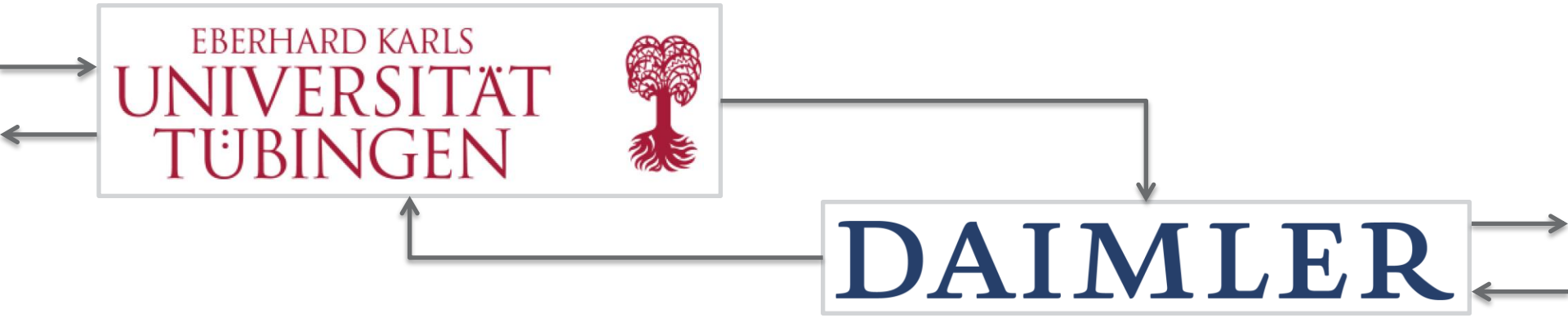


Automatisierte Qualitätsbewertung am Beispiel von MATLAB Simulink-Modellen in der Automobil-Domäne

Vortrag zur Verteidigung der Dissertation

Dipl. Inf. Jan Scheible

16.07.2012

Inhalt

Problemstellung

Lösungsansatz

Prototyp

Evaluation

Zusammenfassung und Ausblick

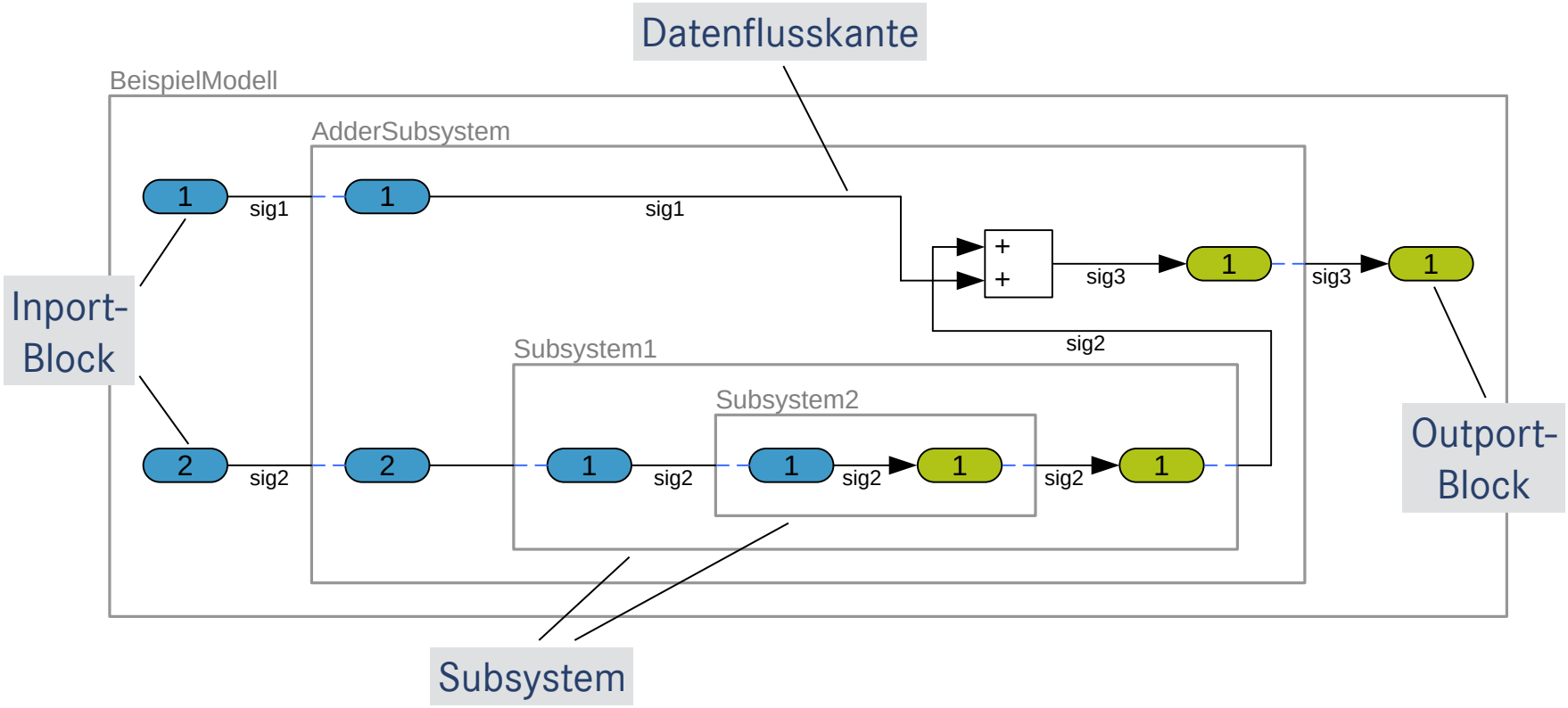
Ausgangssituation

- Automobilindustrie verwendet verstärkt modellbasierte Softwareentwicklung
- Umfang der Modelle wird immer größer
- Simulink-Modell ist das zentrale Arbeitsartefakt, daher hat die Modellqualität einen direkten Einfluss auf die Softwarequalität
- Entwickler haben trotz höherer Abstraktionsebene viele Freiheiten
- Ansätze zur Softwarequalitätsbewertung von eingebetteten Systemen konzentrieren sich bislang vor allem auf den Quellcode

