzt geändert von	14.06.19
Status	offen
Begründung	Durch eine Priorisierung der Wiedergabequellen der Musik soll der Nutzer die Wiedergabequelle nicht selbst einstellen müssen.
Wird umgesetzt	

ID	mus-fkt-prioreset
Anforderungstyp	funktional
Name	Priorisierung der Musikquelle zurücksetzen
Anforderungssatz	Das <i>Ophelia</i> -Modul muss die Priorisierung der Wiedergabequellen der Musik der Nutzer zurücksetzen.
Nachgewiesen durch	
Akzeptanzkriterium	Vorbedingungen, Schritte (Was der Nutzer macht), Nachbedingungen
Abgeleitet von	-
Ursprung	User Story Musik 1
Referenzen	
Zuletzt geändert von	23.06.19 von Artem
Status	offen
Begründung	Eine Priorisierung der Wiedergabequellen der Musik soll zurückgesetzt werden, wenn der Nutzer es wünscht.
Wird umgesetzt	

ID	mus-fkt-laut
Anforderungstyp	funktional

Liste der nicht-funktionalen Anforderungen

Name	Standardlautstärke speichern
Anforderungssatz	Das <i>Ophelia</i> -Modul muss die Standardlautstärke der Musik speichern.
Nachgewiesen durch	
Akzeptanzkriterium	Vorbedingungen, Schritte (Was der Nutzer macht), Nachbedingungen
Abgeleitet von	
Ursprung	User Story Musik 2
Referenzen	Standardlautstärke im Glossar
Zuletzt geändert von	14.06.19
Status	offen
Begründung	Durch eine einmalige Wahl der Musiklautstärke soll die Systeminteraktion mit dem Nutzer verringert werden.
Wird umgesetzt	

ID	mus-fkt-lauterhoehen
Anforderungstyp	funktional
Name	Standardlautstärke erhöhen
Anforderungssatz	Das <i>Ophelia</i> -Modul muss die Standardlautstärke der Musik erhöhen.
Nachgewiesen durch	
Akzeptanzkriterium	Vorbedingungen, Schritte (Was der Nutzer macht), Nachbedingungen
Abgeleitet von	-
Ursprung	User Story Musik 2
Referenzen	Standardlautstärke im Glossar
Zuletzt geändert von	23.06.19 von Artem
Status	offen
Begründung	Durch eine einmalige Erhöhung der Musiklautstärke soll die Systeminteraktion mit dem Nutzer erhöht werden.
Wird umgesetzt	

ID	mus-fkt-lautverringern
Anforderungstyp	funktional
Name	Standardlautstärke verringern
Anforderungssatz	Das <i>Ophelia</i> -Modul muss die Standardlautstärke der Musik verringern.
Nachgewiesen durch	
Akzeptanzkriterium	Vorbedingungen, Schritte (Was der Nutzer macht), Nachbedingungen
Abgeleitet von	-

Ursprung	User Story Musik 2
Referenzen	Standardlautstärke im Glossar
Zuletzt geändert von	23.06.19 von Artem
Status	offen
Begründung	Durch eine einmalige Wahl der Musiklautstärke soll die Systeminteraktion mit dem Nutzer verringert werden.
Wird umgesetzt	

ID	mus-fkt-lautort
Anforderungstyp	funktional
Name	Standardlautstärke Speicherort
Anforderungssatz	Das <i>Ophelia</i> -Modul muss die Standardlautstärke der Musik im Profil des Nutzers speichern.
Nachgewiesen durch	
Akzeptanzkriterium	Vorbedingungen, Schritte (Was der Nutzer macht), Nachbedingungen
Abgeleitet von	mus-fkt-laut
Ursprung	User Story Musik 2
Referenzen	Standardlautstärke im Glossar
Zuletzt geändert von	14.06.19
Status	offen
Begründung	Damit die vom Nutzer gewünschte Standardlautstärke vorgenommen werden kann muss diese in <i>Ophelia</i> gespeichert werden.
Wird umgesetzt	

ID	mus-fkt-lauteing
Anforderungstyp	funktional
Name	Standardlautstärke Eingabe
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Standardlautstärke der Musik eingeben zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Vorbedingungen, Schritte (Was der Nutzer macht), Nachbedingungen
Abgeleitet von	mus-fkt-laut
Ursprung	User Story Musik 2
Referenzen	Standardlautstärke im Glossar
Zuletzt geändert von	14.06.19
Status	offen

Liste der nicht-funktionalen Anforderungen

Begründung	Damit der Nutzer dem System eine Standardlautstärke mitteilen kann, muss er eine Möglichkeit haben diese in <i>Ophelia</i> eingeben zu können.
Wird umgesetzt	

ID	mus-fkt-lauteings
Anforderungstyp	funktional
Name	Standardlautstärke Eingabe Sprache
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Standardlautstärke der Musik über eine Sprachsteuerung eingeben zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Vorbedingungen, Schritte (Was der Nutzer macht), Nachbedingungen
Abgeleitet von	mus-fkt-lauteing
Ursprung	User Story Musik 2
Referenzen	Standardlautstärke im Glossar
Zuletzt geändert von	14.06.19
Status	offen
Begründung	
Wird umgesetzt	

ID	mus-fkt-lauteingg
Anforderungstyp	funktional
Name	Standardlautstärke Eingabe Geste
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Standardlautstärke der Musik über eine Gestensteuerung eingeben zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Vorbedingungen, Schritte (Was der Nutzer macht), Nachbedingungen
Abgeleitet von	mus-fkt-lauteing
Ursprung	User Story Musik 2
Referenzen	Standardlautstärke im Glossar
Zuletzt geändert von	14.06.19
Status	offen
Begründung	
Wird umgesetzt	

Navigation

ID	nav-fkt-statischeins
Anforderungstyp	funktional
Name	Statische Einstellungen
Anforderungssatz	Das <i>Ophelia</i> -Modul muss alle statischen Einstellungen des Nutzers bezüglich der Navigation vor Fahrtantritt im Fahrzeug einstellen können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer statische Einstellungen bezüglich der Navigation in - gespeichert hat (Vorbedingung) und der Nutzer die Fahrt antritt (Aktion), sind die vom Nutzer hinterlegte statischen Einstellungen bezüglich der Navigation im Fahrzeug eingestellt (Nachbedingung).
Abgeleitet von	sys-fkt-vorein
Ursprung	
Referenzen	statisch
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	Vom Nutzer hinterlegte Einstellungen bezüglich der Navigation sollen vor Fahrtantritt eingestellt werden.
Wird umgesetzt	

ID	nav-fkt-slauteinstellen
Anforderungstyp	funktional
Name	Standardnavigationslautstärke einstellen
Anforderungssatz	Das <i>Ophelia</i> -Modul muss die im Nutzerprofil gespeicherte Standardnavigationslautstärke vor Fahrtantritt im Fahrzeug einstellen können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer eine Standardnavigationslautstärke in <i>Ophelia</i> gespeichert hat (Vorbedingung) und der Nutzer die Fahrt antritt (Aktion), ist die vom Nutzer hinterlegte Standardnavigationslautstärke im Fahrzeug eingestellt (Nachbedingung).
Abgeleitet von	nav-fkt-statischeins
Ursprung	User Story Navigation 2
Referenzen	
Zuletzt geändert von	Daniel, 09.07.19
Status	offen

Liste der nicht-funktionalen Anforderungen

Begründung	Um das Fahrzeug den Bedürfnissen des Nutzers anzupassen, muss <i>Ophelia</i> die Standardnavigationslautstärke einstellen können.
Wird umgesetzt	

ID	nav-fkt-slauteing
Anforderungstyp	funktional
Name	Standardnavigationslautstärke Eingabe
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Standardnavigationslautstärke eingeben zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die Absicht hat, seine über einensönlische Standardnavigationslautstärke in <i>Ophelia</i> zu setzen (Vorbedingung), stellt <i>Ophelia</i> ein UI-Element bereit, um die Standardnavigationslautstärke im Fahrzeug eingeben zu können (Nachbedingung).
Akzeptanzkriterium 2	Wenn der Nutzer mit dem UI-Element interagiert (Aktion), wird die Standardnavigationslautstärke im Fahrzeug eingegeben (Nachbedingung).
Abgeleitet von	nav-fkt-slauteinstellen
Ursprung	User Story Navigation 1
Referenzen	Standardlautstärke
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	Damit der Nutzer <i>Ophelia</i> eine Standardnavigationslautstärke mitteilen kann, muss der Nutzer eine Möglichkeit haben diese in <i>Ophelia</i> eingeben zu können.
Wird umgesetzt	

ID	nav-fkt-slauteinghmi
Anforderungstyp	funktional
Name	Standardnavigationslautstärke Eingabe HMI
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Standardnavigationslautstärke über ein HMI im Fahrzeug eingeben zu können.
Nachgewiesen durch	

Liste der nicht-funktionalen Anforderungen

Akzeptanzkriterium	Wenn der Nutzer die Absicht hat, seine Standardnavigationslautstärke in <i>Ophelia</i> zu setzen (Vorbedingung), stellt <i>Ophelia</i> ein UI-Element bereit, um die Standardnavigationslautstärke im Fahrzeug eingeben zu können (Nachbedingung).
Akzeptanzkriterium 2	Wenn der Nutzer mit dem UI-Element interagiert (Aktion), wird die Standardnavigationslautstärke im Fahrzeug eingegeben (Nachbedingung).
Abgeleitet von	nav-fkt-slauteinstellen
Ursprung	User Story Navigation 1
Referenzen	Standardlautstärke
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	Damit der Nutzer <i>Ophelia</i> eine Standardnavigationslautstärke mitteilen kann, muss der Nutzer eine Möglichkeit haben diese in <i>Ophelia</i> eingeben zu können.
Wird umgesetzt	

ID	nav-fkt-slauteings
Anforderungstyp	funktional
Name	Standardnavigationslautstärke Eingabe Sprache
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Standardnavigationslautstärke über einen Sprachbefehl einzugeben zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die Absicht hat, seine über einen Sprachbefehl Standardnavigationslautstärke in <i>Ophelia</i> einzugeben (Vorbedingung), stellt <i>Ophelia</i> einen Sprachbefehl bereit, um die Standardnavigationslautstärke im Fahrzeug einzugeben zu können (Nachbedingung).
Akzeptanzkriterium 2	Wenn der Nutzer den Sprachbefehl aussagt (Aktion), wird die Standardnavigationslautstärke im Fahrzeug eingegeben (Nachbedingung).
Abgeleitet von	nav-fkt-slauteing
Ursprung	User Story Navigation 1
Referenzen	Standardlautstärke
Zuletzt geändert von	Daniel, 09.07.19
Status	offen

Liste der nicht-funktionalen Anforderungen

Begründung	Damit der Nutzer dem System eine Standardnavigationslautstärke mitteilen kann, muss der Nutzer eine Möglichkeit haben diese in <i>Ophelia</i> eingeben zu können. Dabei ist offen gehalten, ob eine Sprachsteuerung oder Gestensteuerung zur Minimierung der Touch- oder Tastenbedienung umgesetzt wird.
Wird umgesetzt	

ID	nav-fkt-slauteingg
Anforderungstyp	funktional
Name	Standardnavigationslautstärke Eingabe Geste
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Standardnavigationslautstärke über eine Geste eingeben zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die Absicht hat, seine über einensönlische Standardnavigationslautstärke in <i>Ophelia</i> einzugeben (Vorbedingung), stellt <i>Ophelia</i> einen Gesten-Befehl bereit, um die Standardnavigationslautstärke im Fahrzeug eingeben zu können (Nachbedingung).
Akzeptanzkriterium 2	Wenn der Nutzer den Gesten-Befehl ausführt (Aktion), wird die Standardnavigationslautstärke im Fahrzeug eingegeben (Nachbedingung).
Abgeleitet von	nav-fkt-slauteing
Ursprung	User Story Navigation 1
Referenzen	Standardlautstärke
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	Damit der Nutzer dem System eine Standardnavigationslautstärke mitteilen kann, muss der Nutzer eine Möglichkeit haben diese in <i>Ophelia</i> eingeben zu können. Dabei ist offen gehalten, ob eine Sprachsteuerung oder Gestensteuerung zur Minimierung der Touch- oder Tastenbedienung umgesetzt wird.
Wird umgesetzt	

ID	nav-fkt-slautaendern
Anforderungstyp	funktional
Name	Standardnavigationslautstärke ändern

Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Standardnavigationslautstärke im Fahrzeug ändern zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die Absicht hat, seine Standardnavigationslautstärke in <i>Ophelia</i> zu ändern (Vorbedingung), stellt <i>Ophelia</i> ein UI-Element bereit, um die Standardnavigationslautstärke im Fahrzeug ändern zu können (Nachbedingung).
Akzeptanzkriterium 2	Wenn eine Standardnavigationslautstärke in <i>Ophelia</i> eingegeben wurde (Vorbedingung), und der Nutzer mit dem UI-Element interagiert (Aktion), wird die Standardnavigationslautstärke im Fahrzeug verändert (Nachbedingung).
Abgeleitet von	
Ursprung	User Story Navigation 1
Referenzen	Standardlautstärke
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	Die Standardnavigationslautstärke soll verändert werden können, damit der Nutzer die Standardnavigationslautstärke an seine Präferenzen anpassen kann.
Wird umgesetzt	

ID	nav-fkt-slauterhoehenhmi
Anforderungstyp	funktional
Name	Standardnavigationslautstärke erhöhen HMI
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Standardnavigationslautstärke über ein HMI im Fahrzeug erhöhen zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die Absicht hat, seine Standardnavigationslautstärke in <i>Ophelia</i> zu erhöhen (Vorbedingung), stellt <i>Ophelia</i> ein HMI bereit, um die Standardnavigationslautstärke im Fahrzeug eingeben zu können (Nachbedingung).
Akzeptanzkriterium 2	Wenn eine Standardnavigationslautstärke in <i>Ophelia</i> eingegeben wurde (Vorbedingung), und der Nutzer mit dem HMI interagiert (Aktion), wird die Standardnavigationslautstärke erhöht (Nachbedingung).
Abgeleitet von	nav-fkt-slautaendern
Ursprung	User Story Navigation 1

Liste der nicht-funktionalen Anforderungen

Referenzen	Standardlautstärke
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	Durch eine Erhöhung der Standardnavigationslautstärke soll diese angepasst werden, falls sie für den Nutzer nicht laut genug ist. Falls der Nutzer die Sprach- oder Gestensteuerung nicht verwenden möchte steht die Verwendung eines HMI zur Verfügung.
Wird umgesetzt	

ID	nav-fkt-slauterhoehensprache
Anforderungstyp	funktional
Name	Standardnavigationslautstärke erhöhen Sprache
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Standardnavigationslautstärke über einen Sprachbefehl erhöhen zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die Absicht hat, seine über einensönliche Standardnavigationslautstärke in <i>Ophelia</i> zu erhöhen (Vorbedingung), stellt <i>Ophelia</i> einen Sprachbefehl bereit, um die Standardnavigationslautstärke im Fahrzeug erhöhen zu können (Nachbedingung).
Akzeptanzkriterium 2	Wenn eine Standardnavigationslautstärke in <i>Ophelia</i> eingegeben wurde (Vorbedingung), und der Nutzer den Sprachbefehl äußert (Aktion), wird die Standardnavigationslautstärke erhöht (Nachbedingung).
Abgeleitet von	nav-fkt-slautaendern
Ursprung	User Story Navigation 1
Referenzen	Standardlautstärke
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	Die Standardnavigationslautstärke soll über einen Sprachbefehl verändert werden können, um die Tasten- und Knopfbedienung im Fahrzeug zu vermindern.
Wird umgesetzt	

ID	nav-fkt-slauterhoehengeste
Anforderungstyp	funktional
Name	Standardnavigationslautstärke erhöhen Geste

Liste der nicht-funktionalen Anforderungen

Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Standardnavigationslautstärke im Fahrzeug über eine Geste erhöhen zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die Absicht hat, seine Standardnavigationslautstärke in <i>Ophelia</i> zu erhöhen (Vorbedingung), stellt <i>Ophelia</i> einen Gesten-Befehl bereit, um die Standardnavigationslautstärke im Fahrzeug erhöhen zu können (Nachbedingung).
Akzeptanzkriterium 2	Wenn eine Standardnavigationslautstärke in <i>Ophelia</i> eingegeben wurde (Vorbedingung), und der Nutzer den Gesten-Befehl ausführt (Aktion), wird die Standardnavigationslautstärke erhöht (Nachbedingung).
Abgeleitet von	nav-fkt-slautaendern
Ursprung	User Story Navigation 1
Referenzen	Standardlautstärke
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	Die Standardnavigationslautstärke soll über einen Gesten-Befehl verändert werden können, um die Tasten- und Knopfbedienung im Fahrzeug zu vermindern.
Wird umgesetzt	

ID	nav-fkt-slautverringernhmi
Anforderungstyp	funktional
Name	Standardnavigationslautstärke verringern HMI
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Standardnavigationslautstärke im Fahrzeug über ein HMI verringern zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die Absicht hat, seine über einensönliche Standardnavigationslautstärke in <i>Ophelia</i> zu verringern (Vorbedingung), stellt <i>Ophelia</i> ein HMI bereit, um die Standardnavigationslautstärke im Fahrzeug eingeben zu können (Nachbedingung).
Akzeptanzkriterium 2	Wenn eine Standardnavigationslautstärke in <i>Ophelia</i> eingegeben wurde (Vorbedingung), und der Nutzer mit dem HMI interagiert (Aktion), wird die Standardnavigationslautstärke verringert (Nachbedingung).
Abgeleitet von	nav-fkt-slautaendern
Ursprung	User Story Navigation 1
Referenzen	Standardlautstärke

Liste der nicht-funktionalen Anforderungen

Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	Die Standardnavigationslautstärke soll verringert werden können, falls die Standardnavigationslautstärke für den Nutzer zu laut ist.
Wird umgesetzt	

