

IT4SCHOOL

Software Engineering

Prof. Dr. Klaus Schmid

schmid@sse.uni-hildesheim.de

Universität Hildesheim



Dr. Frank Houdek

frank.houdek@mercedes-benz.com

Mercedes-Benz AG



Software-Engineering – Was ist das?

Software Engineering \neq Programmieren

Software Engineering: Entwicklung von Software “im Großen”

- Große Software
- Viele Personen



Frage

Wie groß sind die größten Entwicklungsteams?

A 5-10 Personen

B 10-100 Personen

C 100-1000 Personen

D >1000 Personen



Software-Engineering – Was ist das?

Was ist notwendig um Arbeit im Großen zu organisieren?

Prinzipiell

Klare Zieldefinition

Lösungsskizze

Realisierung

Erfolg?

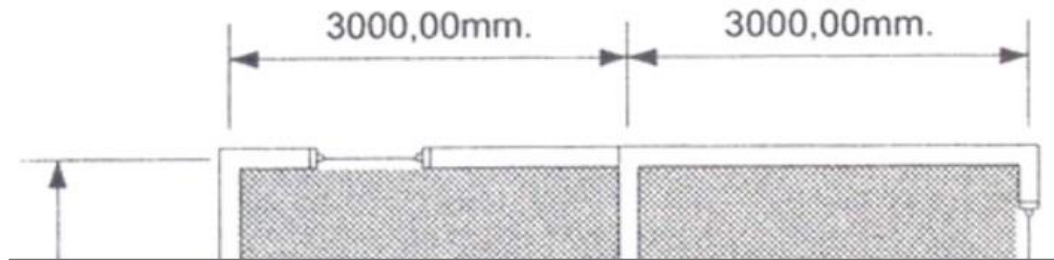
Gebäude

Raumdefinition

Architekturplan

“Bauen”

Abnahme



- Arten der Nutzfläche (Büro, Wohnen, etc.)
- Größe der Fläche
- Funktionen
- ...



Software-Engineering – Was ist das?

Wie sieht das im Software Engineering aus?

Prinzipiell

Klare Zieldefinition

Lösungsskizze

Realisierung

Erfolg?

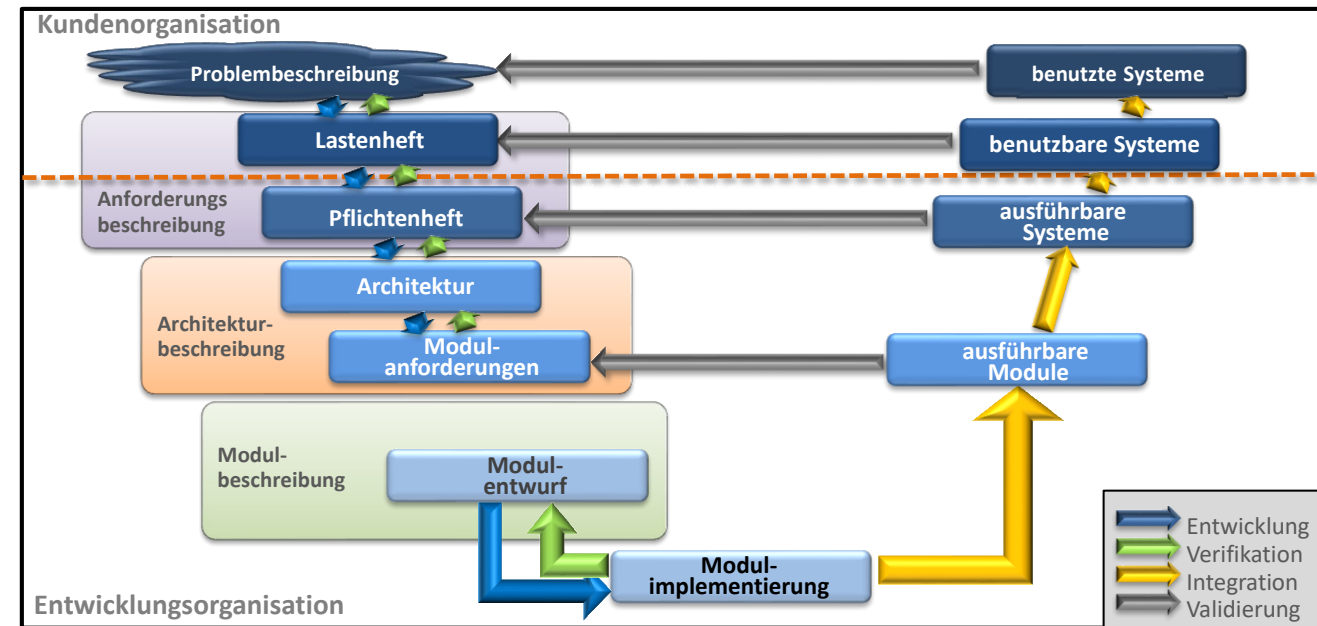
Software Engineering

Anforderungen

Architektur

Implementierung

Testen



Kundenorganisation

Problembeschreibung

Auftraggeber:
Mercedes-Benz AG

benutzte Systeme

Lastenheft

benutzbare Systeme

Anforderungs-
beschreibung

Pflichtenheft

Auftragnehmer:
Robert Bosch GmbH
Continental
ZF Friedrichshafen
...

ausführbare
Systeme

Architektur

Architektur-
beschreibung

Modul-
anforderungen

ausführbare
Module

Modul-
beschreibung

Modul-
entwurf

Modul-
implementierung

Entwicklungsorganisation



Beispiel: Sitzverstellung (1)



