

Testkonzept - Fahrerarbeitsplatz

Inhalt

1	Problemstellung	2
1.1	aktueller Stand	2
2	Begriffserklärung	3
2.1	Projekt	3
2.2	Testspezifikation	3
2.3	Testfall	3
2.4	Testkampagnen	3
3	Ziel des Projekts	4
3.1	Funktionsweise	4
3.2	Basisanforderungen ans Testkonzept	5
3.3	Vergleich mit ähnlichen Software Produkten	6
3.4	Rollendefinition	6
3.5	Use-Case – Diagramm	7
3.6	ER-Diagramm	8
4	Umsetzung	9
4.1	Komponenten	9
4.1.1	Configurator	9
4.1.2	Capture-Tool	10
4.1.3	Player	10
4.2	Blockdiagramm	11
4.3	Petri-Netz	12
5	Schwerpunkte	13
5.1	erster Prototyp	13
5.1.1	Spezifikation	13
5.1.2	Kommunikation/Datenaustausch Arbeitsplatzrechner – Notebook im Fahrzeug	13
5.1.3	GUI und Bedienung des Capture-Tools	13
5.1.4	Steuerung der Messung	13
5.1.5	Auswertung der Daten	13
5.2	spätere Erweiterungen (optionale Funktionen):	14

1 Problemstellung

Die Firma Berner & Mattner beschäftigt sich unter anderem mit dem Test von Fahrerassistenzsystemen (FAS). Diese stellen heute einen der Schwerpunkte in der Automobilentwicklung dar.

Obwohl bei den Tests dieser Systeme zumeist vollautomatische Testwerkzeuge, wie hardware-in-the-Loop-Prüfstände (HIL), zum Einsatz kommen, ist dennoch die Durchführung von Fahrversuchen unter realen Einsatzbedingungen unverzichtbar. Hierbei werden vorher genau festgelegte Fahrmanöver von einem Testfahrer in einem Testwagen durchgeführt. Diese Tests finden auf speziellen Teststrecken, aber auch auf öffentlichen Straßen statt. Als Fahrzeuge werden meist normale oder nur leicht modifizierte Serienfahrzeuge verwendet. Je nach Art des Tests bzw. des Testobjekts sind neben den Fahrmanövern auch verschiedene Randbedingungen (z.B. Wetter, Sicht, Fahrbahnbelag) fest vorgeschrieben oder müssen zumindest während des Tests protokolliert werden. Auch welche Messdaten von Interesse sind hängt stark von dem zu testenden FAS ab. Grundsätzlich können alle im Fahrzeug anfallenden Daten verschiedenster Sensoren (z.B. Geschwindigkeit, Abstand) und Steuergeräte (z.B. ESP, Motorsteuerung) ausgewertet werden.

Für die Durchführung solcher Tests im Fahrzeug existiert kein einheitliches durchgehendes Testkonzept. Es gibt eine starke Trennung zwischen der Spezifizierung von Testfällen, der Durchführung der Tests im Fahrzeug und der anschließenden Auswertung der Messdaten. So kann zwar die Spezifizierung der Tests durch hierfür geeignete Werkzeuge erfolgen, jedoch unterstützen diese im Allgemeinen nicht die Durchführung der Tests im Fahrzeug. Der Testfahrer hat oftmals nur eine Liste der geplanten Fahrmanöver zur Verfügung, welche er nacheinander abarbeitet. Es kann dabei zu erheblichen Abweichungen zwischen den geplanten und den tatsächlich gefahrenen Manövern kommen. Die eindeutige Dokumentation von Abweichungen, Problemen und Änderungen im Testablauf (z.B. Wiederholungen oder Auslassungen einzelner Tests) ist für die Auswertung der Tests von entscheidender Bedeutung, wird jedoch nicht optimal unterstützt. Nicht eindeutige Fahrmanöveranweisungen oder ungenaue Beschreibungen von Randbedingungen führen oft zu Interpretationsspielräumen auf Seiten des Testfahrers. Dies führt dazu, dass die Testergebnisse oft stark vom jeweiligen Testfahrer abhängig sind. Die Aufzeichnung der Messdaten erfolgt durch spezielle Werkzeuge getrennt von der Spezifikation und Durchführung der Tests. Die anschließende Zuordnung der gemessenen Daten zu den entsprechenden Testfällen ist sehr fehleranfällig.

1.1 aktueller Stand

- meist ist ein Beifahrer notwendig
- Beifahrer übernimmt die Bedienung eines Notebooks und ist für die Steuerung der Messung und Kommentierung zuständig. Die Messergebnisse werden alleinstehend gespeichert – kein Zusammenhang mit Spezifikation. Die Zuordnung der Messergebnisse und Kommentare zu den gefahrenen Tests erfolgt quasi per Hand (Notizen zu Zeit und Nummer der Messung/Testfall)
→ große Fehlerquelle: falsche Zuordnung Messdaten - Testfall
- Testspezifikationen als Excel-Tabellen oder Listen auf Papier
→ Anweisungen sind oft nicht eindeutig und werden unterschiedlich interpretiert, starke Abweichungen zwischen geplanten und gefahrenen Manövern und zwischen unterschiedlichen Fahrern.

→ siehe Beispiel – Testkatalog: Hinderniserkennung: [\\Applsrv03\Temp\Klowersa\Projekt - Fahrerarbeitsplatz\Beispiel - Testkatalog\EEFA2_APA_Testkatalog_V23E_20100531_SW.xls](#)

2 Begriffserklärung

2.1 Projekt

Ein Projekt ist die oberste Ordnungskategorie. Ein Projekt kann beispielsweise ein bestimmtes FAS (z.B. Auto Park Assist) behandeln und alle damit zusammenhängenden Testspezifikationen (Testfälle, Kampagnen, ...) beinhalten.