ID	nav-fkt-slautverringernsprache
Anforderungstyp	funktional
Name	Standardnavigationslautstärke verringern Sprache
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Standardnavigationslautstärke über einen Sprachbefehl verringern zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die Absicht hat, seine über einensönlische Standardnavigationslautstärke in <i>Ophelia</i> zu verringern (Vorbedingung), stellt <i>Ophelia</i> einen Sprachbefehl bereit, um die Standardnavigationslautstärke im Fahrzeug verringern zu können (Nachbedingung).
Akzeptanzkriterium 2	Wenn eine Standardnavigationslautstärke in <i>Ophelia</i> eingegeben wurde (Vorbedingung), und der Nutzer den Sprachbefehl äußert (Aktion), wird die Standardnavigationslautstärke verringert (Nachbedingung).
Abgeleitet von	nav-fkt-slautaendern
Ursprung	User Story Navigation 1
Referenzen	Standardlautstärke
Zuletzt geändert von	Daniel, 04.07.19
Status	offen
Begründung	Die Standardnavigationslautstärke soll über einen Sprachbefehl verändert werden können, um die Tasten- und Knopfbedienung im Fahrzeug zu vermindern.
Wird umgesetzt	

ID	nav-fkt-lautstaerkeverringergeste
Anforderungstyp	funktional
Name	Standardnavigationslautstärke verringern Geste
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Standardnavigationslautstärke über einen Geste verringern zu können.
Nachgewiesen durch	

Liste der nicht-funktionalen Anforderungen

Akzeptanzkriterium	Wenn der Nutzer die Absicht hat, seine Standardnavigationslautstärke in <i>Ophelia</i> zu verringern (Vorbedingung), stellt <i>Ophelia</i> einen Gesten-Befehl bereit, um die Standardnavigationslautstärke im Fahrzeug verringern zu können (Nachbedingung).
Akzeptanzkriterium 2	Wenn eine Standardnavigationslautstärke in <i>Ophelia</i> eingegeben wurde (Vorbedingung), und der Nutzer den Gesten-Befehl äußert (Aktion), wird die Standardnavigationslautstärke verringert (Nachbedingung).
Abgeleitet von	nav-fkt-slautaendern
Ursprung	User Story Navigation 1
Referenzen	Standardlautstärke
Zuletzt geändert von	Daniel, 04.07.19
Status	offen
Begründung	Die Standardnavigationslautstärke soll über einen Gesten-Befehl verändert werden können, um die Tasten- und Knopfbedienung im Fahrzeug zu vermindern.
Wird umgesetzt	

ID	nav-fkt-lautwerte
Anforderungstyp	funktional
Name	Navigationslautstärke Wertebereich
Anforderungssatz	Die Navigationslautstärke des <i>Ophelia</i> -Moduls muss natürliche Zahlen zwischen 0 und 30 im Fahrzeug annehmen.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer eine Navigationslautstärke im Fahrzeug eingeben möchte (Vorbedingung), kann <i>Ophelia</i> s Navigationslautstärke natürliche Zahlen zwischen 0 und 30 annehmen (Nachbedingung).
Abgeleitet von	nav-fkt-slauteinstellen
Ursprung	User Story Navigation 1
Referenzen	
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	Die Standardnavigationslautstärke soll einen definierten Wertebereich besitzen.
Wird umgesetzt	

ID	nav-fkt-lautzuruecksetzen
Anforderungstyp	funktional
Name	Navigationslautstärke zurücksetzen

Liste der nicht-funktionalen Anforderungen

Anforderungssatz	Das <i>Ophelia</i> -Modul muss die Navigationslautstärke bei Fahrtantritt zurücksetzen können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer eine Standardnavigationslautstärke in <i>Ophelia</i> hinterlegt hat (Vorbedingung) und die Navigationslautstärke im Fahrzeug während der Fahrt verändert wird (Aktion), ist die Standardnavigationslautstärke beim nächsten Fahrtantritt eingestellt (Nachbedingung).
Abgeleitet von	nav-fkt-slauteinstellen
Ursprung	User Story Navigation 2
Referenzen	
Zuletzt geändert von	Daniel, 04.07.19
Status	offen
Begründung	Wenn die Navigationslautstärke während der Fahrt stark erhöht wird, kann dies den Nutzer beim nächsten Fahrtantritt stören.
Wird umgesetzt	

ID	nav-fkt-slautspeichern
Anforderungstyp	funktional
Name	Standardnavigationslautstärke speichern
Anforderungssatz	Wenn der Nutzer eine Standardnavigationslautstärke im <i>Ophelia</i> -Modul eingegeben hat, muss das <i>Ophelia</i> -Modul die vom Nutzer eingegebene Standardnavigationslautstärke im Fahrzeug speichern können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer eine Standardnavigationslautstärke im Fahrzeug über eine Interaktion in <i>Ophelia</i> eingibt (Aktion), wird die Standardnavigationslautstärke im Fahrzeug gespeichert (Nachbedingung).
Abgeleitet von	nav-fkt-slauteinstellen
Ursprung	User Story Navigation 1
Referenzen	
Zuletzt geändert von	Daniel, 04.07.19
Status	offen
Begründung	Damit diese Einstellung automatisch vorgenommen werden kann, muss die Standardnavigationslautstärke in <i>Ophelia</i> gespeichert werden.
Wird umgesetzt	

Liste der nicht-funktionalen Anforderungen

ID	nav-fkt-lautaendern
Anforderungstyp	funktional
Name	Navigationslautstärke ändern
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die aktuelle Navigationslautstärke ändern zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die Absicht hat, die Navigationslautstärke in <i>Ophelia</i> zu ändern (Vorbedingung), stellt <i>Ophelia</i> ein UI-Element bereit, um die Navigationslautstärke im Fahrzeug ändern zu können (Nachbedingung).
Akzeptanzkriterium 2	Wenn der Nutzer mit dem UI-Element interagiert (Aktion), wird die Navigationslautstärke im Fahrzeug verändert (Nachbedingung).
Abgeleitet von	nav-fkt-slauteinstellen
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel, 04.07.19
Status	offen
Begründung	Die Navigationslautstärke soll verändert werden können, damit der Nutzer die Navigationslautstärke an seine Bedürfnisse anpassen kann.
Wird umgesetzt	

ID	nav-fkt-lauterhoehenhmi
Anforderungstyp	funktional
Name	Navigationslautstärke erhöhen HMI
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die aktuelle Navigationslautstärke über ein HMI erhöhen zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die Absicht hat, die Navigationslautstärke in <i>Ophelia</i> zu erhöhen (Vorbedingung), stellt <i>Ophelia</i> ein HMI bereit, um die Navigationslautstärke im Fahrzeug erhöhen zu können (Nachbedingung).
Akzeptanzkriterium 2	Wenn der Nutzer mit dem HMI interagiert (Aktion), wird die Navigationslautstärke erhöht (Nachbedingung).
Abgeleitet von	nav-fkt-lautaendern
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel, 09.07.19
Status	offen

Liste der nicht-funktionalen Anforderungen

Begründung	Die Navigationslautstärke soll über ein HMI verändert werden können, falls eine Gesten- oder Sprachsteuerung vom Nutzer nicht verwendet wird.
Wird umgesetzt	

ID	nav-fkt-lauterhoehensprache
Anforderungstyp	funktional
Name	Navigationslautstärke erhöhen Sprache
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die aktuelle Navigationslautstärke über einen Sprachbefehl erhöhen zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die Absicht hat, die Navigationslautstärke in <i>Ophelia</i> zu erhöhen (Vorbedingung), stellt <i>Ophelia</i> einen Sprachbefehl bereit, um die Navigationslautstärke im Fahrzeug erhöhen zu können (Nachbedingung).
Akzeptanzkriterium 2	Wenn der Nutzer den Sprachbefehl äußert (Aktion), wird die Standardnavigationslautstärke erhöht (Nachbedingung).
Abgeleitet von	nav-fkt-lautaendern
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	Die Navigationslautstärke soll über einen Sprachbefehl verändert werden können, um die Tasten- und Knopfbeldienung im Fahrzeug zu vermindern.
Wird umgesetzt	

ID	nav-fkt-lauterhoehengeste
Anforderungstyp	funktional
Name	Navigationslautstärke erhöhen Geste
Anforderungssatz	<i>Ophelia</i> muss dem Nutzer die Möglichkeit bieten, die aktuelle Navigationslautstärke über einen Gesten-Befehl erhöhen zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die Absicht hat, die Navigationslautstärke in <i>Ophelia</i> zu erhöhen (Vorbedingung), stellt <i>Ophelia</i> einen Gesten-Befehl bereit, um die Standardnavigationslautstärke im Fahrzeug erhöhen zu können (Nachbedingung).

Liste der nicht-funktionalen Anforderungen

Akzeptanzkriterium 2	Wenn der Nutzer den Gesten-Befehl ausführt (Aktion), wird die Navigationslautstärke erhöht (Nachbedingung).
Abgeleitet von	nav-fkt-lautaendern
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	Die Navigationslautstärke soll über einen Gesten-Befehl verändert werden können, um die Tasten- und Knopfbedienung im Fahrzeug zu vermindern.
Wird umgesetzt	

ID	nav-fkt-lautverringernhmi
Anforderungstyp	funktional
Name	Navigationslautstärke verringern Sprache
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die aktuelle Navigationslautstärke über einen Sprachbefehl verringern zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die Absicht hat, die Navigationslautstärke in <i>Ophelia</i> zu verringern (Vorbedingung), stellt <i>Ophelia</i> einen Sprachbefehl bereit, um die Standardnavigationslautstärke im Fahrzeug verringern zu können (Nachbedingung).
Akzeptanzkriterium 2	Wenn der Nutzer den Sprachbefehl äußert (Aktion), wird die Navigationslautstärke verringert (Nachbedingung).
Abgeleitet von	nav-fkt-lautaendern
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	Die Navigationslautstärke soll über einen Sprachbefehl verändert werden können, um die Tasten- und Knopfbedienung im Fahrzeug zu vermindern.
Wird umgesetzt	

ID	nav-fkt-lautstaerkeverringerngeste
Anforderungstyp	funktional
Name	Navigationslautstärke verringern Geste

Liste der nicht-funktionalen Anforderungen

Anforderungssatz	<i>Ophelia</i> muss dem Nutzer die Möglichkeit bieten, die aktuelle Navigationslautstärke über einen Gesten-Befehl verringern zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die Absicht hat, die Navigationslautstärke in <i>Ophelia</i> zu verringern (Vorbedingung), stellt <i>Ophelia</i> einen Gesten-Befehl bereit, um die Navigationslautstärke im Fahrzeug verringern zu können (Nachbedingung).
Akzeptanzkriterium 2	Wenn der Nutzer den Gesten-Befehl äußert (Aktion), wird die Navigationslautstärke verringert (Nachbedingung).
Abgeleitet von	nav-fkt-lautaendern
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel, 04.07.19
Status	offen
Begründung	Die Navigationslautstärke soll über einen Gesten-Befehl verändert werden können, um die Tasten- und Knopfbedienung im Fahrzeug zu vermindern.
Wird umgesetzt	

ID	nav-fkt-lautstaerkeaktorik
Anforderungstyp	funktional
Name	Navigationslautstärke Aktorik
Anforderungssatz	Das <i>Ophelia</i> -Modul muss die Aktorik für die Audioausgabe im Fahrzeug steuern können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	nav-fkt-slauteinstellen
Ursprung	User Story Navigation 2
Referenzen	
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	Damit die Standardnavigationslautstärke im Fahrzeug einstellt werden kann, muss die benötigte Aktorik gesteuert werden können.
Wird umgesetzt	

ID	nav-fkt-zielvor
Anforderungstyp	funktional

Name	Navigationsziel einstellen
Anforderungssatz	Das <i>Ophelia</i> -Modul muss das vom Nutzer im Nutzerprofil als aktiv ausgewählte Navigationsziel vor Fahrtantritt einstellen können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	sys-fkt-vorein
Ursprung	User Story Navigation 5
Referenzen	
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	Damit der Nutzer dies nicht selbst im Fahrzeug einstellen muss.
Wird umgesetzt	

ID	nav-fkt-zieleingabe
Anforderungstyp	funktional
Name	Navigationsziel eingeben
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, ein Navigationsziel im Fahrzeug eingeben zu können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	nav-fkt-zielvor
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	Der Nutzer muss <i>Ophelia</i> ein Navigationsziel mitteilen können.
Wird umgesetzt	

ID	nav-fkt-tpeinstellen
Anforderungstyp	funktional
Name	Tagesprogramm einstellen
Anforderungssatz	Das <i>Ophelia</i> -Modul muss das vom Nutzer im Nutzerprofil als aktiv ausgewählte Tagesprogramm vor Fahrtantritt als Route im Fahrzeug einstellen können.
Nachgewiesen durch	

Liste der nicht-funktionalen Anforderungen

Akzeptanzkriterium	Wenn der Nutzer eine Tagesprogramm in <i>Ophelia</i> gespeichert hat (Vorbedingung) und das Fahrzeug startet (Aktion), ist das vom Nutzer gespeicherte Tagesprogramm als Route im Fahrzeug eingestellt (Nachbedingung).
Abgeleitet von	sys-fkt-vorein
Ursprung	User Story Navigation 5
Referenzen	
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	<i>Ophelia.</i>
Wird umgesetzt	

ID	nav-fkt-zwischenzieleerinnern
Anforderungstyp	funktional
Name	Erinnerung an Zwischenziele
Anforderungssatz	Wenn die verwendeten Zwischenziele häufig angefahren werden, muss das <i>Ophelia</i> -Modul den Nutzer bei Fahrtantritt an die verwendeten Zwischenziele erinnern können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer Zwischenziele nach den Bedingungen anfährt (Vorbedingung), erfolgt beim nächsten Fahrtantritt eine Erinnerung an diese (Nachbedingung).
Abgeleitet von	sys-fkt-erinnern
Ursprung	User Story Navigation 4
Referenzen	Zwischenziel
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	
Wird umgesetzt	

ID	nav-fkt-zwischenzieleerfassen
Anforderungstyp	funktional
Name	Erfassung von Zwischenzielen
Anforderungssatz	Das <i>Ophelia</i> -Modul muss Zwischenziele aus Routen des Nutzers erfassen können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer Zwischenziele zu einer Route angibt (Vorbedingung) und diese Route in <i>Ophelia</i> speichert (Aktion), speichert <i>Ophelia</i> die Zwischenziele der Route als Zwischenziele (Nachbedingung).
Abgeleitet von	nav-fkt-zwischenzieleerinnern

Ursprung	User Story Navigation 4
Referenzen	Zwischenziel im Glossar
Zuletzt geändert von	Daniel, 10.07.19
Status	offen
Begründung	Damit Das <i>Ophelia</i> -Modul den Nutzer an Zwischenziele erinnern kann, müssen Zwischenziele aus Routen erfasst werden.
Wird umgesetzt	

ID	nav-fkt-zwischenzielespeichern
Anforderungstyp	funktional
Name	Speichern von Zwischenzielen
Anforderungssatz	Das <i>Ophelia</i> -Modul muss verwendete Zwischenziele der Routen des Nutzers im Nutzerprofil speichern können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer Zwischenziele zu einer Route angibt (Vorbedingung) und diese Route in <i>Ophelia</i> speichert (Aktion), muss <i>Ophelia</i> die Zwischenziele der Route als Zwischenziele speichern (Nachbedingung).
Abgeleitet von	nav-fkt-zwischenzieleerfassen
Ursprung	User Story Navigation 4
Referenzen	Zwischenziel im Glossar
Zuletzt geändert von	Daniel, 10.07.19
Status	offen
Begründung	Da Zwischenziele nutzerbezogene Daten sind, werden sie im Nutzerprofil gespeichert.
Wird umgesetzt	

ID	nav-fkt-zwischenzieleanserver
Anforderungstyp	funktional
Name	Zwischenziele an Server
Anforderungssatz	Das <i>Ophelia</i> -Modul muss verwendete Zwischenziele der Routen des Nutzers an den Server senden können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn eine Verbindungsmöglichkeit zwischen dem Fahrzeug und dem Server besteht und Zwischenziele im Fahrzeug gespeichert sind, die noch nicht auf dem Server vorhanden sind (Vorbedingung), und das Fahrzeug die Zwischenziele an den Server sendet (Aktion), müssen die Zwischenziele auf dem Server vorhanden sein.
Abgeleitet von	nav-fkt-zwischenzielespeichern

Liste der nicht-funktionalen Anforderungen

Ursprung	User Story Navigation 4
Referenzen	Zwischenziel im Glossar
Zuletzt geändert von	Daniel, 04.07.19
Status	offen
Begründung	Da Nutzerprofile auf dem Server gespeichert werden, müssen nutzerbezogene Daten an den Server gesendet werden.
Wird umgesetzt	

ID	nav-fkt-routenoptimierung
Anforderungstyp	funktional
Name	Routenoptimierung
Anforderungssatz	Das <i>Ophelia</i> -Modul muss die Route der Nutzer bezüglich der Ankunftszeit optimieren.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer ein/mehrere Ziel(e) in <i>Ophelia</i> eingegeben hat (Vorbedingung) und die Absicht hat, diese(s) Ziel(e) anzufahren (Aktion), ist die dadurch entstandene Route bezüglich der Ankunftszeit optimiert (Nachbedingung).
Abgeleitet von	sys-fkt-navfunk
Ursprung	User Story Navigation 7
Referenzen	Zwischenziel im Glossar
Zuletzt geändert von	Daniel, 10.07.19
Status	offen
Begründung	Um schnellstmöglich am Ziel anzukommen soll die schnellste Route gewählt werden.
Wird umgesetzt	