Ziel

Entwicklung eines Verfahrens zur automatisierten Qualitätsbewertung von Simulink-Modellen

MATLAB Simulink



- Simulink-Modelle sind kombinierte Daten- und Kontrollflussdiagramme
- keine explizite Unterscheidung zwischen dem Platform Independent Model (PIM) und dem Platform Specific Model (PSM)
- mithilfe eines Codegenerators wird C-Code für die Zielsteuergeräte erzeugt

Inhalt

Problemstellung

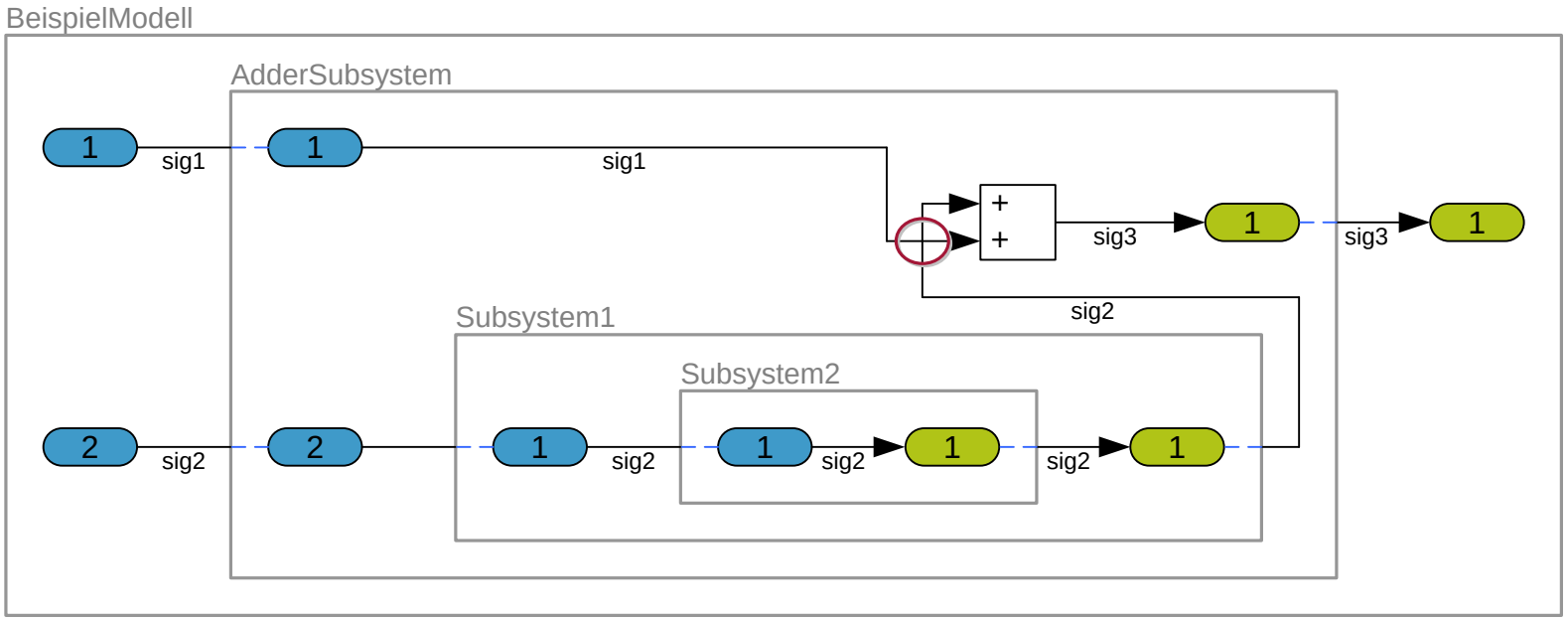
Lösungsansatz

Prototyp

Evaluation

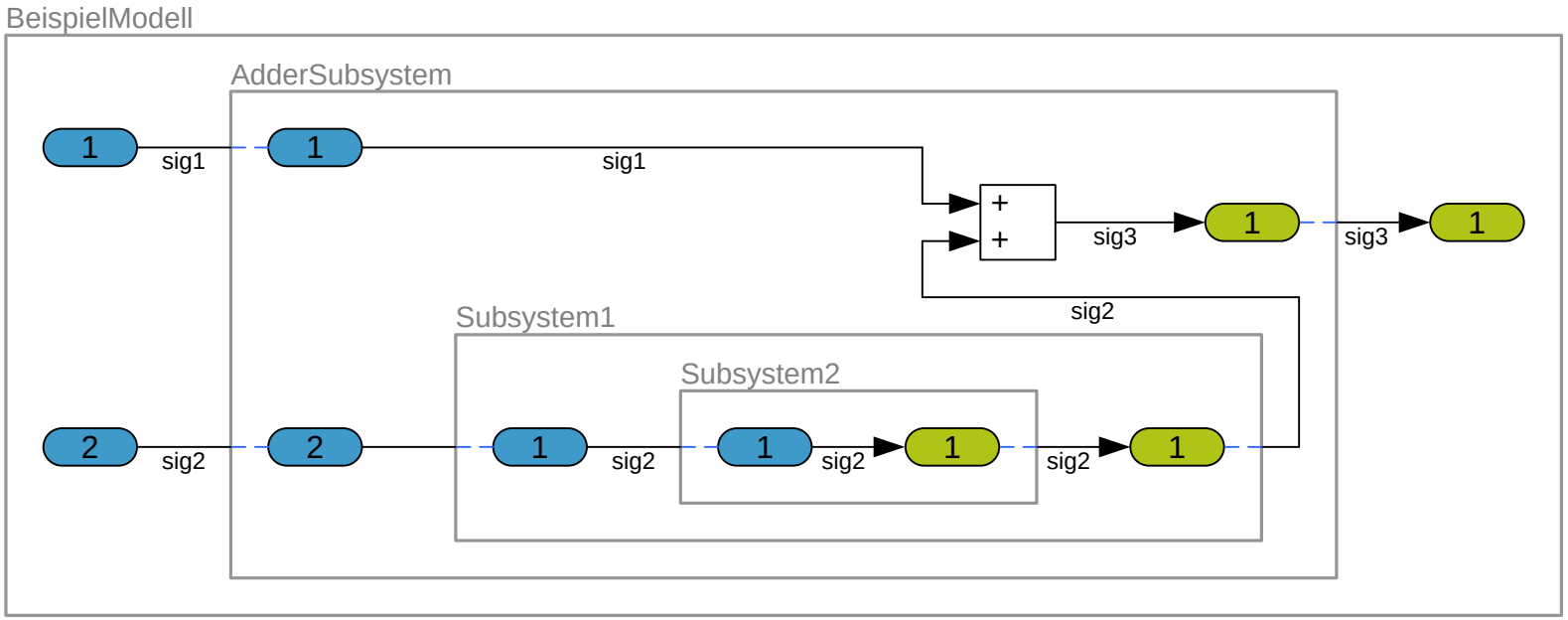
Zusammenfassung und Ausblick

Metriken für Simulink-Modelle



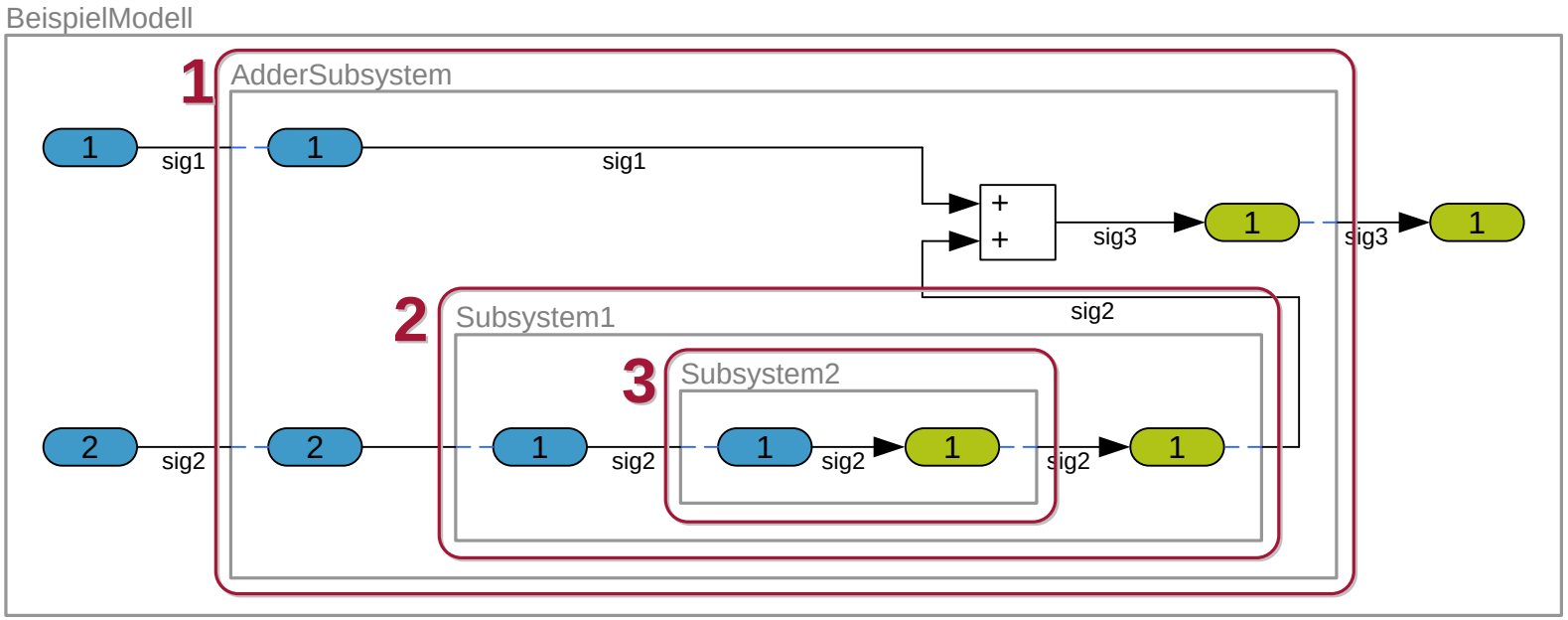
Anzahl der überkreuzenden Linien = 1

Metriken für Simulink-Modelle



Anzahl der überkreuzenden Linien = 0 (1)