2.2 Testspezifikation

Eine Testspezifikation ist eine Sammlung verschiedener Testfälle und Testkampagnen. Dies kann auch eine Sammlung von Testfällen aus anderen Programmen sein (z.B. CTE XL). Die Testspezifikation enthält übergeordnete Informationen, z.B. Ziel und Eigenschaften der Tests.

Anmerkung: *genauere Eingrenzung Projekt – Testspezifikation – Testkampagne notwendig.*

2.3 Testfall

Testfälle sind einzelne Fahrmanöver oder eine Folge von mehreren Fahrmanövern. Sie haben das Ziel eine bestimmte Funktion eines FAS zu testen oder auch nur bestimmte Sensordaten unter im Testfall definierten Bedingungen aufzuzeichnen. Die einzelnen Fahrmanöver können komplex (einparken) sein oder auch aus nur einer Aktion (z.B. Vollbremsung) bestehen.

Ein Testfall enthält neben dem geplanten Fahrmanöver auch eine Folge von eindeutigen Fahrmanövern, welche dem Fahrer den Ablauf des Fahrmanövers genau beschreibt. Dabei sollte kein Raum für unterschiedliche Interpretationen der Anweisungen sein. Zudem sollte auch eine ausführliche Beschreibung von Ablauf und Ziel des Tests gegeben sein. Wenn möglich sollten auch schon die erwarteten Ergebnisse beschrieben sein.

Testfälle lassen sich nach verschiedensten Kategorien (Eigenschaften) unterscheiden:

- Vorbedingungen
- Randbedingungen (Sichtverhältnisse, Wetter, Fahrbahnbeschaffenheit, Fahrzeugzustand, ...)
- Testeigenschaften (benötigte Zeit,)
- Priorität
- Anzahl notwendiger/bereits abgeschlossener Testdurchläufe
- bisherige Erfolgsrate
- benötigte Fahrzeugausstattung/Sensorik
-

Anzahl und Art der Kategorien und ihrer Ausprägung (z.B. ja/nein oder [0...10] oder [trocken, nass, Schnee, Eis]) ist dabei völlig offen. Kriterien die für einen Testfall entscheidend sind, können für andere bedeutungslos bzw. gar nicht definiert sein.

2.4 Testkampagnen

Testkampagnen sind eine Zusammenstellung von Testfällen einer Spezifikation die innerhalb einer Testfahrt abgefahren werden sollen. Die Zusammenstellung ist frei wählbar. Sinnvoll scheint dabei die Zusammenfassung ähnlicher Tests bezüglich eines Kriteriums (gleiche Bedingungen, gleiches Ziel, Priorität) um diese dann gemeinsam zu testen.

➔ Siehe auch: ER-Diagramm

3 Ziel des Projekts

Ziel ist es ein durchgängiges Testkonzept zu entwickeln, welches eine optimale Unterstützung der Testvorbereitung, Testdurchführung und Testauswertung für Fahrzeugtests bietet. Dabei sollen die oben beschriebenen Probleme bei der Durchführung von Tests im Fahrzeug behoben werden.

Die Umsetzung soll durch evolutionäres Prototyping realisiert werden. Zunächst soll also nur ein Prototyp mit den Basisfunktionen entstehen, welcher dann sukzessive durch weitere Funktionen erweitert wird.

3.1 Funktionsweise

Die Software soll eine Möglichkeit bieten Testspezifikationen zu erstellen oder aus anderen Programmen zu importieren. Die Spezifikation sowie die nachträgliche Bearbeitung der Spezifikation soll möglichst komfortabel sein. Spezifikationen bestehen unter anderem aus einzelnen Testfällen.

Diese Testfälle können zu Testkampagnen zusammengestellt werden. Testkampagnen sind eine Folge von zusammengehörigen Testfällen die während einer Testfahrt abgefahren werden sollen. Die Auswahl von Testfällen zu einer Testkampagne kann nach unterschiedlichen Gesichtspunkten erfolgen. So können z.B. Testfälle mit dem gleichen Ziel (zu teste Funktion) oder gleichen Bedingungen (Fahrbahnbelag, Tageszeit, ...) in eine Kampagne zusammengefasst werden. Diese Auswahl kann aber auch nach Kriterien erfolgen, wie höchste Priorität, neuste oder bisher noch nicht bearbeitet Testfälle. Die Zusammenstellung von Testkampagnen soll daher durch Filterfunktionen unterstützt werden. Es sollte also möglich sein alle Testfälle nach den gewünschten Kriterien zu filtern bzw. zu suchen. Da sich diese Kategorien bei verschiedenen Testspezifikationen und auch zwischen einzelnen Testfällen stark unterscheiden können, ist hier eine generische Gestaltung der Spezifikation und der Filterfunktionen notwendig.

Anschließend werden die Testkampagnen auf einem im Fahrzeug befindlichen Rechner übertragen. Dort startet der Fahrer die Kampagne. Er erhält alle wichtigen Informationen über die geplante Testfahrt: Folge einzelner Testfälle, deren Ablauf, Bedingungen und erwartete Ergebnisse bzw. Ziel der Tests. Die Informationen werden möglichst übersichtlich auf einem Bildschirm dargestellt. Beim Fahren der einzelnen Tests soll der Fahrer sicher geleitet werden. Dies geschieht durch eindeutige, unmissverständliche Fahrhinweise vor oder während der Fahrt eines einzelnen Testfalls. Die Anweisungen könnten grafisch oder auch akustisch erfolgen. Außerdem soll der Fahrer die Möglichkeit haben den geplanten Testablauf zu editieren. Er kann die Reihenfolge der Testfälle verändern, Tests wiederholen oder auslassen wenn bestimmte Bedingungen nicht erfüllt sind. Außerdem sollen umfangreiche Möglichkeiten zur Kommentierung der Testfahrt gegeben werden, allgemein schriftlich oder auch akustisch während der Fahrt. Dazu gehört auch das Festhalten variierender Randbedingungen (äußere Bedingungen, ...) welche für die Auswertung von Interesse sein könnten.