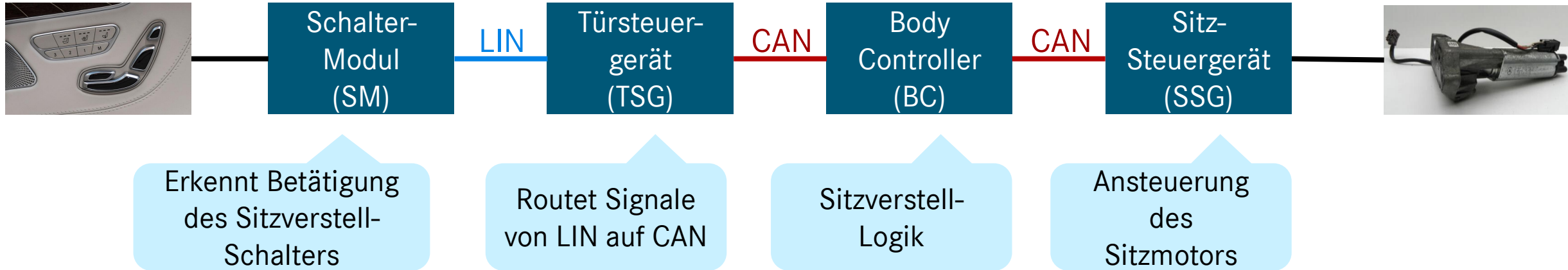
Abschnitt	Inhalt
Name	Sitzverstellung
Akteur	Fahrer, Beifahrer
Vorbedingung	Zündung ist an
Hauptscenario	<ol style="list-style-type: none">1. Fahrer betätigt Sitzverstell-Schalter (und hält den Schalter)2. Sitz verfährt in die gewünschte Richtung3. Fahrer lässt Sitzverstell-Schalter los4. Sitz stoppt
Alternativszenarien	<ol style="list-style-type: none">2a. Sitz erreicht Endposition und beendet Verfahren

Beschreibung der Funktion als Black-Box
(d.h. kein Wissen über den inneren Aufbau)
als Use Case

Beispiel: Sitzverstellung (2)



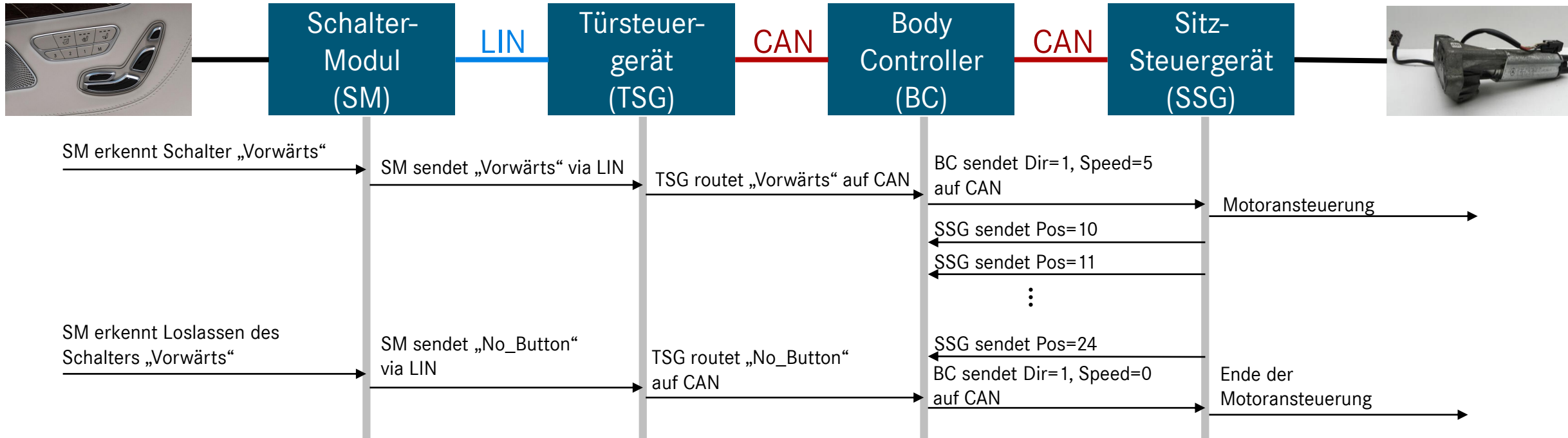
Technische Umsetzung



Beispiel: Sitzverstellung (2)



Technische Umsetzung



Beschreibung des Zusammenspiels der beteiligten Komponenten als MSC (Message Sequence Chart)
→ Ableiten der Anforderungen an die einzelnen Komponenten

Ein paar Zahlen ...



- Anzahl Steuergeräte: ca. 50-100 (je nach Modell und Sonderausstattung)
- Ca. 20 Kommunikationsbusse (CAN, Flexray, Ethernet), zzgl. einige LIN Busse
- Gewicht Leitungssatz: ca. 70 kg
- 50.000+ Signale

- Die meisten Steuergeräte werden von Lieferanten auf Basis von Komponenten-Lastenheften entwickelt

Frage

Warum gibt es so viele Steuergeräte in einem Fahrzeug?

A

Steuergeräte werden von anderen Baureihen übernommen. Dies verhindert das Zusammenfassen von Funktionen in einem Steuergerät

B

Ein Steuergerät hat nicht genug Rechenleistung, um die vielen Aufgaben in sich zu vereinen

C

Weniger Steuergeräte bedeutet mehr Leitungen und damit mehr Gewicht.

D

Wenige Zentralsteuergeräte bedeutet eine höhere Temperatur, und für diese steht nicht genügend Kühlleistung zur Verfügung



Beispiel: Sitzverstellung (3)

Anforderungen an das Schalter-Modul (SM)

...

SM_Req-34 Voraussetzung: Betriebsspannung liegt im Normalbereich, d.h. zwischen 10 und 15V.