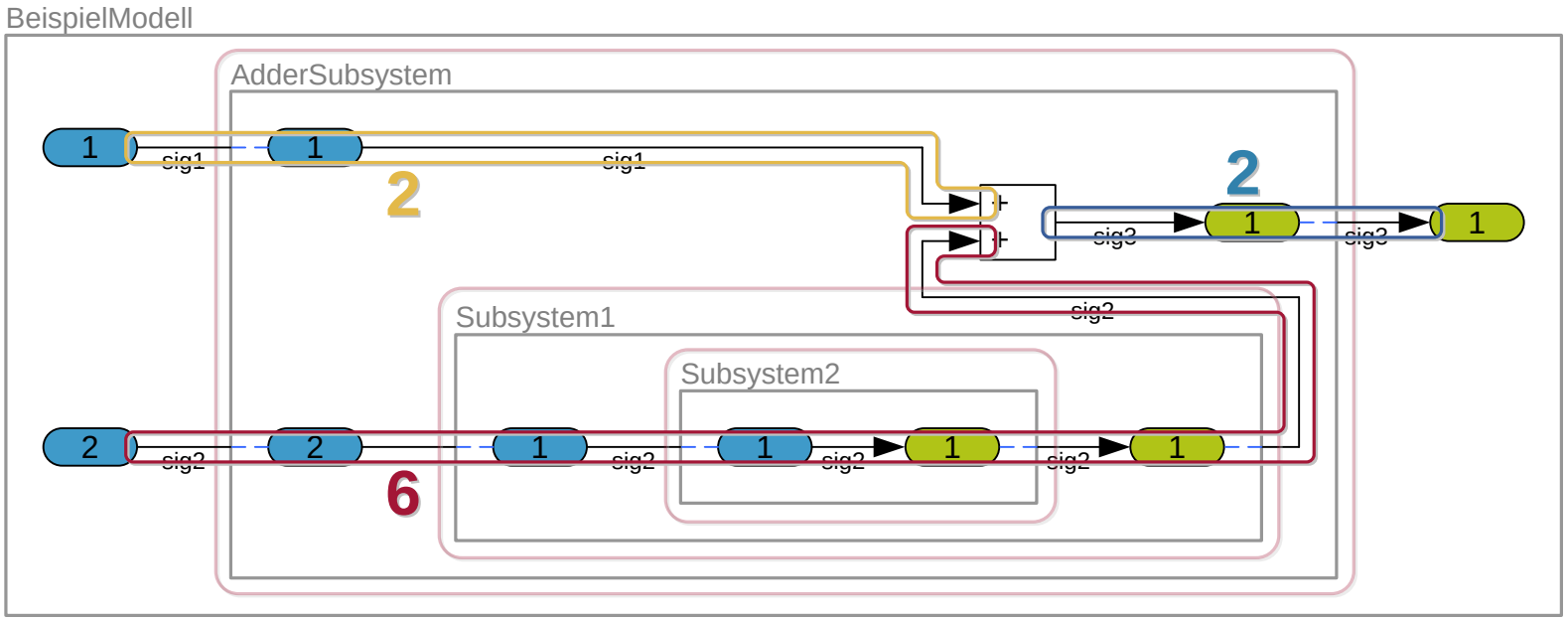
Metriken für Simulink-Modelle



Anzahl der überkreuzenden Linien = 0 (1)

Tiefe der Subsystem-Hierarchie = 3

Metriken für Simulink-Modelle

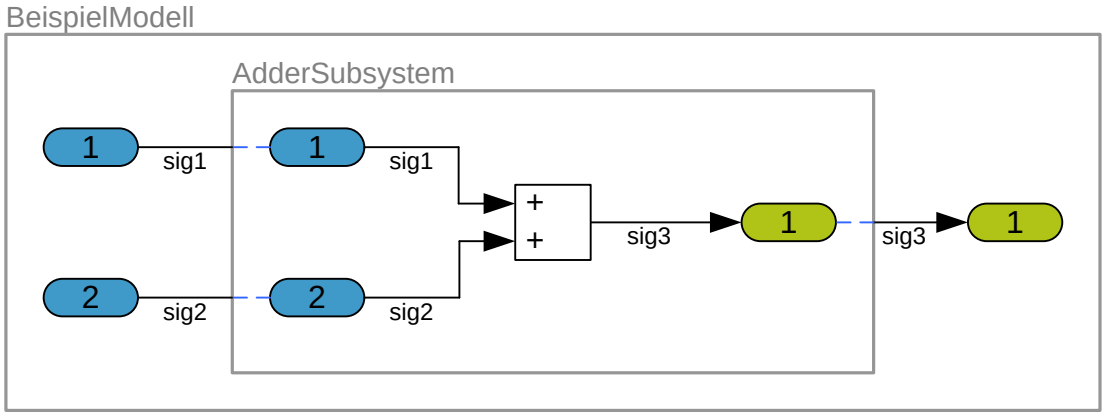


Anzahl der überkreuzenden Linien = 0 (1)