Optional kann der Fahrer auch durch Automaten unterstützt werden. Durch vorher definierte Trigger werden während der Fahrt automatische Aktionen auf eingetretene Ereignisse ausgelöst oder dem Fahrer werden entsprechend Anweisungen signalisiert.

Die Messungen sollen weitgehend automatisch erfolgen. Die Kommunikation mit den Steuergeräten und Sensoren im Fahrzeug soll ohne Eingriffe des Testfahrers erfolgen. Die Messdaten werden sicher zusammen mit allen Fahrerkommentaren und sonstigen Daten (Loginfos, Abweichungen reales-geplantes Fahrmanöver, Fehlermeldungen, ...) in einem Messfile gespeichert. Zu jedem Testfall soll

bei jedem Durchlauf ein eigenes Messfile erzeugt werden. Dabei soll eine eindeutige Zuordnung von Messfile und Testfall gewährleistet werden.

Nach Ablauf der Testfahrt werden die gesamten Testergebnisse wieder auf einen normalen Arbeitsplatzrechner übertragen. Das Programm sollte hier umfangreiche Möglichkeiten zur Aufbereitung und Auswertung der Daten bieten.

3.2 Basisanforderungen ans Testkonzept

- durchgängige Testunterstützung ohne Brüche
 - eine Softwarelösung für Spezifikation – Testfahrt – Auswertung
- Unterstützung der Spezifikation von Testfällen
 - Importfunktion – Importieren von Spezifikationen oder Testfällen aus anderen Programmen (CTE XL, DOORS,...) [optional]
 - Unterstützung bei der Entwicklung bzw. Detailierung und Änderung von Tests (GUI)
 - einfache Spezifikation von Testfällen
 - festlegen von Ausschlusskriterien, notwendigen Vorbedingungen, wichtige Randbedingungen, Fahranweisungen, Prioritäten, Trigger, ...
 - komfortable Filter- und Suchfunktionen
 - generische Gestaltung
 - Einfache Festlegung der Testdurchführungspläne - Zusammenstellen von Testkampagnen
 - Filterung und Auswahl von Testfällen anhand festgelegter Kriterien (Priorität, Randbedingungen, ...)
- Unterstützung des Testfahrers (FAS)
 - Weitgehend automatisierte Basisdienste für die Bereitstellung von Testkampagnen im Fahrzeug
 - Optimale Führung des Fahrers beim Abfahren geplanter Fahrzeugtests
 - eindeutige Fahranweisungen (akustisch und/oder visuell)
 - Einfache Bedienung während der Fahrt (Touchscreen, Sprachsteuerung, ...)
 - Unterstützende Automaten (Trigger – automatische Aktionen) [optional]
 - Entlastung des Fahrers durch (teil-)automatische Steuerung der Messungen
 - Anzeige von Messtechnikdaten in größerem Umfang
 - Möglichst gute Reproduzierbarkeit der Tests, auch bei unterschiedlichen Fahrern
 - eindeutige Fahranweisungen (akustisch und/oder visuell)
 - Unterstützende Automaten (Trigger – automatische Aktionen)
- Automatisierung der Messung
 - automatisierte Basisdienste die sichere Speicherung und eindeutige Zuordnung der Testergebnisse und Messdaten zu ausgeführten Testfällen
 - Vollautomatische Aufzeichnung der Testergebnisse und Messdaten
- Unterstützung bei der Auswertung
 - Komfortables und schnelles Auffinden von Testergebnissen anhand benutzerdefinierter Kriterien
 - Offline Labeling
- Sonstiges
 - Grundfunktionen zur Fahrzeugdatenverwaltung (anlegen, suchen, ...)
 - Grundfunktionen der Benutzerverwaltung (Login, Zugriffsrechte, ...)

Anmerkung:

Es ist bisher noch nicht genau festgelegt worden, welche Funktionen bei den ersten Prototypen in welchem Umfang vorhanden sein sollten. Die obige Aufteilung in Basis und optionale Funktionen sollte vor Implementation noch detaillierter erfolgen. Auf einige Funktionen kann zunächst verzichtet werden oder es genügt diese Funktionen zu simulieren. (siehe auch 5.1)

TODO: genauere Spezifikation weiterer (nichtfunktionale) Anforderungen:

- Produktdaten: langfristig zu speichernde Daten aus Benutzersicht
- Produktleistungen: Anforderungen bezüglich Zeit und Genauigkeit
- Qualitätsanforderungen
- Benutzungsoberfläche: grundlegende Anforderungen, Zugriffsrechte
- Nichtfunktionale Anforderungen: einzuhaltende Gesetze und Normen, Sicherheitsanforderungen, Plattformabhängigkeiten
- technische Produktumgebung
- Schnittstellen
- Gliederung in Teilprodukte ?

➤ **Erstellung eines (groben) Pflichtenhefts**

3.3 Vergleich mit ähnlichen Software Produkten

- **TESTUS** – Ähnlichkeiten in Funktion und Aufbau aber anderer Kontext, eventuell hilfreich bezüglich innerer Struktur (Repräsentation der Daten), grafische Gestaltung und Bedienung. (siehe: <\\Applserv03\\Temp\\Klowersa\\Projekt - Fahrerarbeitsplatz\\verwante Software\\TESTUS\\>)
- **VerA** – eher nicht geeignet...? (siehe: <\\applserv03\\Temp\\Woelfle>)
- ...