SM_Req-38 Wenn Input_Taster_1 von 0 nach 1 wechselt und Input_Taster_2 = 0, sende Seat_Dir = 1.

SM_Req-13 Wenn Input_Taster_1 von 0 nach 1 wechselt und Input_Taster_2 = 1, lege DTC „Malefunction Input Buttons“ an.

SM_Req-39 Wenn Input_Taster_1 von 1 nach 0 wechselt, sende Seat_Dir = 0.

SM_Req-43 Wenn Input_Taster_1 für mehr als 2 Minuten den Wert 1 hat, ist davon auszugehen, dass der Taster hängt. In diesem Fall ist Seat_Dir = 0 zu senden, um die Sitzbewegung abubrechen.

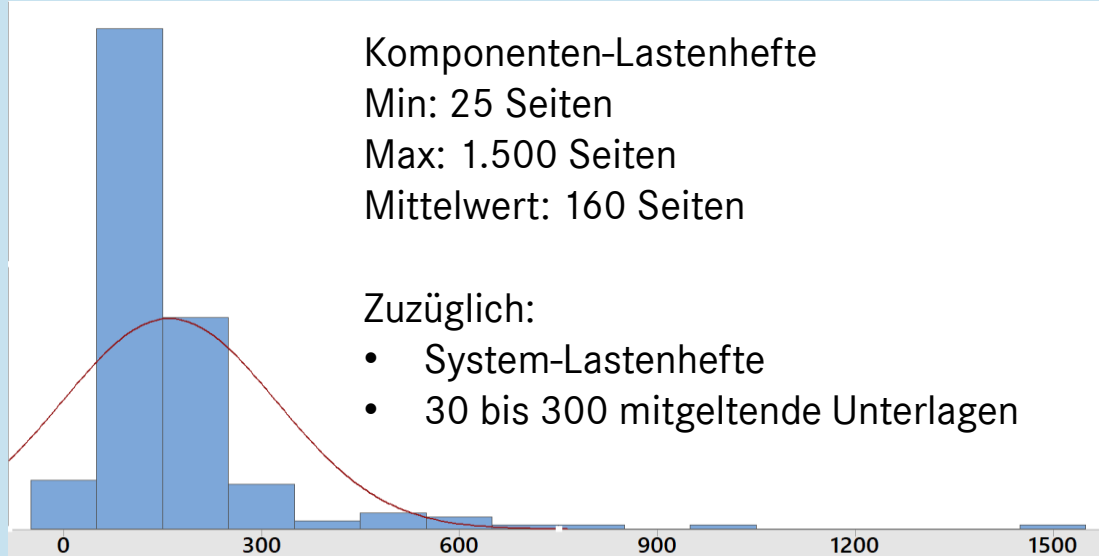
SM_Req_62 Sollte bereits beim Einschalten Input_Taster_1 = 1 anliegen, ist dieser Wert zu ignorieren, bis zum erstem Mal Input_Taster_1 = 0 anliegt.

...

DTC: Diagnostic Trouble Code (Eintrag im Fehlerspeicher)

Ein paar Zahlen ...

Spezifikationsgröße



Spezifikations-Stil



- Basis: Umfangreiches LH-Template (Standard-Texte)
- Im wesentlichen natürliche Sprache (Deutsch, Englisch)
- Kein Einsatz von Satz-Schablonen
- Wo angemessen:
 - Tabellen
 - Abbildungen, Zeichnungen
 - Zustandsautomaten
 - etc.
- Typischerweise keine formale Spezifikation
- Kennzeichnung: Anforderung, Information, Überschrift

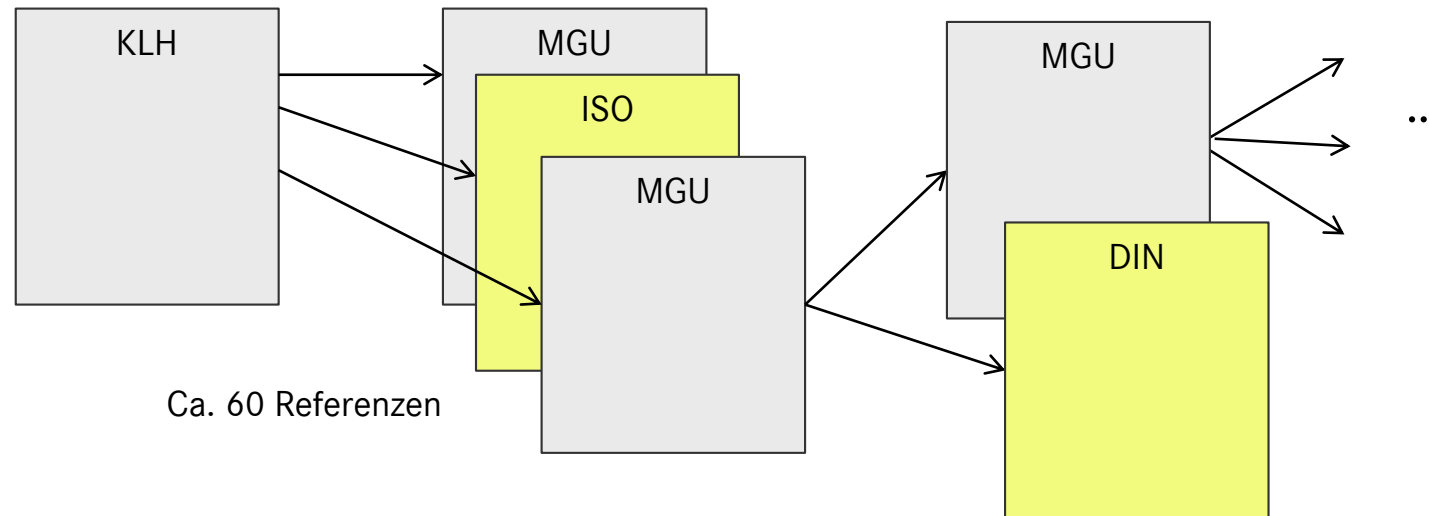
Pro Baureihe ca. 400 Komponenten-Lastenhefte → 64.000 Seiten Anforderungen