Tiefe der Subsystem-Hierarchie = 3

Durchschnittliche Signalpfadlänge = (6 + 2 + 2) / 3 = 3,33…

Metriken für Simulink-Modelle

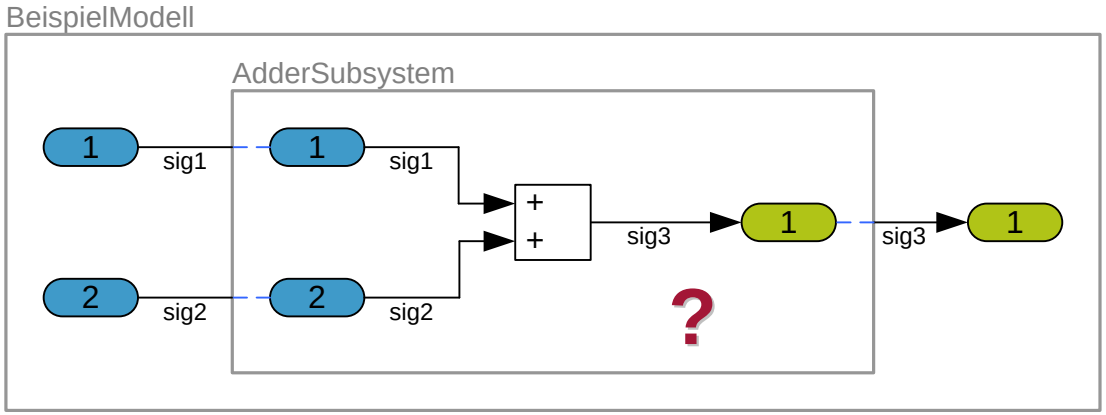


Anzahl der überkreuzenden Linien = 0 (1)

Tiefe der Subsystem-Hierarchie = 1 (3)

Durchschnittliche Signalpfadlänge = (2 + 2 + 2) / 3 = 2 (3,33...)

Metriken für Simulink-Modelle



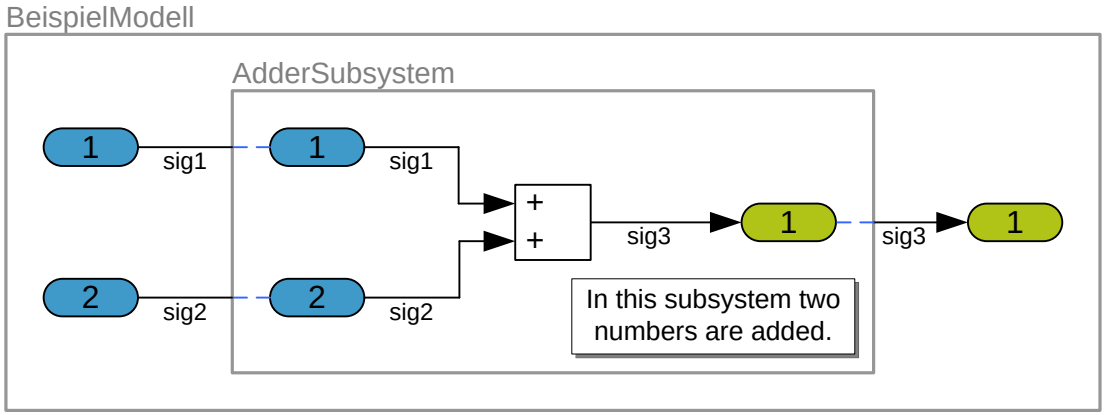
Anzahl der überkreuzenden Linien = 0 (1)

Tiefe der Subsystem-Hierarchie = 1 (3)

Durchschnittliche Signalpfadlänge = 2 (3,33...)

Anzahl der Subsystem-Annotationen = 0

Metriken für Simulink-Modelle



- Anzahl der überkreuzenden Linien = **0** (1)
- Tiefe der Subsystem-Hierarchie = **1** (3)
- Durchschnittliche Signalpfadlänge = **2** (3,33...)
- Anzahl der Subsystem-Annotationen = **1** (0)

Modellmetriken

- bisher existieren in der Literatur nur wenige für Simulink spezifische Metriken
- Metriken für die UML sind meist für Klassendiagramme

Gruppierung der in diesem Ansatz verwendeten Metriken

- Blockmetriken (23)
- Linienmetriken (3)
  - **Anzahl der überkreuzenden Linien**
- Subsystemmetriken (13)
  - **Tiefe der Subsystem-Hierarchie**
- Annotationsmetriken (2)
  - **Anzahl der Subsystem-Annotationen**
- Konstantenmetriken (3)
- Bibliotheksmetriken (5)
- Signalflussmetriken (11)
  - **Durchschnittliche Signalpfadlänge**
- Strukturmetriken (7)
- Stateflow-Metriken (3)
- Codegeneratormetriken (9)
- Artefaktmetriken (5)

Artefakt- und Reportmetriken

- berücksichtigen den gesamten modellbasierten Entwicklungsprozess
- betrachten sowohl weitere Artefakte als auch Ergebnisreports von Tools