Benutzerhandbuch TESTUS und Videos zu Vera befinden sich im Anhang bzw. auf dem Server

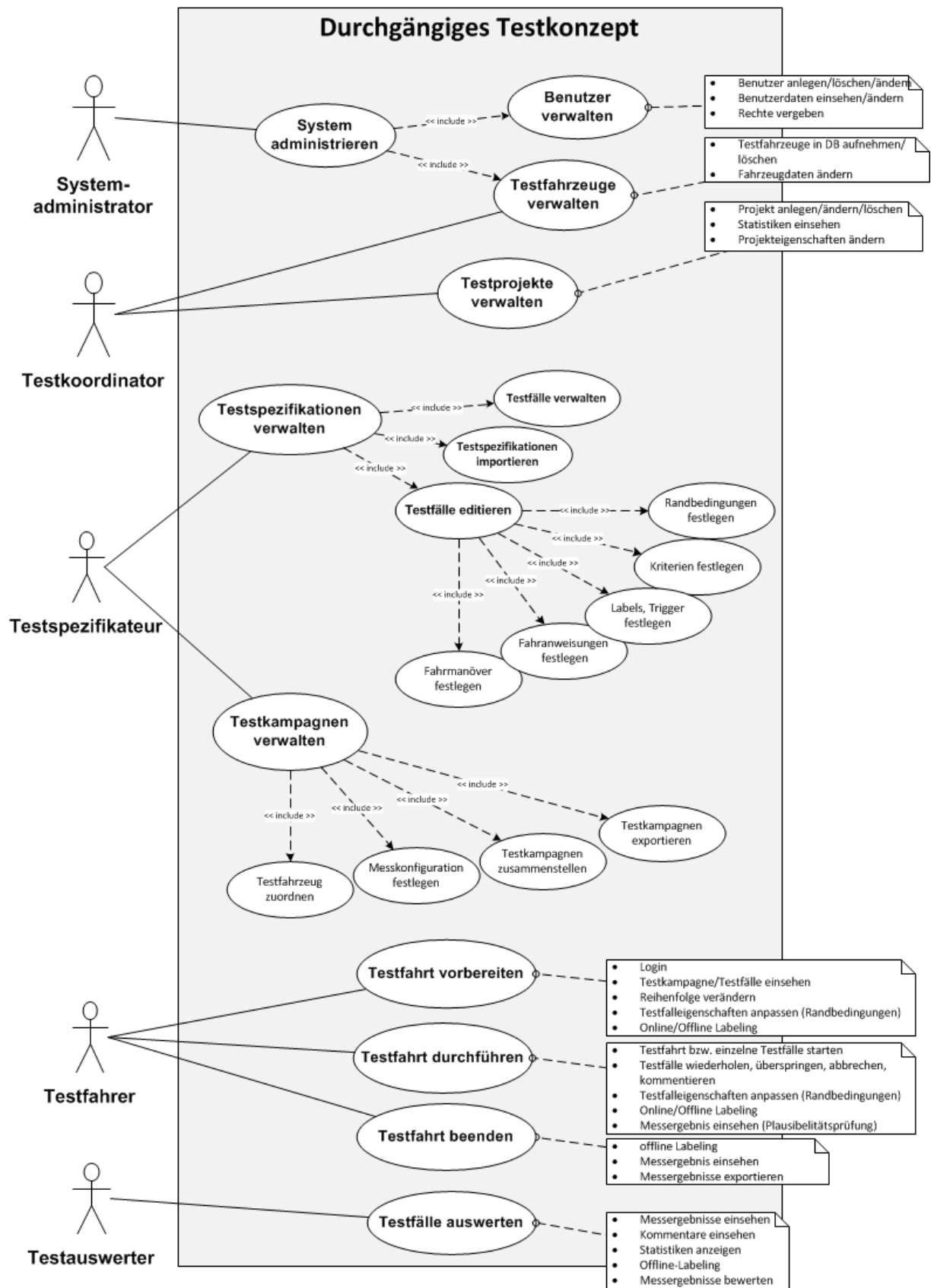
TODO: Weitere Lösungen finden und analysieren