Ca. 10 Anforderungen pro Seite → 640.000 Anforderungen pro Baureihe

Das „echte“ Anforderungsdokument für den Sitzkomfort hat ca. 7.000 Anforderungen

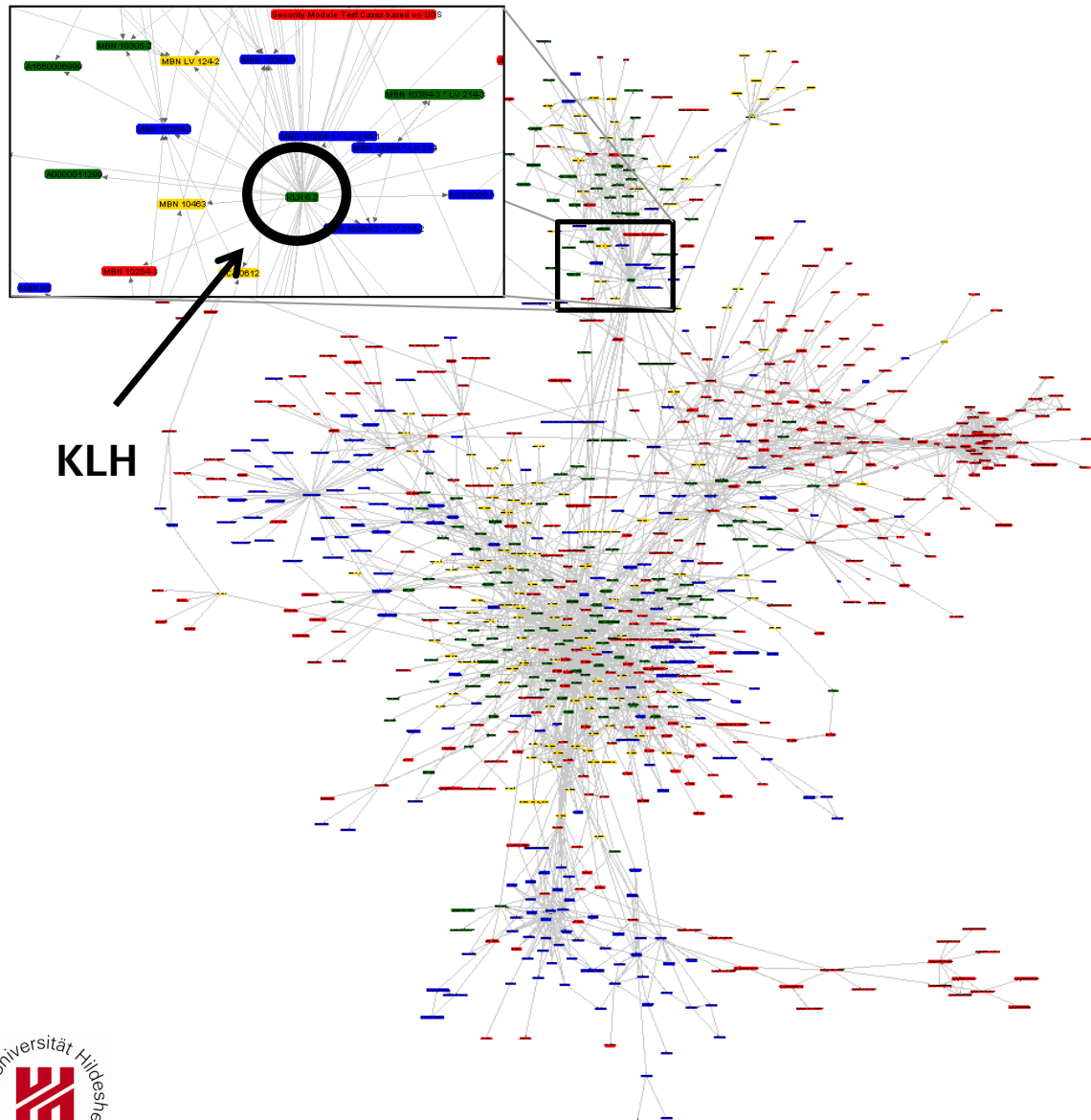
Mitgeltende Unterlagen

- In der Komponenten-Lastenheft-Vorlage werden ca. 60 mitgeltende Unterlagen angezogen
- Komponentenspezifisch wird diese Liste erweitert oder gekürzt



- **Schätzfrage:** Wie viele Seiten muss ein Lieferant an Mercedes-Benz spezifischen Anforderungen durcharbeiten?
- Abbruchkriterium: Externe Normen („Muss ein Lieferant ohnehin kennen“ = Stand der Technik)

Untersuchungsergebnisse aus 2013



Mehr als 2.000 Dokumente
Mehr als 50.000 Seiten

Davon ca. 15.000 Seiten
Mercedes-Benz spezifisch

Arten von Anforderungen

- Funktionale Anforderungen: Was soll das System genau machen?

SM_Req-39: Wenn Input_Taster_1 von 1 nach 0 wechselt, sende Seat_Dir = 0.

- Qualitätsanforderungen: Wie schnell, wie zuverlässig, usw. sollen die Funktionen durchgeführt werden

SM_Req-39-neu: Wenn Input_Taster_1 von 1 nach 0 wechselt, sende Seat_Dir = 0 **nach spätestens 10 msec.**

- Rahmenbedingungen: sonstige Einschränkungen an das System oder die Entwicklung des Systems

Die Software soll auf Windows 10 lauffähig sein.