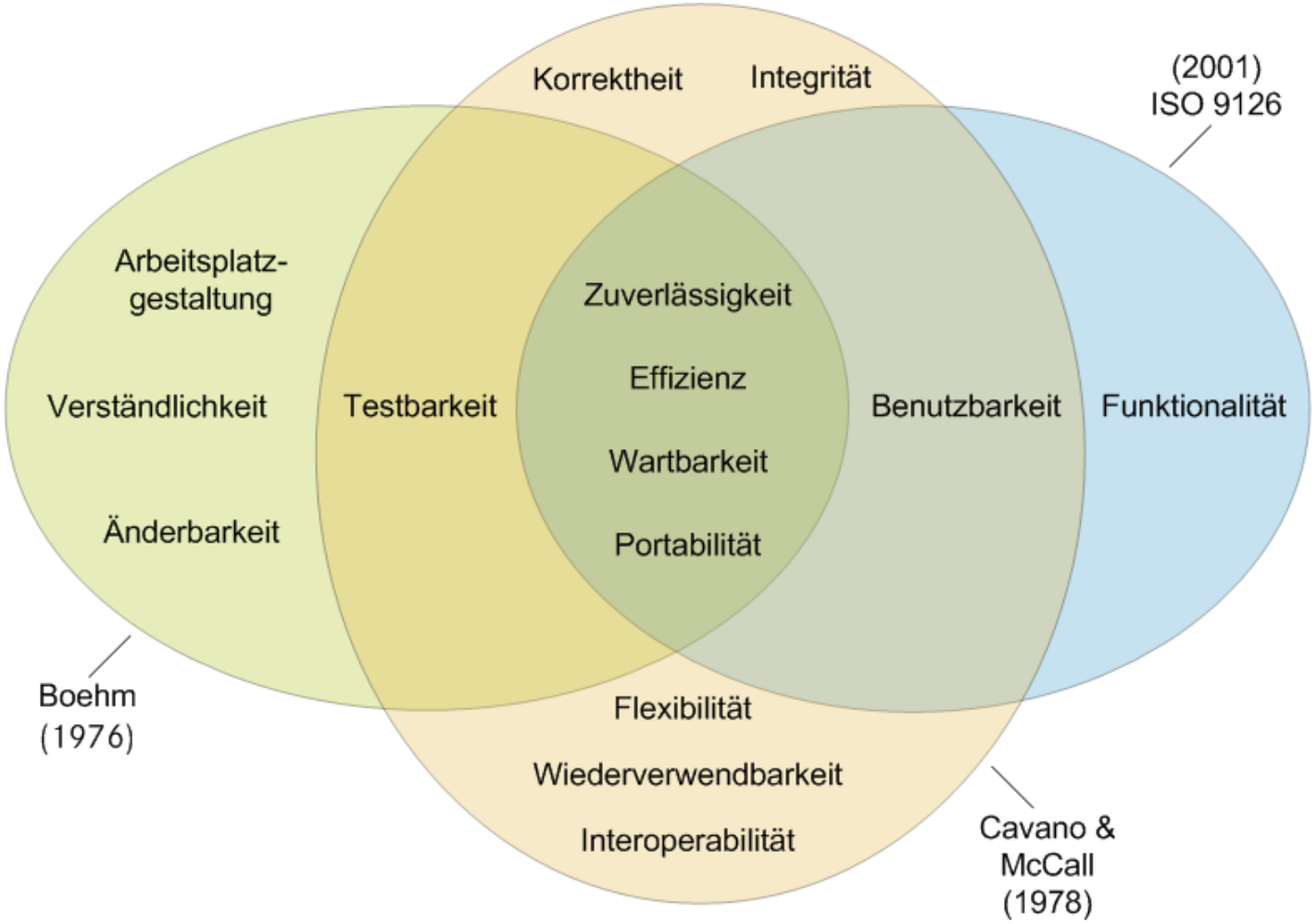
- Blockmetriken (23)
- Linienmetriken (3)
- Subsystemmetriken (13)
- Annotationsmetriken (2)
- Konstantenmetriken (3)
- Bibliotheksmetriken (5)
- Signalflussmetriken (11)
  - **Anzahl der internen Zustände**

- Strukturmetriken (7)
  - **Anzahl der verletzten MISRA TL-Modellierungsrichtlinien**
- Stateflow-Metriken (3)
  - **Anzahl der unerreichbaren Zustände**
- Codegeneratormetriken (9)
  - **Worst-Case-Execution-Time**
- Artefaktmetriken (5)
  - **Prozentsatz erfolgreicher Testfälle**
  - **Testabdeckung**
  - **Anforderungsabdeckung**