3.4 Rollendefinition

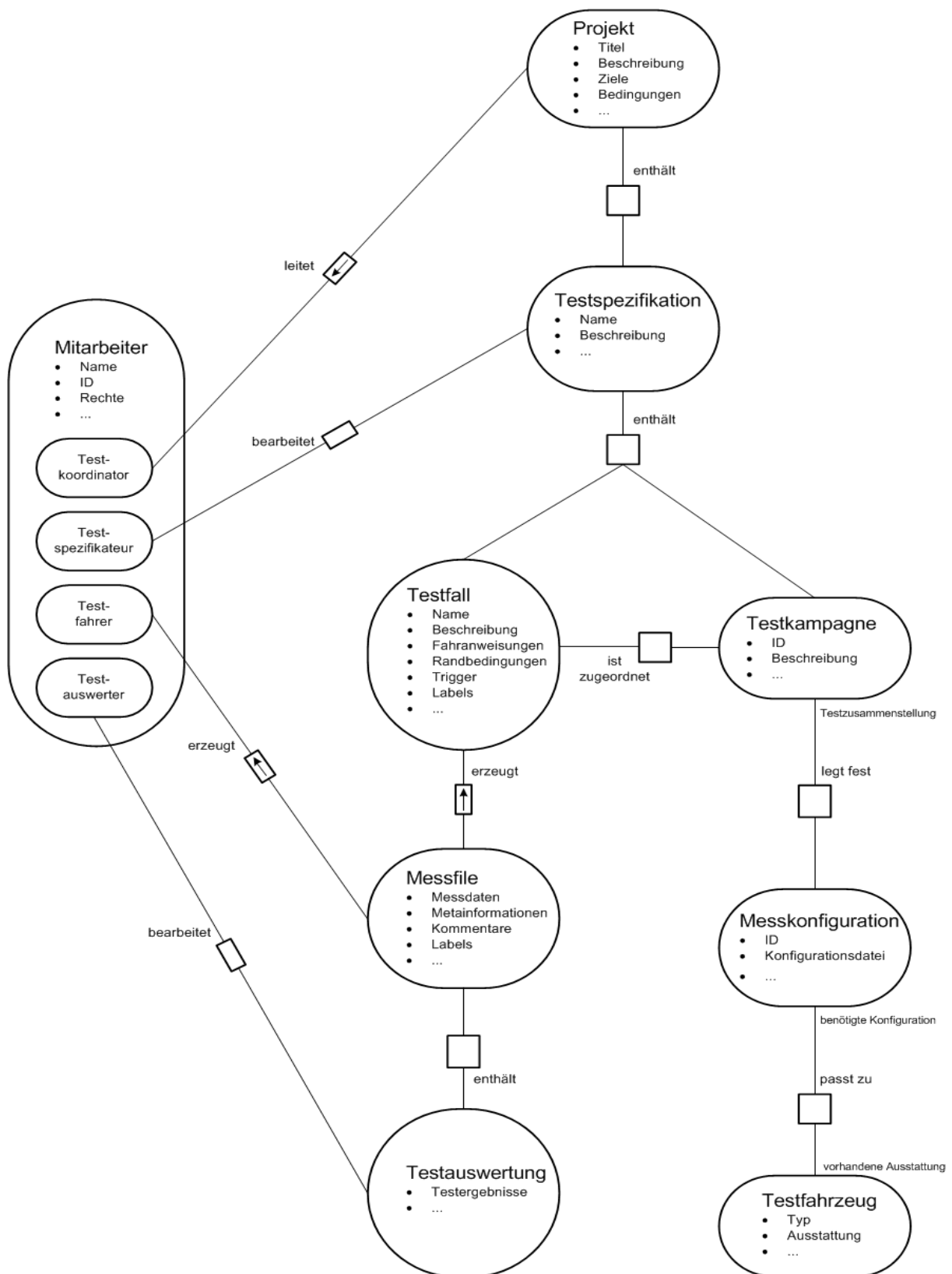
- Admin – allgemeine Administration, Verwaltung der Datenbanken, Benutzerverwaltung,
- Testkoordinator – Projektleitung, Projektverwaltung (evtl. auch gleichzeitig Admin)
- Testspezifikateur – Spezifikation Testfälle, Zusammenstellen von Kampagnen
- Testfahrer – Testdurchführung
- Testauswerter – Analyse der Messdaten

➔ Siehe: Use-Case-Diagramm

3.5 Use-Case - Diagramm



3.6 ER-Diagramm



ER-Diagramm basiert auf FMC (Fundamental Modeling Concepts)

Beziehungstyp: kein Pfeil - n zu m, ein Pfeil - 1 zu n, Doppelpfeil - 1 zu 1

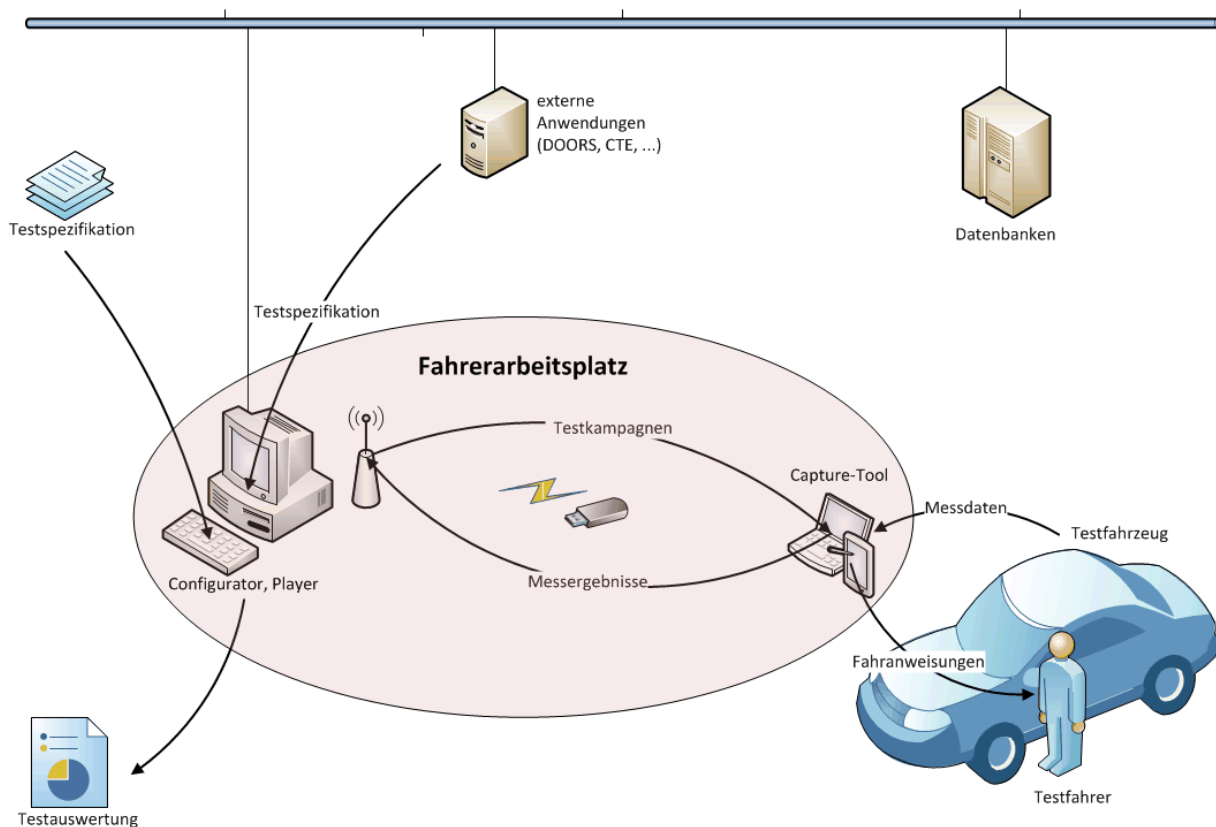
z.B. Ein Messfile gehört zu genau einem Testfall, ein Testfall kann jedoch mehrere Messfiles erzeugen (bei Wiederholung wird ein neues Messfile erzeugt).