Gute Anforderungen schreiben

- Missverständnisse in den Anforderungen führen zu fehlerhaften Systemen!
- Kriterien für gute Anforderungen (Anforderungsdokumente):
 - Atomar - immer nur eine Anforderung

SM_Req-39-neu: Wenn Input_Taster_1 von 1 nach 0 wechselt, sende Seat_Dir = 0 **nach spätestens 10 msec.**

- Ist die Anforderung atomar?

SM_Req-39-a: Wenn Input_Taster_1 von 1 nach 0 wechselt, sende Seat_Dir = 0.

SM_Req-39-b: Nach einem Wechsel des Tasterwerts muss Seat_Dir **nach spätestens 10 msec** den aktualisierten Wert annehmen.



Gute Anforderungen schreiben

- Kriterien für gute Anforderungen (Anforderungsdokumente) – Fortsetzung

- Präzise - das erwartete Verhalten sollte genau beschrieben werden.

SM_Req_62 Sollte bereits beim Einschalten Input_Taster_1 = 1 anliegen, *sollte* dieser Wert ignoriert werden, bis zum erstem Mal Input_Taster_1 = 0 anliegt.

- Eindeutig - es sollte nur eine Interpretation der Anforderung geben
- Widerspruchsfrei – es darf keine zwei Anforderungen geben, die einander widersprechen.

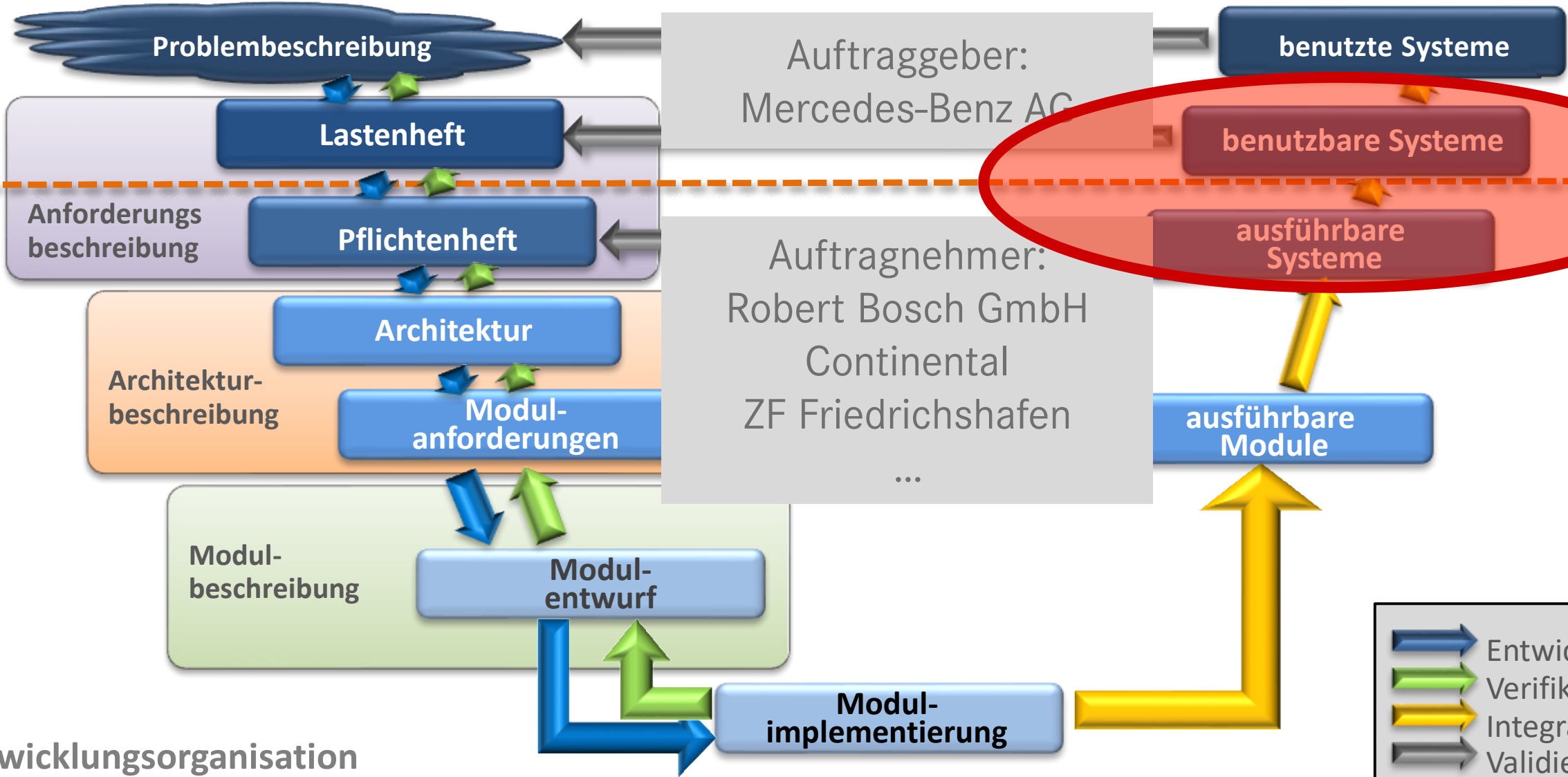
SM_Req-39a Wenn Input_Taster_1 von 1 nach 0 wechselt, sende Seat_Dir = 0.

SM_Req-39b Wenn Input_Taster_1 von 1 nach 0 wechselt, sende Seat_Dir = **1**.