ID	nav-fkt-gegenstaendeerinnern
Anforderungstyp	funktional
Name	Erinnerung an Gegenstände
Anforderungssatz	Das <i>Ophelia</i> -Modul muss den Nutzer an angegebene Gegenstände bezüglich eines Ziels erinnern können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer Gegenstände bezüglich eines Ziels angegeben hat (Vorbedingung) und sich ins Fahrzeug setzt, um dieses Ziel anzufahren (Aktion), erfolgt eine Erinnerung an die Gegenstände im Fahrzeug (Nachbedingung).
Abgeleitet von	
Ursprung	User Story Navigation 8
Referenzen	

Liste der nicht-funktionalen Anforderungen

Zuletzt geändert von	Daniel, 30.06.19
Status	offen
Begründung	Im Rahmen des Kontexts „Helping“ soll vom System eine Erinnerungsfunktion an Gegenstände angeboten werden.
Wird umgesetzt	

ID	nav-fkt-tagesprogrammerinnern
Anforderungstyp	funktional
Name	Erinnerung an Tagesprogramme
Anforderungssatz	Das <i>Ophelia</i> -Modul muss den Nutzer an angegebene Tagesprogramme erinnern können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer Gegenstände bezüglich eines Ziels angegeben hat (Vorbedingung) und sich ins Fahrzeug setzt, um dieses Ziel anzufahren (Aktion), erfolgt eine Erinnerung an die Gegenstände im Fahrzeug (Nachbedingung).
Abgeleitet von	
Ursprung	User Story Navigation 6
Referenzen	
Zuletzt geändert von	Daniel, 30.06.19
Status	offen
Begründung	Im Rahmen des Kontexts „Helping“ soll vom System eine Erinnerungsfunktion an Gegenstände angeboten werden.
Wird umgesetzt	

ID	nav-fkt-heimateingabe
Anforderungstyp	funktional
Name	Heimatadresse Eingabe
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Heimatadresse in <i>Ophelia</i> eingeben zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Vorbedingungen: Der Nutzer soll Das <i>Ophelia</i> -Modul beauftragen die Heimatadresse eingeben zu können. Nachbedingungen: Das <i>Ophelia</i> -Modul speichert die Heimatadresse vorübergehend in <i>Ophelia</i> ab.
Abgeleitet von	
Ursprung	
Referenzen	
Zuletzt geändert von	Artem, 29.06.19
Status	offen

Liste der nicht-funktionalen Anforderungen

Begründung	Damit der Nutzer Das <i>Ophelia</i> -Modul die Heimatadresse mitteilen kann, muss er eine Möglichkeit haben diese in <i>Ophelia</i> eingeben zu können.
Wird umgesetzt	

ID	nav-fkt-heimatsave
Anforderungstyp	funktional
Name	Heimatadresse speichern
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Heimatadresse in <i>Ophelia</i> speichern zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Vorbedingungen: Der Nutzer soll Das <i>Ophelia</i> -Modul beauftragen die Heimatadresse zu speichern. Nachbedingungen: Das <i>Ophelia</i> -Modul speichert die Heimatadresse in <i>Ophelia</i> ab.
Abgeleitet von	
Ursprung	
Referenzen	
Zuletzt geändert von	Artem, 29.06.19
Status	offen
Begründung	Damit Das <i>Ophelia</i> -Modul die Heimatadresse für den Nutzer speichern kann, muss der Nutzer eine Möglichkeit haben diese in <i>Ophelia</i> speichern zu können.
Wird umgesetzt	

ID	nav-fkt-heimatchange
Anforderungstyp	funktional
Name	Heimatadresse ändern
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Heimatadresse in <i>Ophelia</i> ändern zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Vorbedingungen: Der Nutzer soll Das <i>Ophelia</i> -Modul beauftragen die Heimatadresse zu ändern. Nachbedingungen: Das <i>Ophelia</i> -Modul ändert die Heimatadresse in <i>Ophelia</i> ab.
Abgeleitet von	
Ursprung	
Referenzen	
Zuletzt geändert von	Artem, 29.06.19
Status	offen

Liste der nicht-funktionalen Anforderungen

Begründung	Damit Das <i>Ophelia</i> -Modul die Heimatadresse für den Nutzer ändern kann, muss der Nutzer eine Möglichkeit haben diese in <i>Ophelia</i> ändern zu können.
Wird umgesetzt	

ID	nav-fkt-heimatdel
Anforderungstyp	funktional
Name	Heimatadresse löschen
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Heimatadresse löschen zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Vorbedingungen: Der Nutzer soll Das <i>Ophelia</i> -Modul beauftragen die Heimatadresse zu löschen. Nachbedingungen: Das <i>Ophelia</i> -Modul löscht die Heimatadresse in <i>Ophelia</i> .
Abgeleitet von	
Ursprung	
Referenzen	
Zuletzt geändert von	Artem, 29.06.19
Status	offen
Begründung	Damit Das <i>Ophelia</i> -Modul die Heimatadresse für den Nutzer löschen kann, muss der Nutzer eine Möglichkeit haben diese in <i>Ophelia</i> löschen zu können.
Wird umgesetzt	

ID	nav-fkt-heimatziel
Anforderungstyp	funktional
Name	Heimatadresse als Zwischenziel auswählen
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Heimatadresse als Zwischenziel auswählen zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Vorbedingungen: Der Nutzer soll Das <i>Ophelia</i> -Modul beauftragen die Heimatadresse als Zwischenziel auswählen zu können. Nachbedingungen: Das <i>Ophelia</i> -Modul wählt die Heimatadresse als Zwischenziel.
Abgeleitet von	
Ursprung	
Referenzen	
Zuletzt geändert von	Artem, 29.06.19
Status	offen

Liste der nicht-funktionalen Anforderungen

Begründung	Damit Das <i>Ophelia</i> -Modul die Heimatadresse für den Nutzer als Zwischenziel speichern kann, muss der Nutzer eine Möglichkeit haben diese in <i>Ophelia</i> auswählen zu können.
Wird umgesetzt	

ID	nav-fkt-heimatroutedel
Anforderungstyp	funktional
Name	Heimatadresse aus Route entfernen
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Heimatadresse aus der Route zu entfernen.
Nachgewiesen durch	
Akzeptanzkriterium	Vorbedingungen: Der Nutzer soll Das <i>Ophelia</i> -Modul beauftragen die Heimatadresse aus der Route zu entfernen. Nachbedingungen: Das <i>Ophelia</i> -Modul entfernt die Heimatadresse aus der Route.
Abgeleitet von	
Ursprung	
Referenzen	
Zuletzt geändert von	Artem, 29.06.19
Status	offen
Begründung	Damit Das <i>Ophelia</i> -Modul die Heimatadresse für den Nutzer aus der Route entfernen kann, muss der Nutzer eine Möglichkeit haben diese in <i>Ophelia</i> entfernen zu können.
Wird umgesetzt	

ID	nav-fkt-routeverlaufwahl
Anforderungstyp	funktional
Name	Route aus Routenverlauf wählen
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Route aus dem Routenverlauf wählen zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Vorbedingungen: Der Nutzer soll Das <i>Ophelia</i> -Modul beauftragen die Route aus Routenverlauf zu wählen. Nachbedingungen: Das <i>Ophelia</i> -Modul wählt die Route aus dem Routenverlauf.
Abgeleitet von	
Ursprung	
Referenzen	
Zuletzt geändert von	Artem, 29.06.19

Liste der nicht-funktionalen Anforderungen

Status	offen
Begründung	Damit Das <i>Ophelia</i> -Modul die Route aus Routenverlauf für den Nutzer wählen kann, muss der Nutzer eine Möglichkeit haben diese in <i>Ophelia</i> auswählen zu können.
Wird umgesetzt	

ID	nav-fkt-routezieleing
Anforderungstyp	funktional
Name	Zwischenziele der Route eingeben
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Zwischenziele der Route eingeben zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Vorbedingungen: Der Nutzer soll Das <i>Ophelia</i> -Modul beauftragen die Zwischenziele der Route hinzuzufügen. Nachbedingungen: Das <i>Ophelia</i> -Modul fügt die Zwischenziele der Route hinzu.
Abgeleitet von	
Ursprung	
Referenzen	
Zuletzt geändert von	Artem, 29.06.19
Status	offen
Begründung	Damit Das <i>Ophelia</i> -Modul die Zwischenziele der Route für den Nutzer hinzufügen kann, muss der Nutzer eine Möglichkeit haben diese in <i>Ophelia</i> eingeben zu können.
Wird umgesetzt	

ID	nav-fkt-routezielchange
Anforderungstyp	funktional
Name	Zwischenziele der Route ändern
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Zwischenziele der Route ändern zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Vorbedingungen: Der Nutzer soll Das <i>Ophelia</i> -Modul beauftragen die Zwischenziele der Route zu ändern. Nachbedingungen: Das <i>Ophelia</i> -Modul ändert die Zwischenziele der Route.
Abgeleitet von	
Ursprung	
Referenzen	
Zuletzt geändert von	Artem, 29.06.19
Status	offen

Liste der nicht-funktionalen Anforderungen

Begründung	Damit Das <i>Ophelia</i> -Modul die Zwischenziele der Route für den Nutzer ändern kann, muss der Nutzer eine Möglichkeit haben diese in <i>Ophelia</i> ändern zu können.
Wird umgesetzt	

ID	nav-fkt-routezielde1
Anforderungstyp	funktional
Name	Zwischenziele der Route löschen
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Zwischenziele der Route löschen zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Vorbedingungen: Der Nutzer soll Das <i>Ophelia</i> -Modul beauftragen die Zwischenziele der Route zu löschen. Nachbedingungen: Das <i>Ophelia</i> -Modul löscht die Zwischenziele der Route.
Abgeleitet von	
Ursprung	
Referenzen	
Zuletzt geändert von	Artem, 29.06.19
Status	offen
Begründung	Damit Das <i>Ophelia</i> -Modul die Zwischenziele der Route für den Nutzer löschen kann, muss der Nutzer eine Möglichkeit haben diese in <i>Ophelia</i> löschen zu können.
Wird umgesetzt	

ID	nav-fkt-routezielsave
Anforderungstyp	funktional
Name	Zwischenziele der Route speichern
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Zwischenziele der Route speichern zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Vorbedingungen: Der Nutzer soll Das <i>Ophelia</i> -Modul beauftragen die Zwischenziele der Route zu speichern. Nachbedingungen: Das <i>Ophelia</i> -Modul speichert die Zwischenziele der Route.
Abgeleitet von	
Ursprung	
Referenzen	
Zuletzt geändert von	Artem, 29.06.19
Status	offen

Liste der nicht-funktionalen Anforderungen

Begründung	Damit Das <i>Ophelia</i> -Modul die Zwischenziele der Route für den Nutzer speichern kann, muss der Nutzer eine Möglichkeit haben diese in <i>Ophelia</i> speichern zu können.
Wird umgesetzt	

ID	nav-fkt-routezielchange
Anforderungstyp	funktional
Name	Zwischenziele aus der Route entfernen
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, die Zwischenziele aus der Route entfernen zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Vorbedingungen: Der Nutzer soll Das <i>Ophelia</i> -Modul beauftragen die Zwischenziele aus der Route zu entfernen. Nachbedingungen: Das <i>Ophelia</i> -Modul entfernt die Zwischenziele aus der Route.
Abgeleitet von	
Ursprung	
Referenzen	
Zuletzt geändert von	Artem, 29.06.19
Status	offen
Begründung	Damit Das <i>Ophelia</i> -Modul die Zwischenziele aus der Route für den Nutzer entfernen kann, muss der Nutzer eine Möglichkeit haben diese in <i>Ophelia</i> entfernen zu können.
Wird umgesetzt	

ID	nav-fkt-favoritadresse
Anforderungstyp	funktional
Name	Favoritenadressen
Anforderungssatz	Das <i>Ophelia</i> -Modul soll dem Nutzer die Möglichkeit bieten, Favoritenadressen benutzen zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer das Navigationsmenü des Systems öffnet (Aktion), können Favoritenadressen verwendet werden (Nachbedingung).
Abgeleitet von	
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel 1.07.19
Status	offen
Begründung	Um schnell auf bestimmte Adressen zugreifen zu können.

Liste der nicht-funktionalen Anforderungen

ID	nav-fkt-favoritadresseerstellen
Anforderungstyp	funktional
Name	Favoritenadresse erstellen
Anforderungssatz	Das <i>Ophelia</i> -Modul soll dem Nutzer die Möglichkeit bieten, eine Favoritenadresse erstellen zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer im Navigationsmenü Favoritenadressen auswählt (Aktion), stellt <i>Ophelia</i> ein UI-Element bereit, um eine Favoritenadresse erstellen zu können (Nachbedingung).
Abgeleitet von	nav-fkt-favoritadresse
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel 01.07.19
Status	offen
Begründung	Damit der Nutzer selbst eine Favoritenadresse erstellen kann.

ID	nav-fkt-favoritaddressenspeichern
Anforderungstyp	funktional
Name	Favoritenadresse speichern
Anforderungssatz	Das <i>Ophelia</i> -Modul muss die Favoritenadressen der Nutzer speichern können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer Daten einer Favoritenadresse (Postleitzahl, Straße, Hausnummer) eingibt (Vorbedingung), und diese in <i>Ophelia</i> speichert (Aktion), speichert <i>Ophelia</i> die Favoritenadresse im Nutzerprofil (Nachbedingung).
Abgeleitet von	
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel 01.07.19
Status	offen
Begründung	Damit ein Nutzer zu einem späteren Zeitpunkt auf die eingegebene Favoritenadresse zugreifen kann, muss sie in <i>Ophelia</i> gespeichert werden.

ID	nav-fkt-zielalsfavoritwaehlen
Anforderungstyp	funktional
Name	Ziel als Favoriten wählen

Anforderungssatz	Das <i>Ophelia</i> -Modul soll ein Ziel als Favoritenadresse auswählen können.
Nachgewiesen durch	
Akzeptanzkriterium	Der Nutzer möchte eine Adresse als Favorit auswählen(Vorbedingung). Der Nutzer wählt eine Adresse als Favorit aus(Aktion), dann muss <i>Ophelia</i> die vom Nutzer ausgewählte Adresse als Favorit speichern (Nachbedingung).
Abgeleitet von	
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel 01.07.19
Status	offen
Begründung	Um eine vorgegebene Adresse als Favoritadresse auszuwählen und speichern zu können.

ID	nav-fkt-favoritadressealszielwaehlen
Anforderungstyp	funktional
Name	Favoritenadresse als Ziel wählen
Anforderungssatz	Das <i>Ophelia</i> -Modul soll eine Favoritenadresse als Navigationsziel auswählen können.
Nachgewiesen durch	
Akzeptanzkriterium	Der Nutzer möchte eine Favoritenadresse auswählen(Vorbedingung). Wenn der Nutzer eine FavoritenAdresse auswählt(Aktion), muss <i>Ophelia</i> die vom Nutzer ausgewählte Adresse als Zwischenziel zur aktuellen Route hinzufügen (Nachbedingung).
Abgeleitet von	nav-fkt-favoritadresse
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel 01.07.19
Status	offen
Begründung	Damit Favoritenadressen als Ziele angefahren werden können.

ID	nav-fkt-favoritadresseloeschen
Anforderungstyp	funktional
Name	Favoritenadresse löschen
Anforderungssatz	Das <i>Ophelia</i> -Modul soll eine Favoritenadresse löschen können.
Nachgewiesen durch	

Liste der nicht-funktionalen Anforderungen

Akzeptanzkriterium	Wenn der Nutzer die Liste der Favoritenadressen geöffnet hat (Vorbedingung) und eine Favoritenadresse aus der Liste der Favoriten löscht (Aktion), wird die Favoritenadresse im Nutzerprofil gelöscht (Nachbedingung).
Abgeleitet von	nav-fkt-favoritadresse
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel 01.07.19
Status	offen
Begründung	Einem Nutzer soll die Möglichkeit geboten werden, die Favoritenadresse zu löschen.

ID	nav-fkt-parkplatzortspeichern
Anforderungstyp	funktional
Name	Letzten Parkplatz speichern
Anforderungssatz	Das <i>Ophelia</i> -Modul soll den Standort des letzten Parkplatzes speichern können, an dem das Fahrzeug abgestellt wurde.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn das Fahrzeug vom Fahrer an einem Parkplatz abgestellt und verriegelt wurde (Aktion) speichert <i>Ophelia</i> den Standort des Fahrzeugs im Nutzerprofil (Nachbedingung).
Abgeleitet von	nav-fkt-parkplatzortabrufen
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel 01.07.19
Status	offen
Begründung	Um das geparkte Fahrzeug leichter zu finden soll <i>Ophelia</i> den Standort speichern, damit der Nutzer darauf zugreifen kann.

ID	nav-fkt-routenoptimierungverkehrsmeldung
Anforderungstyp	funktional
Name	Verkehrsmeldungen zur Routenoptimierung
Anforderungssatz	Das <i>Ophelia</i> -Modul soll den Nutzer über Verkehrsmeldungen für aktuelle Route informieren können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer auf einer Route fährt (Vorbedingung), und eine Verkehrsmeldung für die Route über das Radio ausgegeben wird (Aktion), wird der Nutzer über die Verkehrsmeldung informiert (Nachbedingung).

Liste der nicht-funktionalen Anforderungen

Abgeleitet von	
Ursprung	
Referenzen	
Zuletzt geändert von	MJ 04.07.19
Status	offen
Begründung	Durch die Analyse der Verkehrsmeldungen kann Das <i>Ophelia</i> -Modul die Route anhand der aktuellen Verkehrssituation optimieren und planen. In besonderen Fällen ist es auch möglich, den Benutzer vor dem vorherigen Plan über die Abreise zu informieren.

ID	nav-fkt-routenoptimierungdynamischeanpassung
Anforderungstyp	funktional
Name	Dynamische Anpassung der Routen
Anforderungssatz	Das <i>Ophelia</i> -Modul soll die aktuelle Route dynamisch nach der Verkehrssituation anpassen können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer auf einer Route fährt (Vorbedingung), und eine Verkehrsmeldung für die Route über das Radio ausgegeben wird (Aktion), prüft <i>Ophelia</i> eine Anpassung der Route (Nachbedingung).
Abgeleitet von	
Ursprung	
Referenzen	
Zuletzt geändert von	MJ 04.07.19
Status	offen
Begründung	by analyzing the traffic news Das <i>Ophelia</i> -Modul can optimized and plan the route based on the latest traffic situation. it is also possible in special cases to inform user to departure earlier than what planed previously.