4 Umsetzung

4.1 Komponenten

Die Grundidee ist die Umsetzung der verschiedenen Anforderungen in drei unterschiedlichen Hauptkomponenten:

1. Configurator
 - Spezifikation
 - Projektmanagement
2. Capture-Tool
 - Unterstützung des Fahrers
 - Steuerung der Messung
3. Player
 - Auswertung der Messdaten



4.1.1 Configurator

Mit dem Configurator erfolgt die Testspezifikation (anlegen, bearbeiten, löschen von Testspezifikation). Dies beinhaltet die Spezifikation der einzelnen Testfälle und das Zusammenstellen von Testkampagnen. Auch das Importieren fertiger Spezifikationen aus anderen Programmen soll dabei möglich sein. Außerdem sind die grundlegenden Funktionen zum Verwalten des Projekts (Projekt erstellen, löschen, Eigenschaften ändern, u.a.) hier angesiedelt.

Der Configurator soll auf einem Standard-Arbeitsplatzrechner laufen (fester Platz in einem Büro). Als Betriebssystem wird eine aktuelle Windows-Version angenommen. Idealerweise ist der Rechner über ein Netzwerk mit anderen Rechnern und externen Datenbanken (Fahrzeugdaten, Benutzerdaten)

verbunden. Alternativ könnten diese benötigten Daten aber auch zunächst auch nur lokal vorhanden sein. Die Importfunktion kann dabei über das Netzwerk aber auch über Speichermedien (CD, USB-Stick) erfolgen.

4.1.2 Capture-Tool

Das Capture-Tool sollte auf einem Notebook oder Notepad laufen, welches sich im Fahrzeug befindet. Mit dieser Komponente wird die Durchführung der Testfahrt gesteuert und die Messungen vorgenommen. Dazu wird zunächst eine vorher zusammengestellte Testkampagne aus dem Configurator importiert (USB-Stick oder Netzwerk) und anschließend gestartet.

Es wird der geplante Verlauf der Testfahrt angezeigt. Zudem gibt es schriftliche Erläuterung zu Ablauf, Ziel und Anforderungen jedes Testfalls. Der Ablaufplan kann vom Testfahrer noch an verändert werden – Anpassung an aktuelle Bedingungen (Deaktivieren einzelner Testfälle, Änderung der Reihenfolge, ...).

Während der Fahrt erhält der Fahrer zu jedem Testfall eindeutige Fahrhinweise. Dies kann optisch durch Text oder aussagekräftige Symbole (Pfeile, Verkehrszeichen) oder auch akustisch erfolgen. Der Testfahrer kann jederzeit in den Testablauf eingreifen. Testfälle können wiederholt, übersprungen, verschoben, editiert und kommentiert werden. Wichtig ist auch Aufzeichnung aktueller veränderter Randbedingungen oder sonstige Faktoren, die die Messung beeinflussen könnten. Neben allgemeinen Kommentaren kann dies durch ‚online Labeling‘ geschehen. Das bedeutet Kommentare können bestimmten Zeitpunkten (Labels) der Messung fest zugeordnet werden um den Zusammenhang mit entsprechenden Messwerten festzuhalten. Offene Randbedingungen sollten gezielt erfasst werden können (z.B. Sicht: hell, dunkel, neblig,...), ähnlich wie bei der Spezifikation der Testfälle.

Parallel zur Steuerung des Fahrablaufs, erfolgt auch die (voll-)automatische Steuerung der Messung. Das Capture-Tool muss dazu auf die Fahrzeugtechnik zugreifen können (ADTF, CAN, ...) und zu jedem Testfall die nötigen Sensordaten auslesen und sicher speichern. Die Speicherung erfolgt in einem Messfile (getrennt für jeden Testfall) zusammen mit möglichen Kommentaren des Fahrers und weiteren Metainformationen (Testfahrzeug, Fahrer-ID, Zeit, Anzahl-Wiederholung, ...). Es muss dabei eine eindeutige Zuordnung von gefahrenen Testfall und gespeicherten Messdaten gewährleistet sein. Auch sollten bereits während der Fahrt erste Ergebnisse wenn möglich und angezeigt werden (z.B. ausgewertet (z.B. Plausibilitätsprüfung der Messdaten -> Erfolg/Misserfolg). Der Fahrer kann dann darauf reagieren und den Testfall wiederholen oder entsprechend kommentieren.

Da die Komponente unter anderem der Information des Fahrers dient, sollte der Rechner entsprechend gut sichtbar im Fahrzeug installiert sein. Idealerweise soll auch eine einfache Bedienung während der Fahrt möglich sein (Touchpad, Lenkradfernbedienung, Spracherkennung, ...?). Das Design der GUI und Bedienung sind noch zu erarbeiten. Anfangs kann aber auch der einfachste Fall mit einem Standard-Notebook und der Bedienung durch einen Beifahrer angenommen werden.