- ...
- Testbar – es muss eine effiziente Vorgehensweise geben, um zu überprüfen ob die Anforderungen eingehalten wurden

TESTEN

Kundenorganisation



Entwicklungsorganisation

Testen

- Testen \neq Ausprobieren
 - Systematische Vorgehensweise
 - Wenn Fehler vorhanden: hohe Wahrscheinlichkeit sie zu finden
 - Möglichst geringer Aufwand

Was ist ein Test?

- Zu beschreiben
 - Zustand zuvor (falls notwendig)
 - Eingaben
 - Ausgaben (erwartete)
 - Zustand danach

Beispiel: Zum business4school Newsletter anmelden.

- Zustand zuvor: Noch nicht für Newsletter angemeldet
- Eingaben: Vorname, Nachname, Email
- Ausgaben: - Bestätigung der Anmeldung
- Begrüßungs-Email
- Zustand danach: Anmeldung ist in der Datenbank gespeichert

Arten von Testfällen

- Positivtest: es wird ein Fall mit gültigen Eingaben betrachtet

Beispiel: Zum business4school Newsletter anmelden.

Anmeldung mit korrekten Daten (bspw. Emailadresse), noch nicht angemeldet

- Negativtest: es wird ein Fall mit ungültigen Eingaben betrachtet

Beispiel: Zum business4school Newsletter anmelden.

Anmeldung mit inkorrekten Daten, beispielsweise:

- Email-adresse falsch geschrieben, oder
- Bereits registriert, oder
- Vor- oder Nachname nicht ausgefüllt, oder
- Emailadresse ist bereits mit anderem Namen registriert, oder
- ...

Übung: Erstellen von Testfällen

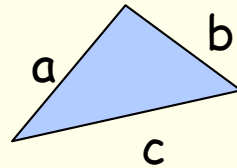
Gegeben: Programm Dreiecks-Test



Das Programm „Dreiecks-Test“ soll nach Eingabe der drei Kantenlängen a , b und c ausgeben, ob das durch die Kanten aufgespannte Dreieck ungleichseitig, gleichschenkelig oder gleichseitig ist.

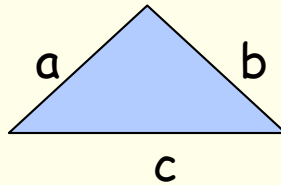
Mathe-Spickzettel:

Ungleichseitig



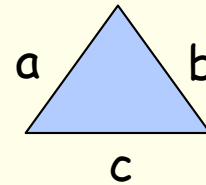
$$a \neq b \neq c \neq a$$

Gleichschenkelig



$$\begin{aligned} &a = b \neq c \text{ oder} \\ &a \neq b = c \text{ oder} \\ &a = c \neq b \end{aligned}$$