ID	nav-fkt-lautstaerkeanpassenpraef
Anforderungstyp	funktional
Name	Standardnavigationslautstärke an Präferenzen anpassen
Anforderungssatz	Das <i>Ophelia</i> -Modul muss die Standardnavigationslautstärke den Präferenzen des Nutzers entsprechend anpassen.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	sys-fkt-lern
Ursprung	User Story Navigation 3
Referenzen	

Liste der nicht-funktionalen Anforderungen

Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	Falls sich die Standardnavigationslautstärke nach den Präferenzen des Nutzers ändern sollte, muss diese angepasst werden.
Wird umgesetzt	

ID	nav-fkt-lautstaerkeeinstellen
Anforderungstyp	funktional
Name	Navigationslautstärke einstellen
Anforderungssatz	Das <i>Ophelia</i> -Modul muss die Navigationslautstärke im Fahrzeug selbstständig einstellen können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die Absicht hat, seine über einensönlche Standardnavigationslautstärke in <i>Ophelia</i> zu setzen (Vorbedingung), besteht die Möglichkeit, die Standardnavigationslautstärke im Fahrzeug eingeben zu können (Nachbedingung).
Abgeleitet von	nav-fkt-lautstaerkeeinstellen
Ursprung	User Story Navigation 1
Referenzen	
Zuletzt geändert von	Daniel, 04.07.19
Status	offen
Begründung	Damit der Nutzer dem System eine Standardnavigationslautstärke mitteilen kann, muss er eine Möglichkeit haben diese in <i>Ophelia</i> eingeben zu können.
Wird umgesetzt	

ID	nav-fkt-lautstaerkeanpassensit
Anforderungstyp	funktional
Name	Navigationslautstärke an Situation anpassen
Anforderungssatz	Das <i>Ophelia</i> -Modul muss die Navigationslautstärke der aktuellen Situation entsprechend anpassen.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	sys-fkt-lern
Ursprung	User Story Navigation 3
Referenzen	
Zuletzt geändert von	Daniel, 02.07.19
Status	offen

Liste der nicht-funktionalen Anforderungen

Begründung	Je nach unterschiedlich herrschender Lautstärke der Umgebung des Fahrers muss die Navigationslautstärke der aktuellen Situation entsprechend angepasst werden.
Wird umgesetzt	

ID	nav-fkt-lautstaerkeanpasseninnen
Anforderungstyp	funktional
Name	Navigationslautstärke anpassen innerhalb
Anforderungssatz	Das <i>Ophelia</i> -Modul muss die Navigationslautstärke der aktuellen Lautstärke im Fahrzeuginnen entsprechend anpassen.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	nav-fkt-lautstaerkeanpassensit
Ursprung	User Story Navigation 3
Referenzen	
Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	Je nach unterschiedlich herrschender Lautstärke der Umgebung des Fahrers muss die Navigationslautstärke der aktuellen Situation entsprechend angepasst werden.
Wird umgesetzt	

ID	nav-fkt-lautstaerkeanpasseninnenmusik
Anforderungstyp	funktional
Name	Navigationslautstärke anpassen Musik
Anforderungssatz	Das <i>Ophelia</i> -Modul muss die Navigationslautstärke der aktuellen Musiklautstärke entsprechend anpassen.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	nav-fkt-lautstaerkeanpasseninnen
Ursprung	User Story Navigation 3
Referenzen	
Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	Je nach unterschiedlich herrschender Lautstärke der Umgebung des Fahrers muss die Navigationslautstärke der aktuellen Musiklautstärke entsprechend angepasst werden.
Wird umgesetzt	

Liste der nicht-funktionalen Anforderungen

ID	nav-fkt-lautstaerkeanpasseninnenradio
Anforderungstyp	funktional
Name	Navigationslautstärke anpassen Radio
Anforderungssatz	Das <i>Ophelia</i> -Modul muss die Navigationslautstärke der aktuellen Radiolautstärke entsprechend anpassen.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	nav-fkt-lautstaerkeanpasseninnen
Ursprung	User Story Navigation 3
Referenzen	
Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	Je nach unterschiedlich herrschender Lautstärke der Umgebung des Fahrers muss die Navigationslautstärke der aktuellen Radiolautstärke entsprechend angepasst werden.
Wird umgesetzt	

ID	nav-fkt-lautstaerkesensoren
Anforderungstyp	funktional
Name	Lautstärkesensoren
Anforderungssatz	Das <i>Ophelia</i> -Modul muss Lautstärkesensoren auslesen können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	nav-fkt-lautstaerkeanpassensit
Ursprung	User Story Navigation 3
Referenzen	
Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	Damit die Navigationslautstärke an andere Geräuschquellen im Fahrzeug angepasst werden kann muss die Lautstärke der Geräuschquellen ermittelt werden.
Wird umgesetzt	

ID	nav-fkt-vorsziele
Anforderungstyp	funktional
Name	Navigationsvorschläge Ziele
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer basierend auf gelernten Zielen Navigationsvorschläge geben können.
Nachgewiesen durch	
Akzeptanzkriterium	

Abgeleitet von	sys-fkt-vorseins
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel, 10.07.19
Status	offen
Begründung	
Wird umgesetzt	

ID	nav-fkt-vorsrouten
Anforderungstyp	funktional
Name	Navigationsvorschläge Routen
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer basierend auf gelernten Zielen und Abfahrtszeiten Routen vorschlagen können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	sys-fkt-vorseins
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	
Wird umgesetzt	

ID	nav-fkt-vorsrouteeinstellen
Anforderungstyp	funktional
Name	Navigationsvorschlag Route einstellen
Anforderungssatz	Wenn ein Routenvorschlag vom Nutzer angenommen wird, muss Das <i>Ophelia</i> -Modul die vorgeschlagene Route als aktuelle Route setzen können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	nav-fkt-vorsrouten
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel, 10.07.19
Status	offen
Begründung	
Wird umgesetzt	

Liste der nicht-funktionalen Anforderungen

ID	nav-fkt-vorsroutezieleerinn
Anforderungstyp	funktional
Name	Navigationsvorschlag Route Erinnerung Zwischenziele
Anforderungssatz	Wenn ein Routenvorschlag vom Nutzer angenommen wird, muss Das <i>Ophelia</i> -Modul den Nutzer an die Zwischenziele der Route erinnern können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	nav-fkt-vorsrouten
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	
Wird umgesetzt	

ID	nav-fkt-vorsrouteablehn
Anforderungstyp	funktional
Name	Navigationsvorschlag Route Ablehnung
Anforderungssatz	Wenn ein Routenvorschlag vom Nutzer abgelehnt wird, muss Das <i>Ophelia</i> -Modul den Vorschlagsdialog beenden.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	nav-fkt-vorsrouten
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	
Wird umgesetzt	

ID	nav-fkt-vorsannehm
Anforderungstyp	funktional
Name	Navigationsvorschläge annehmen
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, Navigationsvorschläge annehmen zu können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	sys-fkt-vorseins
Ursprung	
Referenzen	

Liste der nicht-funktionalen Anforderungen

Zuletzt geändert von	Daniel, 10.07.19
Status	offen
Begründung	
Wird umgesetzt	

ID	nav-fkt-vorsannehmsprache
Anforderungstyp	funktional
Name	Navigationsvorschläge annehmen Sprache
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, Navigationsvorschläge über einen Sprachbefehl annehmen zu können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	nav-fkt-vorsannehm
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	
Wird umgesetzt	

ID	nav-fkt-vorsannehmgeste
Anforderungstyp	funktional
Name	Navigationsvorschläge annehmen Geste
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, Navigationsvorschläge über eine Geste annehmen zu können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	nav-fkt-vorsannehm
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	
Wird umgesetzt	

ID	nav-fkt-vorsablehn
Anforderungstyp	funktional
Name	Navigationsvorschläge ablehnen

Liste der nicht-funktionalen Anforderungen

Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, Navigationsvorschläge ablehnen zu können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	sys-fkt-vorseins
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel, 10.07.19
Status	offen
Begründung	
Wird umgesetzt	

ID	nav-fkt-vorsablehnspalte
Anforderungstyp	funktional
Name	Navigationsvorschläge ablehnen Sprache
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, Navigationsvorschläge über einen Sprachbefehl ablehnen zu können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	nav-fkt-vorsablehn
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel, 10.07.19
Status	offen
Begründung	
Wird umgesetzt	

ID	nav-fkt-vorsablehnngeste
Anforderungstyp	funktional
Name	Navigationsvorschläge ablehnen Geste
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, Navigationsvorschläge über eine Geste ablehnen zu können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	nav-fkt-vorsablehn
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel, 02.07.19

Liste der nicht-funktionalen Anforderungen

Status	offen
Begründung	D
Wird umgesetzt	

ID	nav-fkt-tpeingabe
Anforderungstyp	funktional
Name	Eingabe Tagesprogramms
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, ein Tagesprogramm eingeben zu können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	sys-fkt-navfunk
Ursprung	User Story Navigation 5
Referenzen	Tagesprogramm im Glossar
Zuletzt geändert von	Daniel, 04.07.19
Status	offen
Begründung	Durch die Angabe eines Tagesprogramms wird dem System eine Menge von Zielen mitgeteilt, sodass die geplante Route im Voraus optimiert werden kann.
Wird umgesetzt	

ID	nav-fkt-tpeingabesprache
Anforderungstyp	funktional
Name	Eingabe Tagesprogramm Sprache
Anforderungssatz	Das <i>Ophelia</i> -Modul muss dem Nutzer die Möglichkeit bieten, ein Tagesprogramm über einen Sprachbefehl eingeben zu können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	nav-fkt-tpeingabe
Ursprung	User Story Navigation 5
Referenzen	Tagesprogramm im Glossar
Zuletzt geändert von	Daniel, 04.07.19
Status	offen
Begründung	
Wird umgesetzt	

ID	nav-fkt-tpeingabegeste
Anforderungstyp	funktional
Name	Eingabe Tagesprogramm Geste

Liste der nicht-funktionalen Anforderungen

Anforderungssatz	<i>Ophelia</i> muss dem Nutzer die Möglichkeit bieten, ein Tagesprogramm über einen Gesten-Befehl eingeben zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die <i>Ophelia-App</i> geöffnet hat (Vorbedingung) und in das Navigationsmenü wechselt (Aktion), stellt die <i>Ophelia-App</i> ein UI-Element bereit, um ein Tagesprogramm eingeben zu können.
Abgeleitet von	nav-fkt-tpeingabe
Ursprung	User Story Navigation 5
Referenzen	Tagesprogramm im Glossar
Zuletzt geändert von	Daniel, 04.07.19
Status	offen
Begründung	
Wird umgesetzt	

Statische Einstellungen

Benutzereingaben

ID	ben-fkt-touch
Anforderungstyp	funktional
Name	Befehle ohne Touch-Eingaben
Anforderungssatz	<i>Ophelia</i> muss dem Nutzer die Möglichkeit bieten dem Fahrzeug ohne Touch-Eingaben Befehle zu erteilen.
Nachgewiesen durch	
Akzeptanzkriterium	Vorbedingungen, Schritte (Was der Nutzer macht), Nachbedingungen
Abgeleitet von	sys-fkt-touchmin
Ursprung	User Story Benutzereingaben 1
Referenzen	
Zuletzt geändert von	Daniel, 14.06.19
Status	offen
Begründung	Systemeingaben über Berührungen lenken den Nutzer während der Fahrt ab. Eine Eingabemöglichkeit ohne Berührungen soll dieses Problem lösen.
Wird umgesetzt	

Rollos

Liste der nicht-funktionalen Anforderungen

ID	rol-fkt-ein
Anforderungstyp	funktional
Name	Standardposition der Rollos
Anforderungssatz	Das <i>Ophelia</i> -Modul muss die Standardposition.
Nachgewiesen durch	
Akzeptanzkriterium	Vorbedingungen, Schritte (Was der Nutzer macht), Nachbedingungen
Abgeleitet von	sys-fkt-touchmin
Ursprung	User Story Benutzereingaben 1
Referenzen	
Zuletzt geändert von	Daniel, 14.06.19
Status	offen
Begründung	Systemeingaben über Berührungen lenken den Nutzer während der Fahrt ab. Eine Eingabemöglichkeit ohne Berührungen soll dieses Problem lösen.
Wird umgesetzt	

Klimatisierung

ID	klima-fkt-autoeins
Anforderungstyp	funktional
Name	Automatische Voreinstellungen
Anforderungssatz	Das <i>Ophelia</i> -Modul muss Voreinstellungen aus den Präferenzen des Nutzers lernen können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	-
Ursprung	Vision
Referenzen	Visionspräsentation
Zuletzt geändert von	02.07.19
Status	offen
Begründung	<i>Ophelia</i> soll das Komfortgefühl der Nutzer erhöhen, indem automatisch Voreinstellungen vorgenommen werden.
Wird umgesetzt	

KI

ID	nav-fkt-abfahrzeitdaten
Anforderungstyp	funktional
Name	Abfahrtzeiten Daten beziehen

Liste der nicht-funktionalen Anforderungen

Anforderungssatz	Das <i>Ophelia</i> -Modul muss Testdaten für Abfahrtszeiten aus den angegebenen Abfahrtszeiten in den Tagesprogrammen eines Nutzers entnehmen.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	User Story Navigation 9
Referenzen	
Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	Die zum Lernen benötigten Testdaten werden aus den Angaben der Nutzer im Nutzerprofil entnommen. Dort werden Abfahrtszeiten zu Tagesprogrammen angegeben.
Wird umgesetzt	

ID	nav-fkt-abfahrzeitdaten
Anforderungstyp	funktional
Name	Abfahrtzeiten an Server senden
Anforderungssatz	Das <i>Ophelia</i> -Modul muss Abfahrtszeiten eines Nutzers an den Server senden können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	User Story Navigation 9
Referenzen	
Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	Da die Testdaten auf dem Server benötigt werden müssen diese an den Server gesendet werden. Außerdem sind Abfahrtszeiten nutzerbezogene Daten und werden daher auf dem Server im Nutzerprofil gespeichert.
Wird umgesetzt	

6.4.3. App

Musik

Navigation

ID	nav-fkt-app
Anforderungstyp	funktional

Liste der nicht-funktionalen Anforderungen

Name	Navigationsfunktionen per <i>Ophelia-App</i>
Anforderungssatz	Die <i>Ophelia-App</i> muss dem Nutzer die Möglichkeit bieten, Navigationsfunktionen verwenden zu können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel, 04.07.19
Status	offen
Begründung	
Wird umgesetzt	

ID	nav-fkt-tagesprogrammeingabe
Anforderungstyp	funktional
Name	Angabe eines Tagesprogramms
Anforderungssatz	Die <i>Ophelia-App</i> muss dem Nutzer die Möglichkeit bieten, einen Namen für ein Tagesprogramm eingeben zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die <i>Ophelia-App</i> geöffnet hat (Vorbedingung) und in das Navigationsmenü wechselt (Aktion), stellt die <i>Ophelia-App</i> ein UI-Element bereit, um einen Namen für ein Tagesprogramm eingeben zu können.
Abgeleitet von	nav-fkt-app
Ursprung	User Story Navigation 5
Referenzen	Tagesprogramm im Glossar
Zuletzt geändert von	Daniel, 04.07.19
Status	offen
Begründung	Durch die Angabe eines Tagesprogramms wird <i>Ophelia</i> eine Menge von Navigationszielen mitgeteilt, sodass die geplante Route im Voraus optimiert werden kann.
Wird umgesetzt	

ID	nav-fkt-tagesprogrammziele
Anforderungstyp	funktional
Name	Tagesprogrammziel
Anforderungssatz	Die <i>Ophelia-App</i> muss dem Nutzer die Möglichkeit bieten, Navigationsziele zu einem Tagesprogramm angeben zu können.
Nachgewiesen durch	

Liste der nicht-funktionalen Anforderungen

Akzeptanzkriterium	Wenn der Nutzer die <i>Ophelia-App</i> geöffnet hat (Vorbedingung) und in das Tagesprogramm Menü wechselt (Aktion), stellt die App ein UI-Element zur Eingabe von Navigationszielen bereit (Nachbedingung).
Abgeleitet von	nav-fkt-tagesprogrammeingabe
Ursprung	User Story Navigation 5
Referenzen	
Zuletzt geändert von	Daniel, 26.06.19
Status	offen
Begründung	Ein Tagesprogramm besteht aus einer Menge von Navigationszielen die der Nutzer in <i>Ophelia</i> eingeben können muss.
Wird umgesetzt	

ID	nav-fkt-tagesprogrammspeichern
Anforderungstyp	funktional
Name	Tagesprogramm speichern
Anforderungssatz	Die <i>Ophelia-App</i> muss ein Tagesprogramm eines Nutzers im Nutzerprofil speichern können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer ein Tagesprogramm angibt (Vorbedingung) und dieses in der <i>Ophelia-App</i> speichert (Aktion), muss die <i>Ophelia-App</i> das Tagesprogramm im Nutzerprofil speichern können (Nachbedingung).
Abgeleitet von	nav-fkt-tagesprogrammeingabe
Ursprung	User Story Navigation 5
Referenzen	
Zuletzt geändert von	Daniel, 25.06.19
Status	offen
Begründung	Damit <i>Ophelia</i> auf das Tagesprogramm zugreifen kann, muss es gespeichert werden.
Wird umgesetzt	

ID	nav-fkt-tagesprogrammauswahl
Anforderungstyp	funktional
Name	Tagesprogramm auswählen
Anforderungssatz	Die Ophelia-App muss dem Nutzer die Möglichkeit bieten, ein Tagesprogramm aus der Menge der gespeicherten Tagesprogramme auswählen zu können.
Nachgewiesen durch	