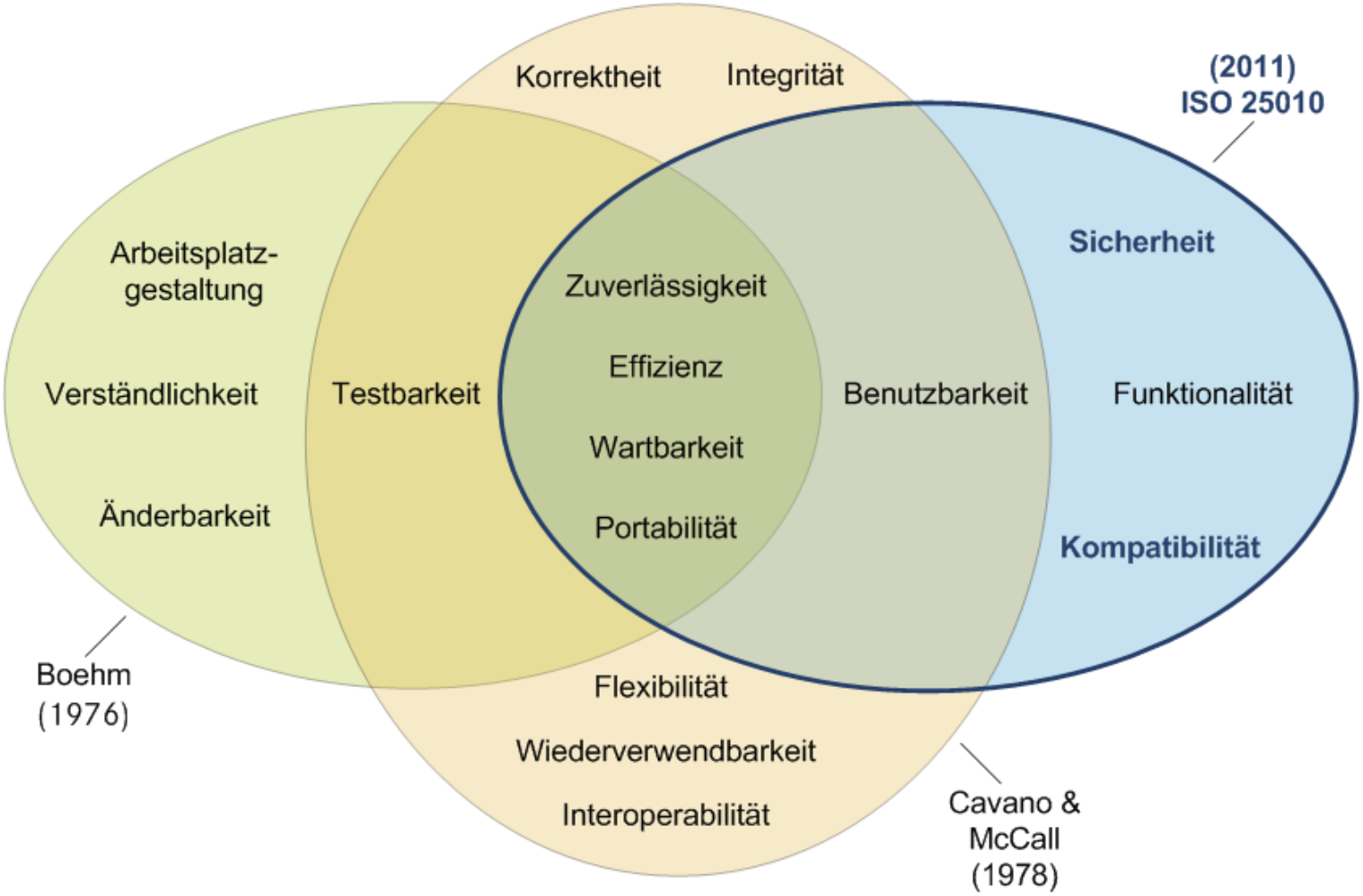
Qualitätsmodell für Simulink-Modelle

- ein Qualitätsmodell
    - gliedert den Begriff der Modellqualität hierarchisch
    - macht die Modellqualität greifbar und überprüfbar
    - enthält auf oberster Ebene abstrakte Faktoren wie z.B. Wartbarkeit, Verständlichkeit oder Testbarkeit
- Modelle mit einer hohen Qualität erfüllen diese abstrakten Faktoren
- für die herkömmliche, handcodierte Softwareentwicklung existieren bereits verschiedene Qualitätsmodelle