Gleichseitig

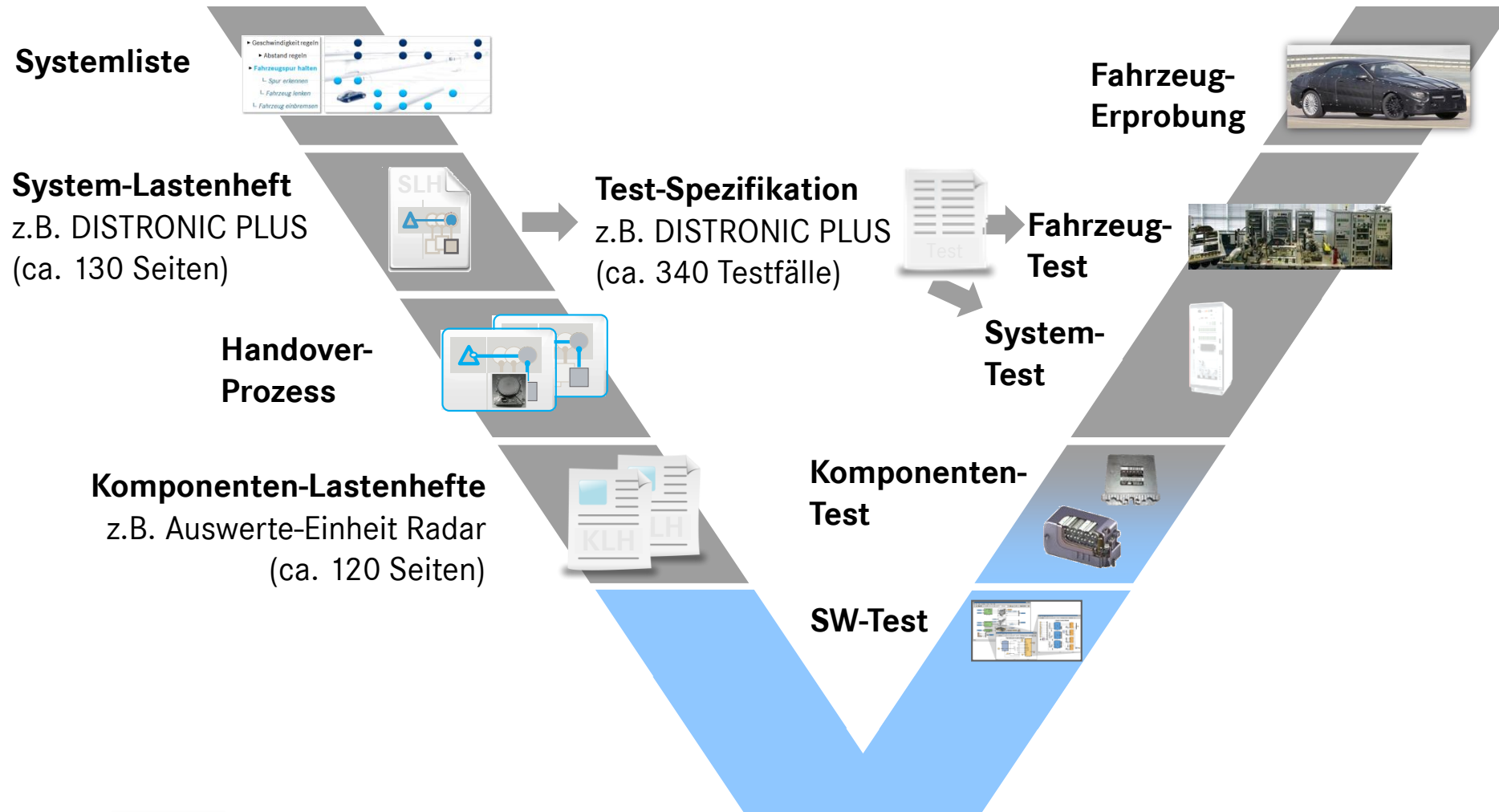


$$a = b = c$$

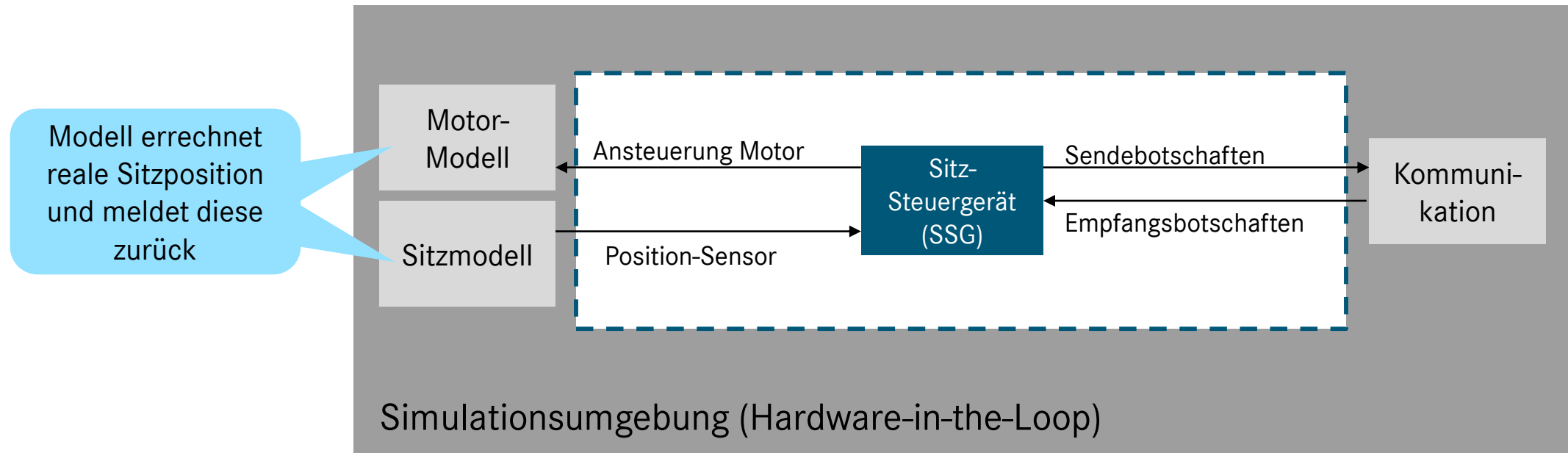


Erstelle Testfälle für das Programm „Dreiecks-Test“

Teststufen in der Automobil-Entwicklung



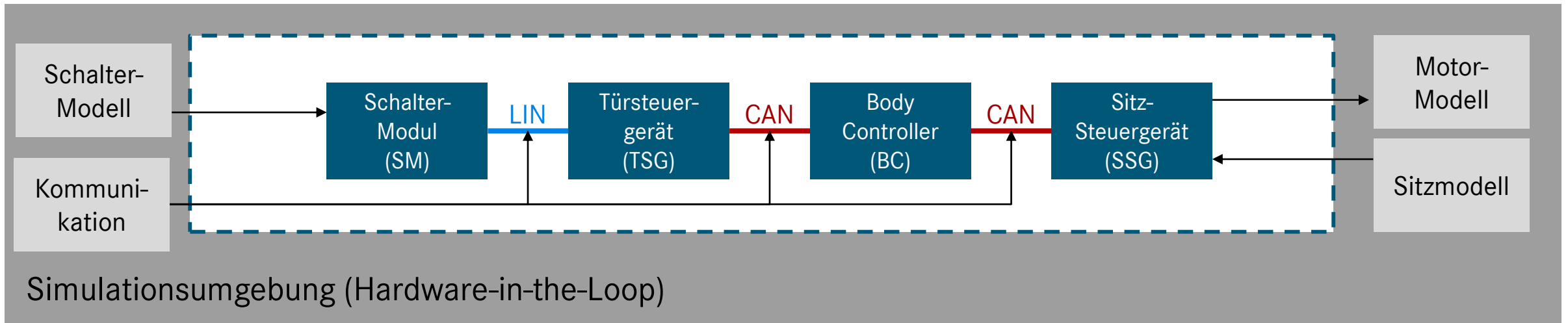
Teststufe Komponenten-Test



- Prüfling: Das echte Sitz-Steuergerät
- Umgebung wird vollständig simuliert
- Test-Ziel: Sicherstellen, dass das SSG die an es gestellten Anforderungen erfüllt

Teststufe System-Test

Voraussetzung: Erfolgreicher Komponenten-Test



- Prüfling: Alle Steuergeräte, die an der Sitzverstellung mitwirken
- Umgebung wird vollständig simuliert
- Test-Ziel: Sicherstellen, dass die Funktion der Sitzverstellung entsprechend den Anforderungen funktioniert