Akzeptanzkriterium	Wenn der Nutzer die <i>Ophelia-App</i> geöffnet hat (Vorbedingung), das Menü für Tagesprogramme öffnet, stellt die <i>Ophelia-App</i> ein UI-Element bereit, um ein Tagesprogramm auswählen zu können (Nachbedingung).
Abgeleitet von	nav-fkt-tagesprogrammeingabe
Ursprung	User Story Navigation 5
Referenzen	
Zuletzt geändert von	Daniel, 30.06.19
Status	offen
Begründung	Der Nutzer soll ein Tagesprogramm auswählen können, das im Fahrzeug umgesetzt werden soll.
Wird umgesetzt	

ID	nav-fkt-tagesprogrammaendern
Anforderungstyp	funktional
Name	Tagesprogramm ändern
Anforderungssatz	Die <i>Ophelia-App</i> muss dem Nutzer die Möglichkeit bieten, ein Tagesprogramm ändern zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die <i>Ophelia-App</i> geöffnet hat (Vorbedingung) und das Menü für Tagesprogramme öffnet (Aktion), stellt die <i>Ophelia-App</i> ein UI-Element bereit, um ein Tagesprogramm ändern zu können (Nachbedingung).
Abgeleitet von	nav-fkt-tagesprogrammeingabe
Ursprung	User Story Navigation 5
Referenzen	
Zuletzt geändert von	Daniel, 26.06.19
Status	offen
Begründung	Damit ein Nutzer Änderungen an einem Tagesprogramm vornehmen kann.
Wird umgesetzt	

ID	nav-fkt-tagesprogrammzielhinzufuegen
Anforderungstyp	funktional
Name	Tagesprogramm Ziel hinzufügen
Anforderungssatz	Die <i>Ophelia-App</i> muss dem Nutzer die Möglichkeit bieten, in einem bestehenden Tagesprogramm neue Ziele hinzufügen zu können.
Nachgewiesen durch	

Liste der nicht-funktionalen Anforderungen

Akzeptanzkriterium	Wenn der Nutzer die <i>Ophelia-App</i> geöffnet hat (Vorbedingung) und das Menü für Tagesprogramme öffnet (Aktion), stellt die <i>Ophelia-App</i> ein UI-Element bereit, um ein neues Ziel zu einem bestehenden Tagesprogramm hinzuzufügen (Nachbedingung).
Abgeleitet von	nav-fkt-tagesprogrammaendern
Ursprung	User Story Navigation 5
Referenzen	
Zuletzt geändert von	Daniel, 26.06.19
Status	offen
Begründung	Damit ein Nutzer Änderungen an einem Tagesprogramm vornehmen kann.
Wird umgesetzt	

ID	nav-fkt-tagesprogrammzielentfernen
Anforderungstyp	funktional
Name	Tagesprogramm Ziel entfernen
Anforderungssatz	Die <i>Ophelia-App</i> muss dem Nutzer die Möglichkeit bieten, in einem bestehenden Tagesprogramm Ziele entfernen zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die <i>Ophelia-App</i> geöffnet hat (Vorbedingung) und das Menü für Tagesprogramme öffnet (Aktion), stellt die <i>Ophelia-App</i> ein UI-Element bereit, um ein Ziel eines bestehenden Tagesprogramms zu entfernen (Nachbedingung).
Abgeleitet von	nav-fkt-tagesprogrammaendern
Ursprung	User Story Navigation 5
Referenzen	
Zuletzt geändert von	Daniel, 26.06.19
Status	offen
Begründung	Damit ein Nutzer Änderungen an einem Tagesprogramm vornehmen kann.
Wird umgesetzt	

ID	nav-fkt-tagesprogrammzielereihenfolge
Anforderungstyp	funktional
Name	Tagesprogramm Ziele Reihenfolge
Anforderungssatz	Die <i>Ophelia-App</i> muss dem Nutzer die Möglichkeit bieten, die Reihenfolge der Ziele in einem Tagesprogramm ändern zu können.

Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die <i>Ophelia-App</i> geöffnet hat (Vorbedingung) und das Menü für Tagesprogramme öffnet (Aktion), stellt die <i>Ophelia-App</i> ein UI-Element bereit, um die Reihenfolge der Ziele zu ändern (Nachbedingung).
Abgeleitet von	nav-fkt-tagesprogrammaendern
Ursprung	User Story Navigation 5
Referenzen	
Zuletzt geändert von	Daniel, 26.06.19
Status	offen
Begründung	Damit ein Nutzer Änderungen an einem Tagesprogramm vornehmen kann.
Wird umgesetzt	

ID	nav-fkt-tagesprogrammaenderungenspeichern
Anforderungstyp	funktional
Name	Tagesprogramm Änderungen speichern
Anforderungssatz	Die <i>Ophelia-App</i> muss Änderungen eines Tagesprogramms speichern können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die <i>Ophelia-App</i> geöffnet hat (Vorbedingung) und das Menü für Tagesprogramme öffnet (Aktion), stellt die <i>Ophelia-App</i> ein UI-Element bereit, Änderungen eines Tagesprogramms zu speichern (Nachbedingung).
Abgeleitet von	nav-fkt-tagesprogrammaendern
Ursprung	User Story Navigation 5
Referenzen	
Zuletzt geändert von	Daniel, 26.06.19
Status	offen
Begründung	Damit die Änderungen eines Nutzers an einem Tagesprogramm gespeichert werden können.
Wird umgesetzt	

ID	nav-fkt-tagesprogrammloeschen
Anforderungstyp	funktional
Name	Tagesprogramm löschen
Anforderungssatz	Die <i>Ophelia-App</i> muss dem Nutzer die Möglichkeit bieten, ein Tagesprogramm im Nutzerprofil löschen zu können.
Nachgewiesen durch	

Liste der nicht-funktionalen Anforderungen

Akzeptanzkriterium	Wenn der Nutzer die <i>Ophelia-App</i> geöffnet hat (Vorbedingung), das Menü für Tagesprogramme öffnet und ein Tagesprogramm löscht (Aktion), muss das Tagesprogramm im Nutzerprofil gelöscht sein (Nachbedingung).
Abgeleitet von	nav-fkt-tagesprogrammeingabe
Ursprung	User Story Navigation 5
Referenzen	
Zuletzt geändert von	Daniel, 25.06.19
Status	offen
Begründung	Damit Datenschutzrichtlinien eingehalten werden, muss ein Nutzer die Möglichkeit haben seine Daten in <i>Ophelia</i> zu löschen.
Wird umgesetzt	

ID	nav-fkt-tagesprogrammsenden
Anforderungstyp	funktional
Name	Tagesprogramm senden
Anforderungssatz	Die <i>Ophelia-App</i> muss die Daten eines Tagesprogramms an das Fahrzeug senden können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn eine Verbindungsmöglichkeit zwischen der <i>Ophelia-App</i> und dem Fahrzeug besteht und Tagesprogramme in der <i>Ophelia-App</i> gespeichert sind, die noch nicht auf dem Fahrzeug vorhanden sind (Vorbedingung), und die <i>Ophelia-App</i> die Tagesprogramme an das Fahrzeug sendet (Aktion), müssen die Tagesprogramme im Fahrzeug vorhanden sein.
Abgeleitet von	nav-fkt-tagesprogrammspeichern
Ursprung	User Story Navigation 5
Referenzen	
Zuletzt geändert von	Daniel, 25.06.19
Status	offen
Begründung	Damit die Daten eines Tagesprogramms im Fahrzeug umgesetzt werden können, müssen diese an das Fahrzeug gesendet werden.
Wird umgesetzt	

ID	nav-fkt-tagesprogrammabfahrtszeit
Anforderungstyp	funktional
Name	Tagesprogramm Abfahrtszeit Eingabe

Liste der nicht-funktionalen Anforderungen

Anforderungssatz	Die <i>Ophelia-App</i> muss dem Nutzer die Möglichkeit bieten, eine Abfahrtszeit bezüglich eines Tagesprogramms eingeben zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die <i>Ophelia-App</i> geöffnet hat (Vorbedingung), in das Tagesprogramm Menü wechselt und eine Abfahrtszeit eingibt (Aktion), erwartet der Nutzer, dass die Eingabe angenommen wird.
Abgeleitet von	nav-fkt-tagesprogrammeingabe
Ursprung	User Story Navigation 5
Referenzen	
Zuletzt geändert von	Daniel, 25.06.19
Status	offen
Begründung	Durch die Angabe einer Abfahrtszeit bezüglich eines Tagesprogramms soll <i>Ophelia</i> die Möglichkeit haben, Einstellungen im Fahrzeug zum passenden Zeitpunkt vorzunehmen.
Wird umgesetzt	

ID	nav-fkt-tagesprogrammabfahrtszeitspeichern
Anforderungstyp	funktional
Name	Tagesprogramm Abfahrtszeit speichern
Anforderungssatz	Die <i>Ophelia-App</i> muss eine Abfahrtszeit bezüglich eines Tagesprogramms eines Nutzers speichern können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die <i>Ophelia-App</i> geöffnet hat (Vorbedingung), in das Tagesprogramm Menü wechselt und eine Abfahrtszeit ein gibt (Aktion), speichert die <i>Ophelia-App</i> die Abfahrtszeit (Nachbedingung).
Abgeleitet von	nav-fkt-tagesprogrammabfahrtszeit
Ursprung	User Story Navigation 5
Referenzen	
Zuletzt geändert von	Daniel, 30.06.19
Status	offen
Begründung	Damit <i>Ophelia</i> auf die Abfahrtszeit eines Tagesprogramms zugreifen kann, muss sie gespeichert werden.
Wird umgesetzt	

ID	nav-fkt-tagesprogrammabfahrtszeitaendern
Anforderungstyp	funktional
Name	Tagesprogramm Abfahrtszeit ändern

Liste der nicht-funktionalen Anforderungen

Anforderungssatz	Die <i>Ophelia-App</i> muss dem Nutzer die Möglichkeit bieten, eine Abfahrtszeit bezüglich eines Tagesprogramms ändern können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die <i>Ophelia-App</i> geöffnet hat (Vorbedingung), in das Tagesprogramm Menü wechselt und eine Abfahrtszeit eingibt (Aktion), stellt die <i>Ophelia-App</i> ein UI-Element bereit, welches Abfahrtszeiten ändert.
Abgeleitet von	nav-fkt-tagesprogrammabfahrtszeit
Ursprung	User Story Navigation 5
Referenzen	
Zuletzt geändert von	Daniel, 25.06.19
Status	offen
Begründung	Damit <i>Ophelia</i> auf die Abfahrtszeit eines Tagesprogramms zugreifen kann, muss sie gespeichert werden.
Wird umgesetzt	

ID	nav-fkt-tagesprogrammabfahrtszeitaendernspeichern
Anforderungstyp	funktional
Name	Tagesprogramm Abfahrtszeit Änderung speichern
Anforderungssatz	Die <i>Ophelia-App</i> muss Änderungen der Abfahrtszeit eines Tagesprogramms speichern können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die <i>Ophelia-App</i> geöffnet hat (Vorbedingung), in das Tagesprogramm Menü wechselt und eine Abfahrtszeit ändert (Aktion), stellt die <i>Ophelia-App</i> ein UI-Element bereit, welches Änderungen der Abfahrtszeiten speichert.
Abgeleitet von	nav-fkt-tagesprogrammabfahrtszeit
Ursprung	User Story Navigation 5
Referenzen	
Zuletzt geändert von	Daniel, 30.06.19
Status	offen
Begründung	Damit ein Nutzer Änderungen an der Abfahrtszeit speichern kann, falls sich diese verändern sollte.
Wird umgesetzt	

ID	nav-fkt-tagesprogrammabfahrtszeitloeschen
Anforderungstyp	funktional
Name	Tagesprogramm Abfahrtszeit löschen

Liste der nicht-funktionalen Anforderungen

Anforderungssatz	Die <i>Ophelia-App</i> muss eine Abfahrtszeit bezüglich eines Tagesprogramms eines Nutzers löschen können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die <i>Ophelia-App</i> geöffnet hat (Vorbedingung), das Menü für Tagesprogramme öffnet und eine Abfahrtszeit eines Tagesprogramms löscht (Aktion), muss die Abfahrtszeit des gewählten Tagesprogramms im Nutzerprofil gelöscht sein (Nachbedingung).
Abgeleitet von	nav-fkt-tagesprogrammabfahrtszeit
Ursprung	User Story Navigation 5
Referenzen	
Zuletzt geändert von	Daniel, 25.06.19
Status	offen
Begründung	Damit Datenschutzrichtlinien eingehalten werden, muss ein Nutzer die Möglichkeit haben seine Daten in <i>Ophelia</i> zu löschen.
Wird umgesetzt	

ID	nav-fkt-tagesprogrammabfahrtszeitsendenfahrzeug
Anforderungstyp	funktional
Name	Tagesprogramm Abfahrtszeit ans Fahrzeug senden
Anforderungssatz	Die <i>Ophelia-App</i> muss eine Abfahrtszeit bezüglich eines Tagesprogramms eines Nutzers an das <i>Ophelia-Modul</i> senden können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn eine Verbindungsmöglichkeit zwischen der <i>Ophelia-App</i> und dem Fahrzeug besteht und Abfahrtszeiten zu Tagesprogrammen in der <i>Ophelia-App</i> gespeichert sind, die noch nicht auf dem Fahrzeug vorhanden sind (Vorbedingung), und die <i>Ophelia-App</i> die Abfahrtszeiten zu Tagesprogrammen an das Fahrzeug sendet (Aktion), müssen die Abfahrtszeiten zu Tagesprogrammen im Fahrzeug vorhanden sein.
Abgeleitet von	nav-fkt-tagesprogrammabfahrtszeit
Ursprung	User Story Navigation 6
Referenzen	
Zuletzt geändert von	Daniel, 25.06.19
Status	offen
Begründung	Damit die Einstellungen bezüglich der Abfahrtszeiten zu Tagesprogrammen im Fahrzeug umgesetzt werden können, müssen diese an das Fahrzeug gesendet werden.
Wird umgesetzt	

Liste der nicht-funktionalen Anforderungen

ID	nav-fkt-tagesprogrammabfahrtszeitsendenserver
Anforderungstyp	funktional
Name	Tagesprogramm Abfahrtszeit an Server senden
Anforderungssatz	Die <i>Ophelia-App</i> muss eine Abfahrtszeit bezüglich eines Tagesprogramms eines Nutzers an den Server senden können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn eine Verbindungsmöglichkeit zwischen der <i>Ophelia-App</i> und dem Server besteht und Abfahrtszeiten zu Tagesprogrammen in der <i>Ophelia-App</i> gespeichert sind, die noch nicht auf dem Server vorhanden sind (Vorbedingung), und die <i>Ophelia-App</i> die Abfahrtszeiten zu Tagesprogrammen an den Server sendet (Aktion), müssen die Abfahrtszeiten zu Tagesprogrammen im Server vorhanden sein.
Abgeleitet von	nav-fkt-tagesprogrammabfahrtszeit
Ursprung	User Story Navigation 6
Referenzen	
Zuletzt geändert von	Daniel, 25.06.19
Status	offen
Begründung	Falls die Daten nicht an das Fahrzeug gesendet werden können, sollen diese auf dem Server gespeichert werden.
Wird umgesetzt	

ID	nav-fkt-lauteingabeapp
Anforderungstyp	funktional
Name	Standardnavigationslautstärke Eingabe App
Anforderungssatz	Die <i>Ophelia-App</i> muss dem Nutzer die Möglichkeit bieten, die Standardnavigationslautstärke von <i>Ophelia</i> eingeben zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer seine persönliche Standardnavigationslautstärke über die <i>Ophelia-App</i> eingeben möchte (Vorbedingung), stellt die <i>Ophelia-App</i> ein UI-Element bereit, über das die Standardnavigationslautstärke eingegeben werden kann (Nachbedingung).
Abgeleitet von	
Ursprung	User Story Navigation 1
Referenzen	Standardlautstärke im Glossar
Zuletzt geändert von	Daniel, 30.06.19
Status	offen

Liste der nicht-funktionalen Anforderungen

Begründung	Damit die gewünschte Standardnavigationslautstärke des Nutzers erfasst werden kann, soll er die Möglichkeit haben diese über die <i>Ophelia-App</i> eingeben zu können.
Wird umgesetzt	

ID	nav-fkt-lauterhoehenapp
Anforderungstyp	funktional
Name	Standardnavigationslautstärke Erhöhung App
Anforderungssatz	Die <i>Ophelia-App</i> muss dem Nutzer die Möglichkeit bieten, die Standardnavigationslautstärke von <i>Ophelia</i> erhöhen zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn ein Wert für die Standardnavigationslautstärke in <i>Ophelia</i> gespeichert ist (Vorbedingung) und der Nutzer die Lautstärke über die <i>Ophelia-App</i> erhöht (Aktion), wird die Standardnavigationslautstärke im Fahrzeug erhöht.
Akzeptanzkriterium 2	Wenn ein Wert für die Standardnavigationslautstärke in <i>Ophelia</i> gespeichert ist (Vorbedingung) und der Nutzer die Lautstärke über die <i>Ophelia-App</i> erhöhen will (Aktion), stellt die <i>Ophelia-App</i> ein UI-Element für die Erhöhung der Standardnavigationslautstärke bereit.
Abgeleitet von	nav-fkt-lauteingabeapp
Ursprung	User Story Navigation 1
Referenzen	Standardlautstärke im Glossar
Zuletzt geändert von	Daniel, 30.06.19
Status	offen
Begründung	Damit die gewünschte Standardnavigationslautstärke des Nutzers erhöht werden kann, soll er die Möglichkeit haben diese über die <i>Ophelia-App</i> erhöhen zu können.
Wird umgesetzt	