4.1.3 Player

Der Player befindet sich wie der Configurator auf einem normalen Arbeitsplatzrechner. Player und Configurator sind logisch getrennte Komponente, können aber zusammen auf dem gleichen Rechner laufen. Der Player soll verschiedenste Möglichkeiten zur Aufbereitung, Analyse und Auswertung der Messdaten bieten. Dies beinhaltet Möglichkeiten zum Einsehen der Ergebnisse (grafische

Darstellung), zur Nachbearbeitung (offline Labeling) und Kommentierung der Messdaten sowie zum Erstellen von Statistiken. Genauere Anforderungen an diese Komponente sind noch zu definieren. Anfangs kann auch davon ausgegangen werden, dass die Messdaten nur sinnvoll aufbereitet und in einem allgemein nutzbaren Format (xml, ... ?) exportiert werden können, damit eine detaillierte Auswertung über andere Programme (Statistiksoftware, Tabellenkalkulation, Diagramme, ...?) erfolgen kann.

Anmerkung

Diese Aufteilung in drei ist nur ein erster Entwurf der auf der unterschiedlichen Aufgaben, sowie den verschiedenen Orten (lokaler Rechner, Laptop im Fahrzeug) basiert. Daran kann auch gut die Hauptfunktionalität erklärt werden.

In den folgenden Grafiken wird auch teilweise eine detailliertere Aufteilung der Komponenten angewendet:

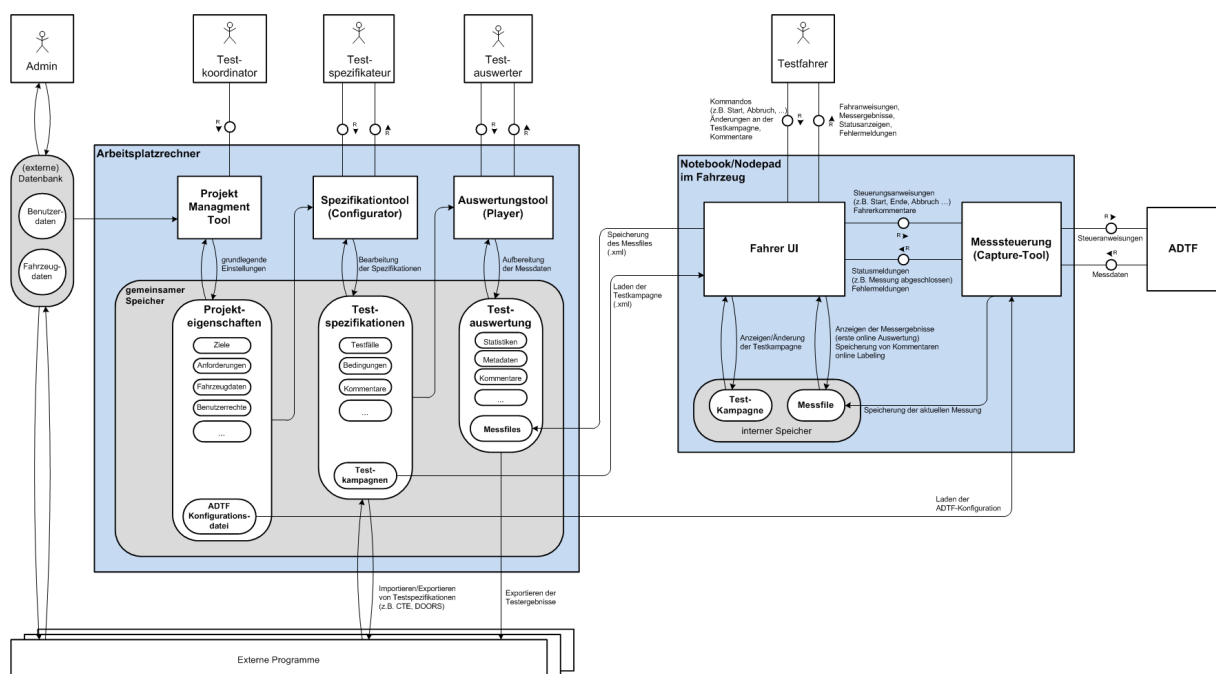
- Configurator – Trennung in ein Management-Tool und ein Spezifikations-Tool
- Capture-Tool – Trennung in Fahrer UI und Messsteuerung
- Player – alternative Bezeichnung als Auswertungs-Tool

Bei den Namen handelt es sich lediglich Arbeitstitel, sie können bzw. sollten noch angepasst werden