Beispiel: Testfälle für Sitzverstellung

Feld	Inhalt								
Name	Sitzverstellung Vorwärts								
Vorbedingung	Spannung = 13V, keine Einträge im Fehlerspeicher								
Testschritte	<table><thead><tr><th>Aktion</th><th>Erwartetes Ergebnis</th></tr></thead><tbody><tr><td>1) Zündung einschalten</td><td>LIN: Seat_Dir = 0</td></tr><tr><td>2) Setze Taster Vorwärts = 1</td><td>LIN: Seat_Dir = 1, Ansteuerung Motor mit Speed = 5 in Richtung Vorwärts</td></tr><tr><td>3) Setze Taster Vorwärts = 0</td><td>LIN: Seat_Dir = 0, Ansteuerung Motor mit Speed = 0 in Richtung Vorwärts</td></tr></tbody></table>	Aktion	Erwartetes Ergebnis	1) Zündung einschalten	LIN: Seat_Dir = 0	2) Setze Taster Vorwärts = 1	LIN: Seat_Dir = 1, Ansteuerung Motor mit Speed = 5 in Richtung Vorwärts	3) Setze Taster Vorwärts = 0	LIN: Seat_Dir = 0, Ansteuerung Motor mit Speed = 0 in Richtung Vorwärts
Aktion	Erwartetes Ergebnis								
1) Zündung einschalten	LIN: Seat_Dir = 0								
2) Setze Taster Vorwärts = 1	LIN: Seat_Dir = 1, Ansteuerung Motor mit Speed = 5 in Richtung Vorwärts								
3) Setze Taster Vorwärts = 0	LIN: Seat_Dir = 0, Ansteuerung Motor mit Speed = 0 in Richtung Vorwärts								

- Ideen für weitere Testfälle?



Ein paar Zahlen ...

Im Mittel hat ein Systemlastenheft **2.000** Items in der Datenbank
... davon sind **1.100** Anforderungen (55%)
... davon sind **835** testrelevante Anforderungen (75% aller Anforderungen)
... davon sind **440** HiL-relevante Anforderungen (40% aller Anforderungen)

HiL = Hardware-in-the-Loop

4,6 HiL Testfälle pro
Anforderung

4,9 Anforderungen pro
HiL Testfall

Im Mittel umfasst eine
Testspezifikation **340** HiL-relevante
Testfälle

Im Mittel hat ein Testfall vier
Testschritte

Das System Sitzkomfort besteht aus ca. 7.000
Anforderungen

Komponenten-Test: ca. 2.000 Testfälle
System-Test: Ca. 1.500 Testfälle

- Reale n:m Verknüpfung zwischen Anforderungen und Testfällen
- Hohe Standard-Abweichung: von 1:1 bis 1:200
- Anforderungen sind “vollständig” durch Tests abgedeckt

Zusammenfassung

- Software Engineering
 - Entwicklung „im Großen“
 - Definierte Qualität
- (Software-)System vs. Software
- Systematische Vorgehensweise
- Anforderungsanalyse / Requirements Engineering
 - Was sind Anforderungen?
 - Die Herausforderungen der Anforderungsanalyse bei Mercedes-Benz
- Testen / Feststellen der Software- & System-qualität
 - Wie sehen gute Tests aus?
 - Testen bei Mercedes-Benz



Hinweis (Werbeblock)

Bei Interesse an weiteren Informationen

- <https://t1p.de/inf-schueler>

Informatik-fuer-Schueler -- Newsletter zu Informatik-Veranstaltungen für Schülerinnen und Schüler

Über Informatik-fuer-Schueler

Deutsch

Neben all den Veranstaltungen, die die Universität Hildesheim für Studierende anbietet, bieten wir immer wieder verschiedene Informationen für Schülerinnen und Schüler an:

- Vorträge
- Workshops
- Sommerschule
- Studiengangsinformationen

Dieser spezielle Newsletter dient der Verteilung aktueller Informationen zu solchen Veranstaltungen an der Universität Hildesheim. Er weist ausschließlich auf solche Veranstaltungen hin von denen wir erwarten, dass sie potentiell für Dich interessant sein können. Insbesondere sind die Veranstaltungen immer für Schülerinnen und Schüler offen!

Die Nachrichten des Newsletters werden in unregelmäßigen Abständen und relativ selten (ca. 1-mal pro Monat) versendet. Natürlich kannst Du Dich jederzeit wieder abmelden, dann werden alle relevanten Daten von Dir gelöscht.

Abonnieren von Informatik-fuer-Schueler

Abonnieren Sie Informatik-fuer-Schueler, indem Sie das folgende Formular ausfüllen: Dies ist eine geschlossene Mailingliste, die nur für die Zwecke des Newsletters verwendet wird. Die Entscheidung des Moderators wird Ihnen per E-Mail mitgeteilt. Dies ist auch eine versteckte Mailingliste, was bedeutet, dass

Ihre E-Mail-Adresse:	<input type="text"/>
Ihr Name (optional):	<input type="text"/>
Sie können weiter unten ein Passwort eingeben. Dieses Passwort bietet nur eine geringe Sicherheit, sollte aber verhindern, dass andere Ihr Abonnement mit Ihrem wertvolles Passwort, da es ab und zu im Klartext an Sie geschickt wird!	
Wenn Sie kein Passwort eingeben, wird für Sie ein Zufallspasswort generiert und Ihnen zugeschickt, sobald Sie Ihr Abonnement bestätigt haben. Sie können Ihr Passwort per E-Mail zuschicken lassen, wenn Sie weiter unten die Seite zum ändern Ihrer persönlichen Einstellungen aufrufen.	
Wählen Sie ein Passwort:	<input type="password"/>
Erneute Eingabe zur Bestätigung:	<input type="password"/>
Welche Sprache bevorzugen Sie zur Benutzerführung?	Deutsch

Und wenn Ihr noch Fragen habt, Interesse an Schulbesuchen oder auch Praktika, gerne melden:

schmid@sse.uni-hildesheim.de