ID	nav-fkt-lautverringernapp
Anforderungstyp	funktional
Name	Standardnavigationslautstärke Verringern App
Anforderungssatz	Die <i>Ophelia-App</i> muss dem Nutzer die Möglichkeit bieten, die Standardnavigationslautstärke von <i>Ophelia</i> verringern zu können.
Nachgewiesen durch	

Liste der nicht-funktionalen Anforderungen

Akzeptanzkriterium	Wenn ein Wert für die Standardnavigationslautstärke in <i>Ophelia</i> gespeichert ist (Vorbedingung) und der Nutzer die Lautstärke über die <i>Ophelia-App</i> verringert (Aktion), wird die Standardnavigationslautstärke im Fahrzeug verringert.
Akzeptanzkriterium 2	Wenn ein Wert für die Standardnavigationslautstärke in <i>Ophelia</i> gespeichert ist (Vorbedingung) und der Nutzer die Lautstärke über die <i>Ophelia-App</i> verringern will (Aktion), stellt die <i>Ophelia-App</i> ein UI-Element für die Verringerung der Standardnavigationslautstärke bereit.
Abgeleitet von	nav-fkt-lauteingabeapp
Ursprung	User Story Navigation 1
Referenzen	Standardlautstärke im Glossar
Zuletzt geändert von	Daniel, 30.06.19
Status	offen
Begründung	Damit die gewünschte Standardnavigationslautstärke des Nutzers verringert werden kann, soll er die Möglichkeit haben diese über die <i>Ophelia-App</i> verringern zu können.
Wird umgesetzt	

ID	nav-fkt-gegenstaendeangabe
Anforderungstyp	funktional
Name	Angabe von Gegenständen
Anforderungssatz	Die <i>Ophelia-App</i> muss dem Nutzer die Möglichkeit bieten, optional Gegenstände bezüglich eines Ziels eingeben zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die <i>Ophelia-App</i> geöffnet hat (Vorbedingung) und in das Routenmenü wechselt (Aktion), besteht die Möglichkeit, Gegenstände bezüglich eines Ziels eingeben zu können (Nachbedingung).
Abgeleitet von	nav-fkt-gegenstaendeerinnern
Ursprung	User Story Navigation 8
Referenzen	
Zuletzt geändert von	Daniel, 30.06.19
Status	offen
Begründung	Im Rahmen des Kontexts „Helping“ soll durch <i>Ophelia</i> eine Erinnerungsfunktion an Gegenstände angeboten werden.
Wird umgesetzt	

Liste der nicht-funktionalen Anforderungen

ID	nav-fkt-gegenstaendeaendern
Anforderungstyp	funktional
Name	Gegenstände ändern
Anforderungssatz	Die <i>Ophelia-App</i> muss dem Nutzer die Möglichkeit bieten, optionale Gegenstände bezüglich eines Ziels ändern zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die <i>Ophelia-App</i> geöffnet hat (Vorbedingung) und in das Routenmenü wechselt (Aktion), besteht die Möglichkeit, Gegenstände bezüglich eines Ziels eingeben zu können (Nachbedingung).
Abgeleitet von	nav-fkt-gegenstaendeerinnern
Ursprung	User Story Navigation 8
Referenzen	
Zuletzt geändert von	Daniel, 30.06.19
Status	offen
Begründung	Optionale Gegenstände sollen änderbar sein, sodass bei einer falschen Angabe nicht der Gegenstand gelöscht und neu erstellt wird.
Wird umgesetzt	

ID	nav-fkt-gegenstaendloeschen
Anforderungstyp	funktional
Name	Angabe von Gegenständen
Anforderungssatz	Die <i>Ophelia-App</i> muss dem Nutzer die Möglichkeit bieten, optionale Gegenstände bezüglich eines Ziels löschen zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die <i>Ophelia-App</i> geöffnet hat (Vorbedingung) und in das Routenmenü wechselt (Aktion), besteht die Möglichkeit, Gegenstände bezüglich eines Ziels eingeben zu können (Nachbedingung).
Abgeleitet von	nav-fkt-gegenstaendeerinnern
Ursprung	User Story Navigation 8
Referenzen	
Zuletzt geändert von	Daniel, 30.06.19
Status	offen
Begründung	Im Rahmen des Kontexts „Helping“ soll durch <i>Ophelia</i> eine Erinnerungsfunktion an Gegenstände angeboten werden.
Wird umgesetzt	

Liste der nicht-funktionalen Anforderungen

ID	nav-fkt-gegenstaendepushapp
Anforderungstyp	funktional
Name	Gegenstände Push-Benachrichtigung
Anforderungssatz	Die <i>Ophelia-App</i> muss den Nutzer eine Stunde vor Abfahrtszeit über eine Push-Benachrichtigung an Gegenstände bezüglich eines Ziels erinnern können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer Gegenstände bezüglich eines Ziels angegeben hat (Vorbedingung), erfolgt eine Push-Benachrichtigung über die <i>Ophelia-App</i> (Nachbedingung).
Abgeleitet von	nav-fkt-gegenstaendeerinnern
Ursprung	User Story Navigation 8
Referenzen	
Zuletzt geändert von	Daniel, 30.06.19
Status	offen
Begründung	Im Rahmen des Kontexts „Helping“ soll durch <i>Ophelia</i> eine Erinnerungsfunktion an Gegenstände angeboten werden.
Wird umgesetzt	

ID	nav-fkt-gegenstaendedepushappdeak
Anforderungstyp	funktional
Name	Deaktivierung Push-Benachrichtigung zur Erinnerung
Anforderungssatz	Die <i>Ophelia-App</i> muss dem Nutzer die Möglichkeit bieten, Push-Benachrichtigungen bezüglich der Erinnerung an Gegenstände deaktivieren zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die <i>Ophelia-App</i> geöffnet hat (Vorbedingung) und in das Menü zur Erinnerung wechselt (Aktion), stellt die <i>Ophelia-App</i> ein UI-Element bereit, um die Erinnerung zu deaktivieren.
Akzeptanzkriterium 2	Wenn der Nutzer Gegenstände bezüglich eines Ziels angegeben hat und Push-Benachrichtigungen bezüglich der Erinnerung an Gegenstände deaktiviert hat (Vorbedingung), sendet die <i>Ophelia-App</i> keine Push-Benachrichtigung bezüglich der Erinnerung an Gegenstände an den Nutzer (Nachbedingung).
Abgeleitet von	nav-fkt-gegenstaendepushapp
Ursprung	User Story Navigation 8
Referenzen	
Zuletzt geändert von	Daniel, 30.06.19
Status	offen

Begründung	Falls ein Nutzer die Funktion der Push-Benachrichtigung nicht verwenden möchte, müssen Push-Benachrichtigungen deaktiviert werden können.
Wird umgesetzt	

ID	nav-fkt-parkplatzortabrufen
Anforderungstyp	funktional
Name	Parkplatzort abrufen
Anforderungssatz	Die <i>Ophelia</i> -App soll dem Nutzer die Möglichkeit bieten, den Standort des letzten Parkplatzes des Fahrzeugs abrufen zu können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn das Fahrzeug vom Fahrer an einem Parkplatz abgestellt und verriegelt wurde und der Nutzer den Parkplatzort abrufen möchte (Vorbedingung), stellt die <i>Ophelia</i> -App ein UI-Element bereit, worüber der Parkplatzort abgerufen werden kann (Nachbedingung).
Abgeleitet von	
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel 01.07.19
Status	offen
Begründung	Um das geparkte Fahrzeug leichter zu finden soll der Nutzer den Standort des Fahrzeugs über <i>Ophelia</i> abrufen können.

6.4.4. Server

Navigation

ID	nav-fkt-lautspeicherver
Anforderungstyp	funktional
Name	Standardnavigationslautstärke Speicherort
Anforderungssatz	Der <i>Ophelia</i> -Server muss die Standardnavigationslautstärke von <i>Ophelia</i> im Nutzerprofil speichern können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn die Standardnavigationslautstärke vom Fahrzeug oder der <i>Ophelia</i> -App an den Server gesendet wird (Vorbedingung), soll der Server die Standardnavigationslautstärke im Profil des Nutzers speichern (Nachbedingung).

Liste der nicht-funktionalen Anforderungen

Abgeleitet von	
Ursprung	User Story Navigation 1
Referenzen	Standardlautstärke im Glossar
Zuletzt geändert von	Daniel, 14.06.19
Status	offen
Begründung	Da die Standardnavigationslautstärke ein nutzerbezogenes Datum ist, wird es im Nutzerprofil gespeichert.
Wird umgesetzt	

ID	nav-fkt-lautloeschserver
Anforderungstyp	funktional
Name	Standardnavigationslautstärke löschen
Anforderungssatz	Der Ophelia-Server muss die Standardnavigationslautstärke von <i>Ophelia</i> im Nutzerprofil löschen können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die Standardnavigationslautstärke im Fahrzeug oder in der App löscht (Vorbedingung), ist die Standardnavigationslautstärke im Nutzerprofil gelöscht (Nachbedingung).
Abgeleitet von	
Ursprung	User Story Navigation 1
Referenzen	Standardlautstärke im Glossar
Zuletzt geändert von	Daniel, 25.06.19
Status	offen
Begründung	Damit Datenschutzrichtlinien eingehalten werden, muss ein Nutzer die Möglichkeit haben seine Daten in <i>Ophelia</i> zu löschen.
Wird umgesetzt	

ID	nav-fkt-routenspeichserver
Anforderungstyp	funktional
Name	Abfahrtszeiten speichern
Anforderungssatz	Der Ophelia-Server muss Routen des Nutzers im Nutzerprofil speichern können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn die Abfahrtszeiten vom Fahrzeug oder der <i>Ophelia-App</i> an den Server gesendet werden (Vorbedingung), soll der Server die Abfahrtszeiten im Profil des Nutzers speichern (Nachbedingung).
Abgeleitet von	
Ursprung	User Story Navigation 9

Liste der nicht-funktionalen Anforderungen

Referenzen	
Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	Da die Abfahrtszeiten nutzerbezogene Daten sind, werden sie im Nutzerprofil gespeichert.
Wird umgesetzt	

ID	nav-fkt-routenloeschserver
Anforderungstyp	funktional
Name	Abfahrtszeiten Speicherort
Anforderungssatz	Der Ophelia-Server muss Routen des Nutzers im Nutzerprofil löschen können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn die Abfahrtszeiten vom Fahrzeug oder der <i>Ophelia-App</i> an den Server gesendet werden (Vorbedingung), soll der Server die Abfahrtszeiten im Profil des Nutzers speichern (Nachbedingung).
Abgeleitet von	
Ursprung	User Story Navigation 9
Referenzen	
Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	Da die Abfahrtszeiten nutzerbezogene Daten sind, werden sie im Nutzerprofil gespeichert.
Wird umgesetzt	

ID	nav-fkt-zielespeichserver
Anforderungstyp	funktional
Name	Ziele speichern
Anforderungssatz	Der Ophelia-Server muss Ziele des Nutzers im Nutzerprofil speichern können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn die Abfahrtszeiten vom Fahrzeug oder der <i>Ophelia-App</i> an den Server gesendet werden (Vorbedingung), soll der Server die Abfahrtszeiten im Profil des Nutzers speichern (Nachbedingung).
Abgeleitet von	
Ursprung	User Story Navigation 9
Referenzen	
Zuletzt geändert von	Daniel, 02.07.19
Status	offen

Liste der nicht-funktionalen Anforderungen

Begründung	Da die Abfahrtszeiten nutzerbezogene Daten sind, werden sie im Nutzerprofil gespeichert.
Wird umgesetzt	

ID	nav-fkt-zieleloeschserver
Anforderungstyp	funktional
Name	Routen speichern
Anforderungssatz	Der Ophelia-Server muss Routen des Nutzers im Nutzerprofil löschen können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn die Abfahrtszeiten vom Fahrzeug oder der <i>Ophelia-App</i> an den Server gesendet werden (Vorbedingung), soll der Server die Abfahrtszeiten im Profil des Nutzers speichern (Nachbedingung).
Abgeleitet von	
Ursprung	User Story Navigation 9
Referenzen	
Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	Da die Abfahrtszeiten nutzerbezogene Daten sind, werden sie im Nutzerprofil gespeichert.
Wird umgesetzt	

ID	nav-fkt-abfahrtszeitspeichserver
Anforderungstyp	funktional
Name	Abfahrtszeiten speichern
Anforderungssatz	Der Ophelia-Server muss Abfahrtszeiten des Nutzers im Nutzerprofil speichern können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn die Abfahrtszeiten vom Fahrzeug oder der <i>Ophelia-App</i> an den Server gesendet werden (Vorbedingung), soll der Server die Abfahrtszeiten im Profil des Nutzers speichern (Nachbedingung).
Abgeleitet von	
Ursprung	User Story Navigation 9
Referenzen	
Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	Da die Abfahrtszeiten nutzerbezogene Daten sind, werden sie im Nutzerprofil gespeichert.
Wird umgesetzt	

Liste der nicht-funktionalen Anforderungen

ID	nav-fkt-tpspeichserver
Anforderungstyp	funktional
Name	Tagesprogramme speichern
Anforderungssatz	Der Ophelia-Server muss Tagesprogramme des Nutzers im Nutzerprofil speichern können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	User Story Navigation 9
Referenzen	
Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	
Wird umgesetzt	

ID	nav-fkt-gegenstaendespeichserver
Anforderungstyp	funktional
Name	Gegenstände Speicherort speichern
Anforderungssatz	Der Ophelia-Server muss Gegenstände des Nutzers im Nutzerprofil speichern können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	
Ursprung	User Story Navigation 9
Referenzen	
Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	
Wird umgesetzt	

ID	nav-fkt-gegenstaendeloeschserver
Anforderungstyp	funktional
Name	Abfahrtszeiten löschen
Anforderungssatz	Der Ophelia-Server muss Gegenstände des Nutzers im Nutzerprofil löschen können.
Nachgewiesen durch	
Akzeptanzkriterium	Wenn der Nutzer die Abfahrtszeiten im Fahrzeug oder in der <i>Ophelia-App</i> löscht (Vorbedingung), sind die Abfahrtszeiten im Nutzerprofil gelöscht (Nachbedingung).
Abgeleitet von	
Ursprung	User Story Navigation 9

Liste der nicht-funktionalen Anforderungen

Referenzen	
Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	Damit Datenschutzrichtlinien eingehalten werden, muss ein Nutzer die Möglichkeit haben seine Daten in <i>Ophelia</i> zu löschen.
Wird umgesetzt	

KI

ID	ki-fkt-lernkomponente
Anforderungstyp	funktional
Name	Lernkomponente
Anforderungssatz	Der Ophelia-Server muss eine Lernkomponente enthalten.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	sys-fkt-lern
Ursprung	
Referenzen	Standardlautstärke im Glossar
Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	Damit Nutzerpräferenzen gelernt werden können muss der Server eine Lernkomponente enthalten, worin mit Nutzerdaten gelernt wird.
Wird umgesetzt	

ID	ki-fkt-abfahrzeitzielelern
Anforderungstyp	funktional
Name	Abfahrtzeiten und Ziele lernen
Anforderungssatz	Ophelia muss die Abfahrtszeiten der Nutzer in Zusammenhang mit Zielen lernen können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	sys-fkt-lern
Ursprung	User Story Navigation 9
Referenzen	
Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	.
Wird umgesetzt	

Liste der nicht-funktionalen Anforderungen

ID	ki-fkt-lautaenderlern
Anforderungstyp	funktional
Name	Navigationslautstärkenänderungen lernen
Anforderungssatz	Ophelia muss die Navigationslautstärkenänderungen der Nutzer lernen können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	sys-fkt-lern
Ursprung	User Story Navigation 9
Referenzen	
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	Das Lernen von Navigationslautstärkenänderungen dient dem Zweck, dass <i>Ophelia</i> diese selbstständig einstellt.
Wird umgesetzt	

ID	ki-fkt-lautlernausgeben
Anforderungstyp	funktional
Name	Navigationslautstärken ausgeben
Anforderungssatz	Die Lernkomponente des Ophelia-Servers muss Navigationslautstärken ausgeben können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	sys-fkt-lern
Ursprung	User Story Navigation 9
Referenzen	
Zuletzt geändert von	Daniel, 09.07.19
Status	offen
Begründung	<i>Ophelia</i> muss gelernte Navigationslautstärken ausgeben, damit diese an das <i>Ophelia</i> -Modul gesendet werden können.
Wird umgesetzt	

ID	ki-fkt-zielelernausgeben
Anforderungstyp	funktional
Name	Ziele ausgeben
Anforderungssatz	Die Lernkomponente des Ophelia-Servers muss Ziele ausgeben können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	sys-fkt-lern

Liste der nicht-funktionalen Anforderungen

Ursprung	User Story Navigation 9
Referenzen	
Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	Die gelernten Ziele sollen im Fahrzeug eingestellt werden oder dem Fahrer während der Fahrt vorgeschlagen werden.
Wird umgesetzt	

ID	ki-fkt-abfahrtzeitlernausgeben
Anforderungstyp	funktional
Name	Lernkomponente
Anforderungssatz	Die Lernkomponente des Ophelia-Servers muss Abfahrtszeiten ausgeben können.
Nachgewiesen durch	
Akzeptanzkriterium	
Abgeleitet von	ki-fkt-lernkomponente
Ursprung	
Referenzen	
Zuletzt geändert von	Daniel, 02.07.19
Status	offen
Begründung	Die Lernkomponente soll Abfahrtszeiten der Nutzer lernen und diese ausgeben.
Wird umgesetzt	

7. Fazit

Die zugrundeliegende Aufgabenstellung, der sich diese Projektgruppe widmet, ist die Entwicklung eines serienmäßigen Komfortsystems. Die Projektgruppe hat zur Realisierung der Aufgabenstellung einen umfangreichen Projektmanagementplan sowie ein Vorgehensmodell entworfen. Im Rahmen der ersten drei Monate hat die Projektgruppe sich dabei verschiedenen Herausforderungen fachlicher sowie persönlicher Inhalte gestellt (siehe Kapitel B Herausforderungskatalog). Verschiedene Maßnahmen und Methoden haben dazu geführt, dass sich die einzelnen Projektmitglieder gegenseitig besser verstehen und die eigenen Kompetenzen konkreter einschätzen können.