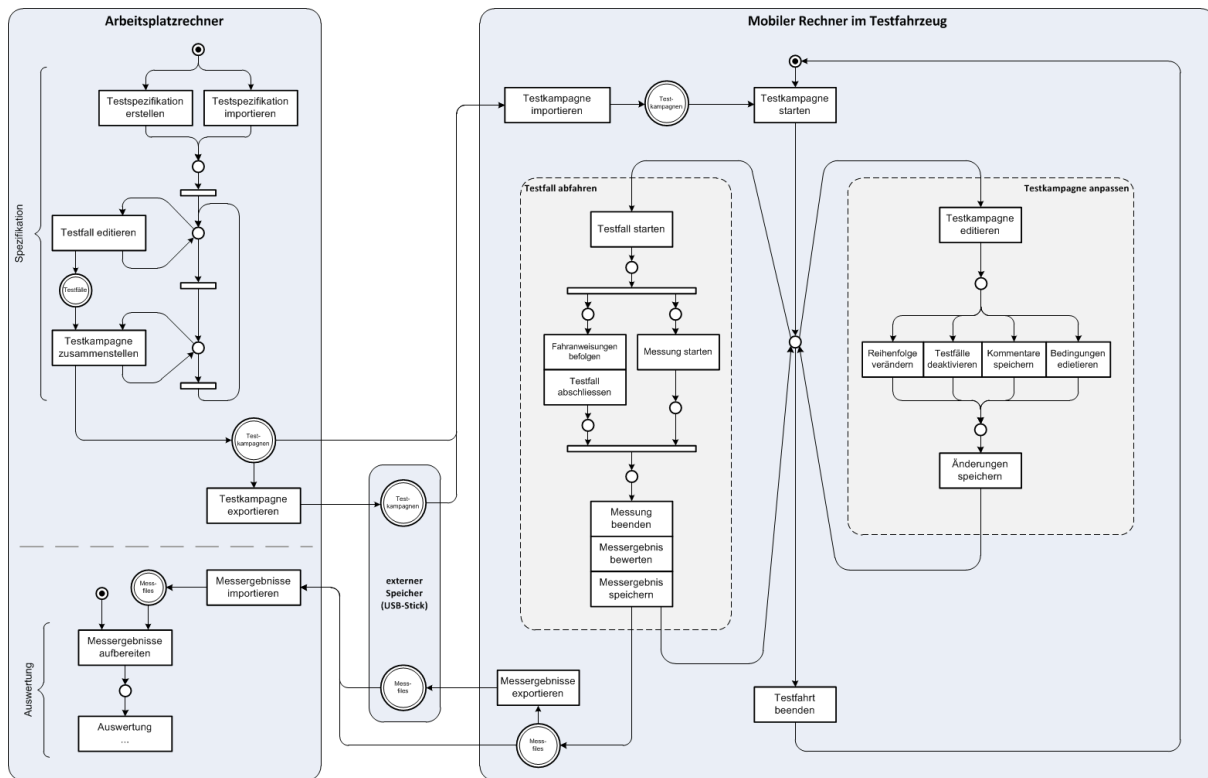
Programmstruktur und Programmablauf sind nur grob dargestellt. Es sind noch keine Implementationsdetails enthalten. Eine detailliertere Modellierung erfolgt zu späteren Zeitpunkten.

- *Grafiken basieren auf FMC (Fundamental Modeling Concepts)*

4.2 Blockdiagramm



4.3 Petri-Netz



5 Schwerpunkte

Aus der oben Beschriebenen geplanten Struktur und Funktionsweise ergeben sich folgende Schwerpunkte zur Implementation eines Prototyps:

5.1 erster Prototyp

5.1.1 Spezifikation

- Format/Darstellung der gesamten Testspezifikation (Testfälle, Testkampagnen)
- sinnvolle Parameter und Kriterien bei Spezifikation der Testfälle
 - statische Kategorien: Welche Kategorien sinnvoll bzw. immer vorhanden? (Priorität, Messerfolge, Anzahl absolvierter Versuche, ..?)
 - generische Gestaltung für Definition neuer Kategorien/Parameter
 - Filter- und Suchfunktionen

5.1.2 Kommunikation/Datenaustausch Arbeitsplatzrechner – Notebook im Fahrzeug

- Umfang und Format der Daten
 - xml ?
- Art der Kommunikation
 - Beschränkung auf Datenaustausch via USB-Stick

5.1.3 GUI und Bedienung des Capture-Tools

- Welche Informationen wie darstellen ?
 - Beschränkung auf wichtigste Infos
- Art der Bedienung
 - standart-Tastatur auf normalen Notebook
 - eventuell auch Touchhscreen - Touchpad

5.1.4 Steuerung der Messung

- Kommunikation mit Fahrzeugtechnik,
- Konfiguration der Messtechnik

➔ ADF, CAN, ...

5.1.5 Auswertung der Daten

- Welche Funktionen wichtig?
 - zunächst nur Funktion zum Extrahieren der Messdaten vorgesehen

5.2 spätere Erweiterungen (optionale Funktionen):

- Importfunktion von anderen Tools

Je nach Ausgangssoftware und damit zusammenhängenden Inhalt und Format müsste die jeweiligen Spezifikationen in ein nutzbares Format konvertiert und eventuell fehlende Daten ergänzt werden. Dies könnte durch einzelne Plugins für jede gewünschte Software geschehen. Diese Erweiterung kann zu späteren Zeitpunkten erfolgen, sollte aber beim Entwurf schon berücksichtigt werden. Sinnvoll wäre zunächst die Importfunktionen für den CTE XL zu verwirklichen, da dieses Produkt auch von Berner & Mattner stammt. Hier stehen Quellen, Spezifikationen oder eventuell auch ähnliche Tools zu Verfügung die Umsetzung der Importfunktion erleichtern.

- Erweiterung/Optimierung der GUI

Die einfache Bedienung (Basisfunktionalität) sollte durch komfortablere Methoden erweitert werden. Neben dem Touchscreen sind auch eine Sprachsteuerung oder die Bedienung über eine Lenkradfernbedienung denkbar. Auch die optische Gestaltung der GUI und dargestellten Informationen lassen sich zu späteren Zeitpunkten noch erweitern. Fahrhinweise könnten sowohl textuell, symbolisch und auch akustisch ausgegeben werden. Ziel sollte dabei eine einfache und komfortable Bedienung allein durch den Fahrer sein. Ein Beifahrer sollte zur Durchführung der Testfahrt nicht mehr benötigt werden.

- Erweiterte Möglichkeiten zur Auswertung, Analyse und Darstellung der Testergebnisse
- Erweiterte Fahrassistenten: Trigger – automatische Aktionen
- Erweiterung der Kommunikation zwischen Arbeitsplatzrechner und Notebook auf Netzwerkverbindung
- ...

Weitere Informationen auf dem Server: <\\\\Applserv03\\Temp\\Klowersa>