In der ersten Phase des Projektes wurde eine konkrete Vision und Aufgabenstellung sowie gut definierte User Stories und Use Case Diagramme erstellt. Auf Basis dieser konnte mit der Anforderungserhebung begonnen werden. Der zugehörige Meilenstein ist dabei aufgrund fehlerhafter Zeitabschätzung nicht erreicht worden. Problematisch in diesem Zyklus des Projekts ist mangelnde Erfahrungen der einzelnen Projektmitglieder in Bezug auf Aufwandsabschätzung und in bestimmten Bereichen des Projektmanagements gewesen.

Zusammenfassend lässt sich sagen, dass die erste Phase dieses Projekts jedes Projektmitglied sowohl gefordert als auch weitergebildet hat. Ausschlaggebend für diese erste Entwicklung ist das Leben eines neuen Vorgehensmodells und die damit verbundenen Aufgaben zur Fertigstellung der einzelnen Phasen. Jedes Projektmitglied ist sich über die persönlichen Chancen, die ein solcher Entwicklungsprozess mit sich bringt, bewusst und ist intrinsisch motiviert dieses über die weiteren Phasen des Projekts zu nutzen.

8. Ausblick

Im Folgenden wird vorgestellt, was die Projektgruppe CHILL 2.0 bis zum nächsten Review erreichen soll. Die Projektgruppe orientiert sich bei der Planung an dem, im Abschnitt 5.3.1 definierten, Vorgehensmodell. Die genaue Zeitplanung lässt sich der Tabelle 8.1 Zeitplan entnehmen. Zudem wurden weitere Meilensteine definiert, welche sich im Meilensteinkatalog befinden.

Tabelle 8.1.: Zeitplan

Task	Zeitspanne
Dokumentation	15.07.2019 - 31.03.2020
Anforderungserhebung	15.07.2019 - 02.09.2019
Hazard-Risk-Analyse	23.07.2019 - 16.09.2019
Systemarchitektur erstellen	29.07.2019 - 16.09.2019
Anforderungen reviewen	26.08.2019 - 02.09.2019
Entwicklung 1. Prototyp	16.09.2019 - 30.10.2019
Review Vorbereitung, erster Akzeptanztest	16.10.2019 - 30.10.2019
Implementierungs- und Testphase	30.10.2019 - 02.03.2020
Review Vorbereitung, Bug Fix, letzter Akzeptanztest	02.03.2020 - 31.03.2020

Anforderungserhebung Zum ersten Review wurde die Vision und die erste Version der User Stories und Use Cases fertiggestellt. Zudem wurden erste Anforderungen nach den definierten Qualitätskriterien erhoben. Auf dieser Basis kann nun die weitere Anforderungserhebung stattfinden und planungsmäßig Anfang September fertiggestellt werden. Für die Anforderungserhebung ist eine Review-Phase Ende August eingeplant, um die Qualität der Anforderungen sicherzustellen.

Systemarchitektur Im weiteren Vorgehen soll, neben der Anforderungserhebung, auch die Systemarchitektur erarbeitet werden. Der Umfang der Systemarchitektur gliedert sich in die drei Schritte *Funktionale Spezifikation*, *High Level Design* und *Low Level Design*. Die Systemarchitektur soll für die unterschiedlichen Spezifikationsniveaus, die Teilkomponenten und deren Schnittstellen festlegen. Die vollständige Definition der Systemarchitektur ist für die Entwicklungsphase wichtig, da ein einheitliches Verständnis der zu implementierenden Komponenten sichergestellt wird. Da es geplant ist, die Systemarchitektur neben der Anforderungserhebung zu erstellen, muss das Vorgehensmodell angepasst werden.

8. Ausblick

ISO 26262 Im weiteren Verlauf des Projekts wird der Standard ISO 26262 in den Entwicklungsprozess integriert, soweit dies der Rahmen des Projekts zulässt. Dazu wird eine Hazard-Risk-Analyse durchgeführt und ein Sicherheitskonzept erarbeitet. Dazu werden zu jedem Risiko, das nach ASIL klassifiziert wird, Sicherheitsanforderungen und Maßnahmen aus dem ISO abgeleitet. Das Sicherheitskonzept definiert, inwiefern diese Maßnahmen in den Entwicklungsprozess integriert werden. Die Hazard-Risk-Analyse erfolgt teilweise parallel zu der Anforderungserhebung und der Erstellung der Systemarchitektur.

Testmanagement Bis Mitte August ist ein Testplan zu erstellen, welcher den Testumfang, Qualitätskriterien für Testfälle und -implementierungen, Test-Tools und die Testumgebung festlegt. Des Weiteren ist eine enge Zusammenarbeit mit dem Sicherheitsmanager notwendig, um die Konformität *Ophelias* mit der ISO 26262, im von der Projektgruppe festgelegten Rahmen, zu gewährleisten. Durch die Erstellung einer Vorlage für Testfälle, Testimplementierungen und den Testbericht soll eine angemessene Dokumentation erzielt werden.

Erster Prototyp Zu dem zweiten Review soll ein vertikaler Prototyp entwickelt werden. Dieser Prototyp soll einen Aktor des Versuchsträgers über die Ophelia-App ansteuern können.

Intern Innerhalb der Gruppe soll eine positive Stimmung aufrecht erhalten werden. Durch eine offene Feedback-Kultur und durch die Sprint-Retrospektive wollen wir gewährleisten, dass Probleme direkt angesprochen und geklärt werden können.

Akronyme

CHILL 2.0 Configurating & Helping, Independent Learning Limousine.

KI künstliche Intelligenz.

OSRMT Open Source Requirements Management Tool.

PG Projektgruppe.

SME Subject Matter Expert.

VW Volkswagen.

Glossar

Aktuator Ein Aktuator ändert durch ein elektrisches Signal eine physikalische Größe wie zum Beispiel den Druck.

Aufgabe Aufgaben sind maximal heruntergebrochene Tätigkeiten, die im Jira dokumentiert und gemanagt werden und deren Aufwand abschätzbar ist.

Bedienelement Ein Bedienelement ermöglicht dem Nutzer die Steuerung eines Teilsystems des Fahrzeugs.

Buddy-System Ein von der Projektgruppe eingeführtes System, bei welchem ein Projektmitglied mit sprachlichen Einschränkungen mit einem weiteren Projektmitglied zusammen arbeitet, um etwaige Verständnisprobleme direkt lösen zu können.

CarPC Der CarPC ist ein Rechner vom Modell VTC 7240, der primär im Automobilberich eingesetzt wird. Zwei dieser Rechner stehen der Projektgruppe zur Verfügung. Einer davon ist im Versuchsträger verbaut.

dynamisch Dynamische Einstellungen umfassen alle Einstellungen, die von Ophelia mit Verwendung einer KI vor der Fahrt oder während der Fahrt durchgeführt werden (gelernt).

Epic Ein in Jira festgehaltenes Arbeitspaket, welches in einzelne Aufgaben unterteilt wird.

Epic-Beauftragter Projektgruppenmitglied, welches die Verantwortung zur Erfüllung aller Tasks des Epics innerhalb eines Sprints trägt.

Glossar Ein Glossar ist eine Liste von Wörtern mit beigefügten Erklärungen oder Übersetzungen.

No-Touch-Interaktion Nutzerinteraktion mit Ophelia durch Sprach- oder Gestesteuerung.

Nutzerprofil Sammlung von statischen und dynamischen Einstellungen mit Bezug zum Nutzer.

Glossar

Ophelia-Modul Das Software Modul des Systems Ophelia, welches auf dem Steuergerät des Fahrzeugs ausgeführt wird.

Regelmeeting Wöchentliches Pflichtmeeting mit der IAV und den Betreuern.

Scrum Master Moderator für Weekly Scrum und das Sprint Ende.

Sensor Ein Sensor ist ein Gerät, welches physikalische oder chemische Eigenschaften qualitativ erfassen kann und diese in elektrische Signale umformt.

Sitzungsleiter Der Sitzungsleiter leitet die Regelmeetings und die Arbeitstreffen.

Standardlautstärke Die Audiolautstärke eines Systems zum Systemstart. Wird das System abgeschaltet, nimmt es beim Einschalten die gesetzte Standardlautstärke als Lautstärke an.

statisch Statische Einstellungen umfassen alle Einstellungen, die von Ophelia ohne die Verwendung einer KI vor der Fahrt durchgeführt werden (nicht gelernt).

Subject Matter Expert Ein Subject Matter Expert ist im Rahmen der Projektgruppe ein Experte.

SWOT-Analyse Einordnen von Einflüssen auf das Projekt in die Klassen Stärke, Schwäche, Chance, Risiko.

Szenario Ein Szenario ist eine Sammlung von Einstellungen, die ein Nutzer vorab eingestellt hat.

Voreinstellung Eine Voreinstellung ist eine Einstellung, die vor Fahrtantritt automatisch durchgeführt wird.

Weekly Scrum Vorstellung und die Rekapitulation der in der Woche erledigten Aufgaben.

Zwischenziel Wenn eine Route mehr als ein Ziel beinhaltet, werden alle Ziele vor dem letzten Ziel als Zwischenziele bezeichnet.

Literaturverzeichnis

- [1] REEK, Felix: *Die Digitalisierung des Autos überfordert den Fahrer.* <https://www.sueddeutsche.de/auto/fahrzeugentwicklung-die-digitalisierung-des-autos-ueberfordert-den-fahrer-1.3507491>. Version: 2017
- [2] KLEIN, Rahel: *Hightech-Systeme in Autos: Viele Fahrer finden Technik-Schnickschnack überflüssig.* <https://www.wiwo.de/technologie/mobilitaet/hightech-systeme-in-autos-viele-fahrer finden-technik-schnickschnack-ueberfluessig/12248412.html>. Version: 2015
- [3] BÄHNISCH, Stephan: *Dekra-Studie zur Bedienung im Auto.* <https://www.autobild.de/artikel/dekra-studie-zur-bedienung-im-auto-3345908.html>. Version: 2012
- [4] SÜDDEUTSCHE ZEITUNG ; DPA ; PRAMSTALLER, Christopher ; REEK ; HART ; BAVO: *Sicherheitsrisiko Touchscreen im Auto.* <https://www.sueddeutsche.de/digital/digitale-technik-im-auto-sicherheitsrisiko-touchscreen-1.1910456>. Version: 2014
- [5] PLÖCHL, Martin: *Ein strukturierter Ansatz zur Anforderungserhebung von Kollaborationssystemen in Forschungseinrichtungen*, Technische Universität Graz, Diss.. – 86 S.
- [6] VWA: *Forschungsmethode: Leitfadengestütztes Interview.* http://www.ahs-vwa.at/pluginfile.php/2982/mod_page/content/123/leitfadengstutztesInterview.pdf. Version: 2013
- [7] AUTOHAUS EIHUSEN & WILKEN GMBH & Co. KG: *Upgrade in der Business-Klasse: die neue Audi A6 Limousine.* <https://www.eihusen-wilken.de/32.1/upgrade-in-der-business-klasse-die-neue-audi-a6-limousine.php>. Version: 2018
- [8] AUDI AG: *Audi Voice.* <https://www.audimv.com/faq/audi-voice-commands.html>
- [9] AG, Audi: *Head-up display.* https://www.audi-technology-portal.de/en/electrics-electronics/controls/head-up-display_en

Literaturverzeichnis

- [10] CRAIN COMMUNICATIONS GMBH: CES 2015: Audi zeigt lernendes Auto. (2015). <https://www.automobilwoche.de/article/20150106/AGENTURMELDUNGEN/301069916/ces--audi-zeigt-lernendes-auto>
- [11] BMW AG: *BMW Konfigurator.* <https://www.bmw.de/de/ssl/configurator.html#/>. Version: 2019
- [12] FORD MOTOR COMPANY: *Der Ford App Katalog.* <https://secure.ford.de/Rund-um-den-Service/Ford-SYNC/App-Katalog/>. Version: 2018
- [13] MICROSOFT: A Technical Companion to {Windows Embedded Automotive 7}. (2010), Nr. July
- [14] FORD MOTOR COMPANY: Ford SYNC 3 Sprachbefehle. Version: 2016. https://www.ford.de/content/dam/guxeu/de/documents/brochures/service/betriebsanleitungen-hilfe/sync/FordSync_Liste_der_Sprachbefehle_v2.pdf. 2016. – Forschungsbericht
- [15] APPLE: *Apple CarPlay Der perfekte Beifahrer.* <https://www.apple.com/de/ios/carplay/>. Version: 2019
- [16] ANDROID: *Android Auto.* https://www.android.com/intl/de_de/auto/. Version: 2019
- [17] FORD MOTOR COMPANY: *AppLink / SYNC.* <https://owner.ford.com/how-tos/sync-technology/sync-with-navigation/applink.html>
- [18] FORD MOTOR COMPANY: *Adjusting ambient lighting in SYNC 3.* <https://owner.ford.com/support/how-tos/sync/sync-3/settings/how-to-choose-ambient-light-color-with-sync-3.html>
- [19] FORD MOTOR COMPANY: *Ford® AppLink® App Catalog / Hands-Free Apps in Your Vehicle / Official Ford Owner Site.* <https://owner.ford.com/how-tos/sync-technology/all/applink/applink-app-catalog.html>
- [20] FORD MOTOR COMPANY: *FordPass Connect / Ford DE.* https://www.ford.de/content/dam/guxeu/de/shop/informieren/technologien/komfort/BR-fordpass_connect.pdf. Version: 2018
- [21] MERCEDES-BENZ: *Smartphone Integration.* <https://shop.mercedes-benz.com/de-de/connect/pdp/Smartphone-Integration/602?variantCode=QEY111AHWVDX&forceLoadTemplate=true&isInitialTransition=1>
- [22] MERCEDES-BENZ: *Mercedes-Benz Apps.* <https://www.mercedes-benz.com/de/mercedes-benz/lifestyle/mercedes-benz-apps/>
- [23] OPEL: *GRANDLAND X INFOTAINMENT.* <https://www.opel.de/opel-erleben/infotainment/infotainment-grandland-x.html>. Version: 2019

- [24] QUANZ, Philipp: *Infotainment der Zukunft: Insignia ist Vorreiter Opel-Blog.* <https://www.opel-blog.com/2018/06/04/infotainment-der-zukunft-insignia-ist-vorreiter/>. Version: 2018
- [25] OPEL: *Opel Grandland X Konfigurator.* <https://www.opel.de/fahrzeuge/grandland-x/konfigurator.html>. Version: 2019
- [26] PORSCHE: *MyPorsche.* <https://newsroom.porsche.com/fallback/de/themenwelt-porsche-digital/porsche-plattform-my-porsche-verknuepfung-online-offline-porsche-id-kunden-service-porsche-connect-panamera-cayenne-app-infotainment-14900.html>
- [27] PORSCHE AG ; PORSCHE DEUTSCHLAND GMBH: *Porsche Connect App.* https://connect-store.porsche.com/de/de/porsche-connect-app/p/bundle_connectapp_v2
- [28] PORSCHE DEUTSCHLAND GMBH ; PORSCHE AG: *Porsche Connect.* <https://www.porsche.com/germany/connect/>
- [29] PORSCHE SMART MOBILITY GMBH: *Navigation und Infotainment Paket.* https://connect-store.porsche.com/de/de/navigation---infotainment-paket/p/bundle_navigation_infotainment_v2
- [30] VERTICAL MEDIA GMBH: *Die coolsten Tech-Features des Tesla Model S.* <https://www.gruenderszene.de/galerie/features-vom-tesla-model-s?pid=11874%0A>
- [31] UG eomento: *Tesla arbeitet an Batterie-Vorheizfunktion, um Effizienz bei kaltem Wetter zu verbessern (Update).* <https://teslamag.de/news/tesla-batterie-vorheizfunktion-effizienz-wetter-17286>. Version: 2017
- [32] NUCCIO, Fabio D.: *Strukturwandel in der Automobilindustrie - Digitalisierte Mobilität und nachhaltige Ressourceneffizienz*, Diss., 2016
- [33] HEGMANS, Niels: *Ausstattung der verschiedenen Tesla Model 3 Versionen.* <https://www.model3.info/de/tesla-model-3-ausstattung%0A>. Version: 2019
- [34] TOYOTALEXINGTON: *2019 Toyota Sienna Infotainment Systems.* <https://www.lexingtontoyota.com/blog/what-technology-features-are-in-the-2019-toyota-sienna/>. Version: 2019
- [35] TOYOTASPEC19: *Toyota 2019 Specification.* <https://www.toyota.com/CHR/features/s/2402/2404/2408>. Version: 2019
- [36] VOLKSWAGEN AG: *Technik Lexikon A-E / Volkswagen Deutschland.* <https://www.volkswagen.de/de/technologie/technik-lexikon/a-e.html>. Version: 2019

Literaturverzeichnis

- [37] VOLKSWAGEN AG: *Technik Lexikon F-J / Volkswagen Deutschland*. <https://www.volkswagen.de/de/technologie/technik-lexikon/f-j.html>. Version: 2019
- [38] VOLKSWAGEN AG: *Technik Lexikon K-O / Volkswagen Deutschland*. <https://www.volkswagen.de/de/technologie/technik-lexikon/k-o.html>. Version: 2019
- [39] VOLKSWAGEN AG: *Technik Lexikon P-T / Volkswagen Deutschland*. <https://www.volkswagen.de/de/technologie/technik-lexikon/p-t.html>. Version: 2019
- [40] VOLKSWAGEN AG: *Technik Lexikon U-Z / Volkswagen Deutschland*. <https://www.volkswagen.de/de/technologie/technik-lexikon/u-z.html>. Version: 2019
- [41] PROJECT MANAGEMENT INSTITUTE INC.: *A Guide to the Project Management body of Knowledge (PMBOK® Guide)–Sixth edition*. 2017. <http://dx.doi.org/10.1002/pmj.20125>. – ISBN 978-1628251920
- [42] ANGERMEIER, Georg D.: *Liefergegenstand*. <https://www.projektmagazin.de/glossarterm/liefergegenstand>. Version: 2009
- [43] WINTER, Andreas: Wegweiser Anforderungen. (2016)
- [44] BOOMGAARDEN, Raphael ; BUHR, Moritz ; GRUNDT, Dominik ; HEBIG, Kimberly ; LIEBERT, Lynn ; LOOSE, Johann ; MARX, Wiebke ; MEINEN, Mario ; SAGER, Jens ; SCHUSTER, Patrick: Abschlussbericht PG CHILL / Universität Oldenburg. 2019. – Forschungsbericht
- [45] T2INFORMATIK: *Was ist eine Anforderung? – Wissen kompakt – t2informatik*. <https://t2informatik.de/wissen-kompakt/anforderung/>
- [46] HOOD GMBH: Hilfe zur Formulierung von Anforderungen / HOOD. Version: 2015. <http://www.hood-group.com/requirements/beratung/>. 2015. – Forschungsbericht. – 2 S.
- [47] AUGSTEN, Stephan: *Was ist das Wasserfallmodell?* <https://www.devinsider.de/was-ist-das-wasserfallmodell-a-680501/>
- [48] JOHNER, Christian: *V-Modell vs. Wasserfallmodell für Hardware- und Softwareentwicklung*. <https://www.johner-institut.de/blog/iec-62304-medizinische-software/v-modell/>
- [49] SCRUM GUIDES: *The Scrum Guide*. <https://www.scrumguides.org/scrum-guide.html>. Version: 2017
- [50] INA SCHAEFER: *Aufwandsabschätzung und Projektplanung*. <https://www.isf.cs.tu-bs.de/cms/teaching/2012w/se1/VL12.pdf>. Version: 2012

- [51] MEINSPIEL GMBH & Co. KG: *Planning Poker - Aufwandsschätzung in spielerischer Form*. <https://www.planningpoker.de/>
- [52] TRANTER, Leon: *Alternatives to Planning Poker*. <https://www.extremeuncertainty.com/alternatives-planning-poker/>
- [53] BOLES, Dietrich: Leitfaden zur Durchführung von Projektgruppen. 0 (2014)
- [54] GROLMANN FLORIAN: *Feedbackregeln: Feedback geben leicht gemacht*. <https://organisationsberatung.net/feedbackregeln-feedback-geben/>. Version: 2019
- [55] MICHAEL LANT: *Five Simple Steps to Agile Risk Management*. <https://michaellant.com/2010/06/04/five-simple-steps-to-agile-risk-management/>. Version: 2010
- [56] REDMINE: *Redmine Guide*. <https://www.redmine.org/guide>. Version: 2019
- [57] ATlassian: Jira. (2019). <https://de.atlassian.com/software/jira>
- [58] ATlassian: *Confluence – Kollaborationssoftware für Teams*. <https://de.atlassian.com/software/confluence>. Version: 2019
- [59] DEVELOPERS, OSRMT: *OSRMT Documentation*. <https://github.com/osrmt/osrmt>. Version: 2019
- [60] IO, Draw: *draw.io Online*. <https://about.draw.io/integrations/>. Version: 2018
- [61] INKSCAPE: InkScape launches version 0924. (2019). <https://inkscape.org/news/2019/01/17/inkscape-launches-version-0924/>
- [62] JENKINS.IO: Jenkins. <https://jenkins.io/>. Version: 2019
- [63] MENDELEY: *Mendeley Guides*. <https://www.mendeley.com/guides>. Version: 2019
- [64] GIT-SCM: *Git Documentation*. <https://git-scm.com/docs>. Version: 2019

A. Entscheidungskatalog

A.1. Team

- Wir führen eine offene Feedback-Kultur.
- Informationen, die formal die Gruppe betreffen, sind immer über den (zumindest internen) E-Mailverteiler zu verbreiten
- Rollen müssen definiert und zugewiesen werden. Die Rollenverteilung muss nicht zwangsläufig final sein. Es besteht die Möglichkeit die Rollen zu einem späteren Zeitpunkt nochmal umzuverteilen
- Es können nach eigenem Ermessen Kommunikationswege eingeführt werden

A.2. Termine

- Montags 8:15 Uhr bis 10:00 Uhr Gruppensitzung, Anwesenheitspflicht, Raum U104
 - unentschuldigtes Fehlen: 50 Euro Strafe; gültig ab dem 29.04.2019
 - unentschuldigtes Zuspätkommen (später als 8:15 Uhr): 5 Euro Strafe, iterative Steigerung von 5 Euro bei jedem weiteren Verstoß
 - Alle für die Gruppensitzung benötigten Dokumente und Unterlagen müssen vor Beginn der Sitzung im Repository auf dem Server vorhanden sein.
 - Bis Samstags 23:59 Uhr wird die Agenda für die kommende Sitzung veröffentlicht (siehe auch Konzepte, Ideen und Vorschläge)
- Montags 10:00 Uhr (bzw. nach Ende der Sitzung) bis 14:00 Uhr Arbeitstreffen
- Mittwochs 12:00 Uhr bis 20:00 Uhr Arbeitstreffen, Raum U104
- Freitag 12:00 Uhr bis 18:00 Uhr Arbeitstreffen
- Es gibt einen gemeinsamen Onlinekalender unter mail.uni-oldenburg.de
 - Dieser Kalender wird unregelmäßig (jedoch mindestens täglich) mit dem Kalender in Confluence synchronisiert

A. Entscheidungskatalog

- Zusätzlich gibt es im PG Gruppenraum einen Wandkalender in dem Termine festgehalten werden können

A.3. Protokoll

- Der Protokollant wird alphabetisch bestimmt
- Der Protokollant leitet das darauffolgende Meeting
- Das Protokoll wird mit Hilfe der LaTeX Vorlage *Dokumentation/Templates/protokoll_vorlage.tex* erstellt
- Das Protokoll im PD-Format und die Quelldatei werden in einem Ordner im Repository hinterlegt Protokolle
- Das Protokoll wird bis Dienstag 12:00 Uhr (bzw. Montag 24:00 Uhr) hochgeladen
- Es werden auch die Arbeitssitzungen protokolliert
 - Der Protokollant der Gruppensitzung ist für die Protokolle der ganzen Woche verantwortlich
- Der Protokollant soll den Entscheidungskatalog in der Woche pflegen
- Der Protokollant der aktuellen Woche dokumentiert die Sprint Retrospektive und dokumentiert den neuen Sprint im Sprinkatalog.

A.3.1. Inhalt

- Die Erfüllung der zum Meeting geplanten Aufgaben wird überprüft.

A.4. Dokumentation/Dokumente

- LaTeX wird für die Erstellung von Dokumenten verwendet.
- Redmine ist für offizielle Dokumente.
- Akzeptanztests werden nur für Anforderungen definiert, nicht für User Stories.

A.5. Anforderungserhebung

- Anforderungen sollen nach Vision, nicht nach Machbarkeit erstellt werden.
- Anforderungen zur Navigation bis zum 1. Review stehen.

A.6. Präsentationen

- Microsoft PowerPoint wird für die Erstellung von Präsentationen verwendet.
- Eine Vorlage für Präsentationen findet sich hier: *Dokumentation/Templates/Vorlage_Praesentation_Chill2.potx*.

A.7. Urlaub

- Jeder hat für den Zeitraum vom 01.04.2019 bis zum 29.02.2020 28 Urlaubstage zur freien Verfügung.
- Der Urlaub muss mindestens 30 Tage vor Beginn bei der Projektleitung beantragt werden.
- Gleitzeit ist zusätzlich erlaubt.

A.8. Vorgehensmodell

- Wir haben zusammen ein Vorgehensmodell für unser Projekt erarbeitet. Unser Modell basiert auf dem V-Modell und Scrum. Wir haben die Phasen des V-Modells in einen iterativen Prozess nach Scrum überführt.

A.9. Aufwandsabschätzung

- Zur Aufwandsabschätzung wird die Planning Poker-Methode mit den Fibonacci Zahlen angewendet.
- Die Aufwandsabschätzung wird für Use-Cases, die Anforderungserhebung und Implementierungsaufgaben verwendet.

A.10. Tätigkeitsberichte

- Wir haben uns dazu entschlossen die Tätigkeitsberichte zum Ende des Monats zu generieren.

A. Entscheidungskatalog

A.11. Aufgaben

- Aufgabenerfüllung wird von nun an bei jedem Meeting überprüft.
- Es wird eine Strichliste für nicht erledigte Aufgaben geführt. Diese wird von Noelle und Niklas geführt.
- Für jede Aufgabe soll ein Task in JIRA erstellt werden und dessen Aufwand abgeschätzt werden.
- Wichtige Aufgaben werden zugeteilt und alle anderen Aufgaben werden selbstständig zugewiesen.

A.12. Review Prozess

- Jedes Dokument soll von 2 Personen gereviewt werden.
- Die zweite Person soll immer Teil des Review Expertenteams sein.
- Der Reviewprozess soll mit dem JIRA Workflow übereinstimmen.

A.13. Tools

- Zum Zeichnen von Diagrammen verwenden wir draw.io.
- Für unser Logo und ähnliche Zeichnungen verwenden wir Inkscape.
- Wir verwenden Jira zur Organisation und Planung unserer Sprints, sowie der zugehörigen Aufgaben.
- Wir nutzen in Redmine das Wiki, um Ergebnisse, Entscheidungen und Daten festzuhalten, und das Ticketsystem für Tickets, die einen Bezug zur IAV haben.
- Wir haben uns dazu entschlossen Mendeley zur Unterstützung unserer Literaturrecherche zu verwenden.
- Wir haben uns für das Requirements Management Tool OSRMT entschieden, aber später entschieden darauf zu verzichten.
- Latex wird für die Anforderungserhebung als Tool verwendet.

B. Herausforderungskatalog

Tabelle B.1.: Herausforderungskatalog

Nr.	Beschreibung	Lösung
1	Unterschiedliche Herangehensweise bei der Erstellung von User Stories und Use Cases	Definition von Qualitätskriterien
2	Fehlende Kompetenzen im wissenschaftlichen Arbeiten	Definition von Qualitätskriterien und ein Review-Expertenteam
3	Unzureichende Kommunikation durch sprachliche Barrieren	Einführung des Buddy-Systems
4	Mangelnde Umsetzung des definierten Jira-Workflows	Epic-Beauftragter eingeführt

C. Meilensteinkatalog

Tabelle C.1.: Meilensteinkatalog

Nr.	Deadline	Titel	Beschreibung	Erreicht
1	27.05.19	Projektmanagementpläne v1.0	Fertigstellung der Projektmanagementpläne. Hierzu zählen die Pläne für das Umfangsmanagement, Anforderungsmanagement, Stakeholdermanagement, Zeitmanagement, Qualitätsmanagement, Ressourcenmanagement, Kommunikationsmanagement und Risikomanagement.	Ja
2	01.07.19	Anforderungen vollständig	Alle Anforderungen sollen erhoben sein.	Nein
3	16.09.19	Fertige Systemarchitektur	Alle Anforderungen sind erhoben und gereviewt. Zudem soll eine Funktionale Spezifikation, ein High Level und Low Level Design definiert sein. Ebenso wurde eine Hazard-Risk-Analyse durchgeführt.	
4	30.10.19	Erster Prototyp	Erster vertikaler Prototyp entwickelt. Dieser Prototyp soll einen Aktor des Versuchsträgers über die Ophelia-App ansteuern können.	

D. Meilensteinkontrolle

Tabelle D.1.: Meilensteinkontrolle

Nr.	Datum	Ergebnis
1	27.05.19	<i>Meilenstein 1: Projektpläne fertigstellen</i> wurde erreicht. Die Qualität der User Stories und Use Cases entspricht nicht den Erwartungen der Projektleitung. Ebenso waren diese unvollständig. Da diese das Fundament des <i>2. Meilensteins: Anforderungen fertigstellen</i> bilden, müssen diese zunächst angepasst und vervollständigt werden. Hierzu wurden Qualitätskriterien erstellt und ein Sprint zur Erledigung dieser Aufgaben durchgeführt.
2	10.06.19	<i>Meilenstein 2: Anforderungen fertigstellen</i> wird voraussichtlich nicht erreicht. Aufgrund des initial inhaltlich anders geplanten Sprints, zur Qualitätssicherung der User Stories und Use Cases, gingen zwei Wochen für die Erhebung der Anforderungen verloren. Zudem hatten die Projektgruppenmitglieder kein einheitliches Verständnis für die Erhebung von Anforderungen. Des Weiteren bot das im Voraus gewählte Tool zur Anforderungserhebung nicht alle für uns nötigen Funktionen. Als Resultat darauf musste eine neue Methode zur Anforderungserhebung gefunden und umgesetzt werden.
3	01.07.19	<i>Meilenstein 2: Anforderungen fertigstellen</i> wurde nicht erreicht. Es wurden für die weitere Arbeit Qualitätskriterien definiert und die Struktur der Anforderungen diskutiert. Die Projektgruppenmitglieder haben nun ein einheitliches Verständnis von der Anforderungserhebung. Die erste Version der User Stories und Use Cases sowie erste Anforderungen wurden fertiggestellt, sodass nun mit der Anforderungserhebung begonnen werden kann. Die Anforderungen sollen nun am 02.09.2019 erhoben sein.

E. Sprintkatalog

Tabelle E.1.: Sprintkalatog

Nr.	Zeitraum	Sprintziel	Storypoints	Abschlussanalyse
1	13.05.19 bis 27.05.19	Fertigstellung von Projektplänen, Visionsdokumentation, Userstories und Use Cases.	42/266	Fokussierteres Arbeiten, zur Qualitätssteigerung. Bessere Einhaltung des Workflows. Stärkeres Vertreten der Meinung der Gruppenmitglieder. Wissenschaftlicheres Arbeiten. Gewissenhafteres Reviewen, mit Fokus auf den Inhalt. Siehe auch Tabelle F.1 Retrospektive vom 27.05.2019.

E. Sprintkatalog

2	27.05.19 bis 10.06.19	Vorbereitung und Abhalten der IAV-Präsentation. Nacharbeitung der nicht erledigten Tasks des vorherigen Sprints.	181/195	Weniger in der großen Gruppe arbeiten. Mehr an Diskussionen teilnehmen. Weniger Ablenkung während der Gruppenarbeit. Wenn noch Arbeitskapazitäten übrig sind, selbstständig Aufgaben zuweisen. Wichtige Aufgaben werden weiterhin zugeteilt. Mehr Maßnahmen zum Teambuilding. Treffen besser vorbereiten (Wiki). Eine Agenda für jedes Treffen vorbereiten. Aufgaben erst abschließen bevor neue begonnen werden. Pausenzeiten während der Treffen konsequent umsetzen. Siehe auch Tabelle F.2 Retrospektive vom 12.06.2019.
---	--------------------------	--	---------	--

3	12.06.19 bis 01.07.19	Zwischenbericht und Präsentation für das 1. Review vorbereiten. User Stories und Use Cases abschließen. Mit der Erhebung der Anforderungen beginnen. Nacharbeitung der nicht erledigten Tasks des vorherigen Sprints.	129/433	Berichtsreview war umfangreicher als erwartet. Ein Projektgruppenmitglied hat die Projektgruppe verlassen und dies erst zum Sprintende erzählt. Zu viel Einarbeitung in neue Themen. Reviewprozess wurde nicht gelebt. Es soll mehr Zeit zur Arbeit eingeplant werden und an vorgegebene Storygrenzen halten.
---	-----------------------------	---	---------	---

F. Sprint Retrospektive

Tabelle F.1.: Retrospektive vom 27.05.2019

Weniger von	Mehr von	Aufhören mit	Starten mit	Weitermachen mit
Quantität	Qualität		Feedback Regeln einhalten	Arbeitsmoral
Metadiskussionen	eine eigene Meinung in Diskussionen vertreten		Arbeitspartner MJ/Jochen	grammatikalische Fehler in Reviews direkt umsetzen
Unqualifizierten Aussagen	Workflow einhalten		wissenschaftlich Schreiben	ergebnisorientiert Arbeiten
	Inhaltliche Reviews		Dokumentation umsetzen	
	Lob		Montagsmeetings vorbereiten	
	Klare Wortdefinitionen verwenden		Gewissenhaftes Review der eigenen Arbeit vor dem Pushen	
			Protokolle ansehen und notwendige Informationen herausziehen	

F. Sprint Retrospektive

Tabelle F.2.: Retrospektive vom 12.06.2019

Weniger von	Mehr von	Aufhören mit	Starten mit	Weitermachen mit
Gruppenarbeit in der ganzen Gruppe	Teilnahme an Diskussionen	Ablenkung während der Arbeitstreffen	sich selbstständig Aufgaben zuweisen	wichtige Aufgaben zuteilen
	Teambuilding		Nutzen der Agenda im Wiki für Montagsmeetings	Agenda für Treffen vorbereiten
	Aufgaben erst abschließen			
	Pausenzeiten konsequent umsetzen			

Tabelle F.3.: Retrospektive vom 01.07.2019

Weniger von	Mehr von	Aufhören mit	Starten mit	Weitermachen mit
Aufgaben während des Sprints hinzufügen	Motivation bei den Treffen	Konflikte über Whatsapp klären	Agieren statt reagieren	Weniger Abschweifen bei Meetings
Lange bei Gruppenaktivitäten labern	Zwischenstände liefern	Nachts puscheln, wenn reviewt werden muss	Kleingruppenarbeit	Mittwochstreffen
	Der rote Faden im Projekt	Off-Topic Diskussionen	Selbstständiges Arbeiten	Anzahl der Leute bei Gruppentreffen
	Updates bei Aufgaben	Aufgaben zu spät anfangen	Aussprechen von Unbehagen	
	Loben	Trödeln	Private Treffen	