Vergleich mit anderen etablierten Qualitätsmodellen

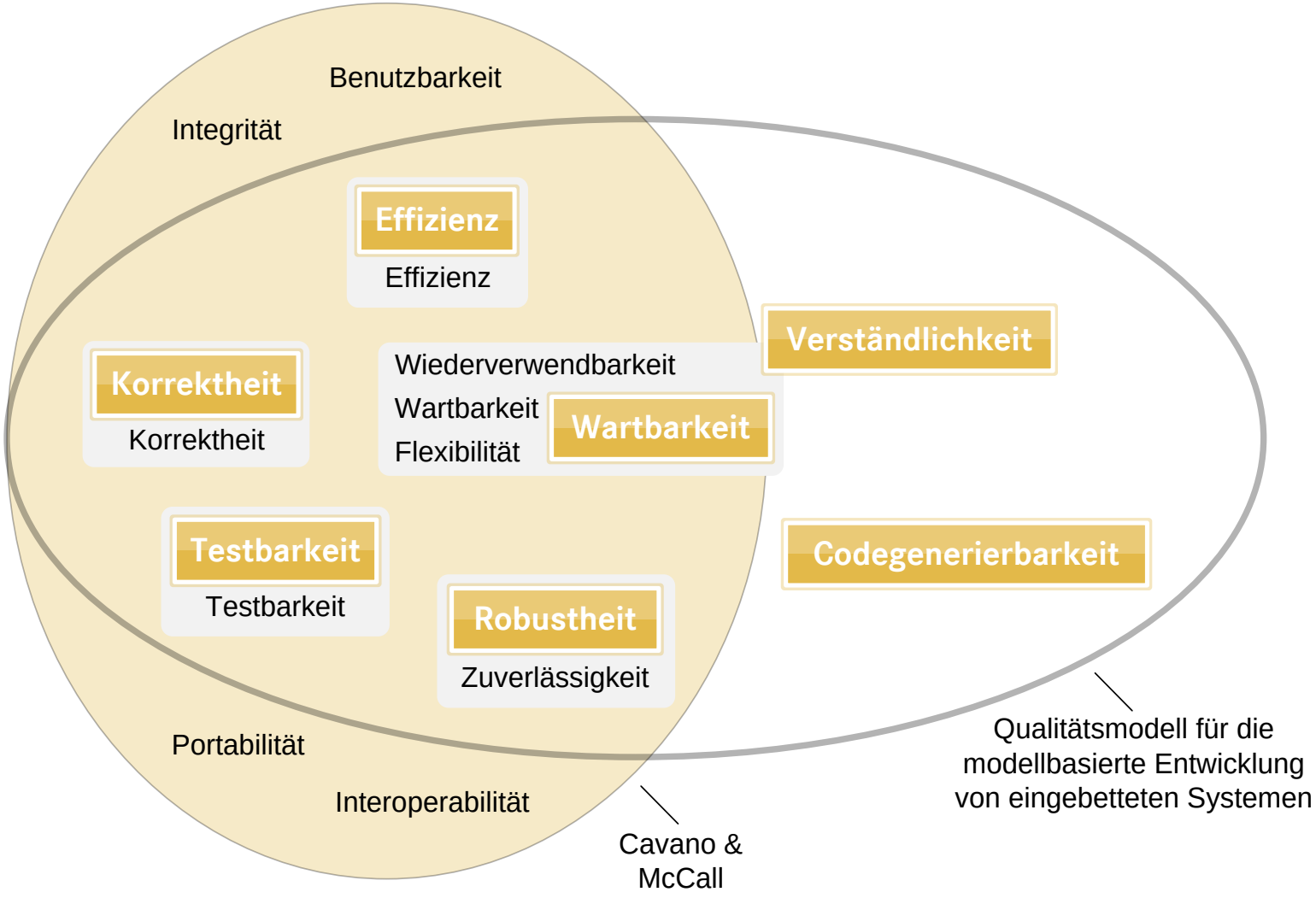


Vergleich mit anderen etablierten Qualitätsmodellen



Vergleich mit anderen etablierten Qualitätsmodellen

Vergleich mit anderen etablierten Qualitätsmodellen



Einordnung der Metriken in das Qualitätsmodell

Faktoren
sind gewünschte Eigenschaften für qualitativ hochwertige Modelle

Verständlichkeit

Wartbarkeit

Metriken

#Subsystem-annotationen

Tiefe der Subsys-temhierarchie

#Überkreuzender Linien

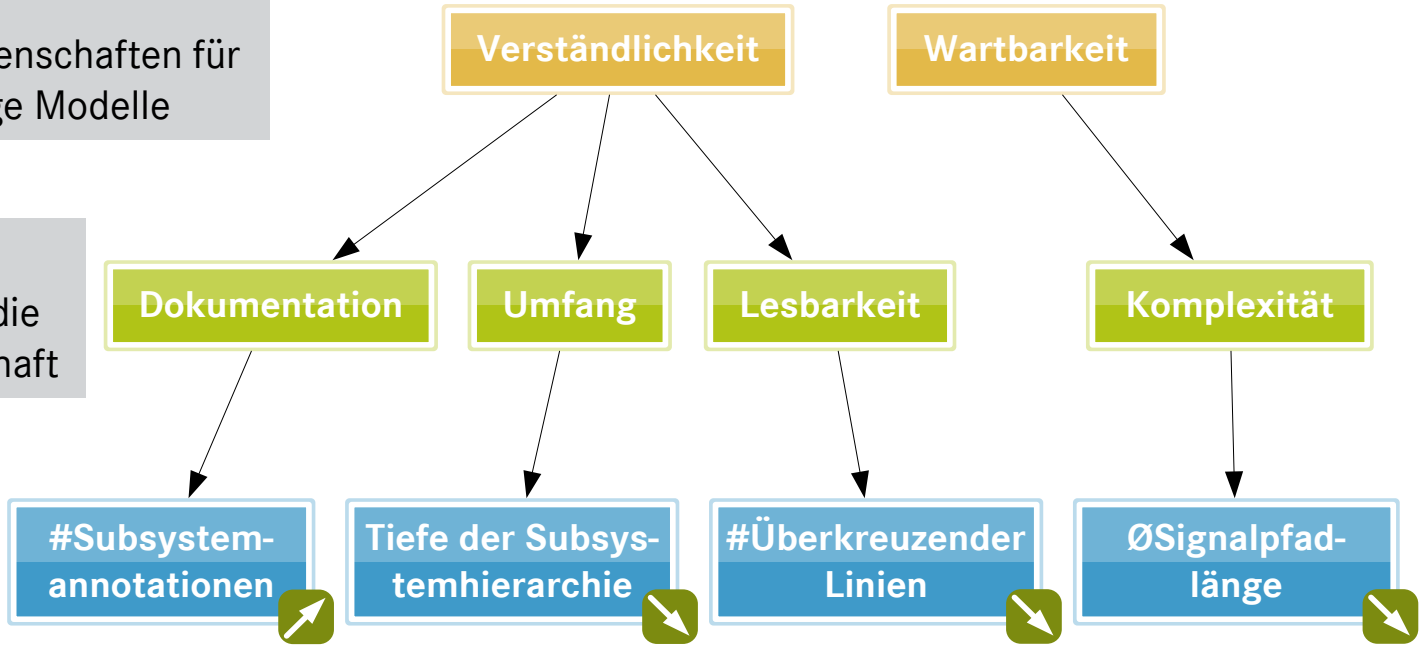
ØSignalpfad-länge

Einordnung der Metriken in das Qualitätsmodell

Faktoren
sind gewünschte Eigenschaften für qualitativ hochwertige Modelle

Kriterien
sind Indikatoren für die gewünschte Eigenschaft

Metriken
prüfen die Erfüllung der Kriterien

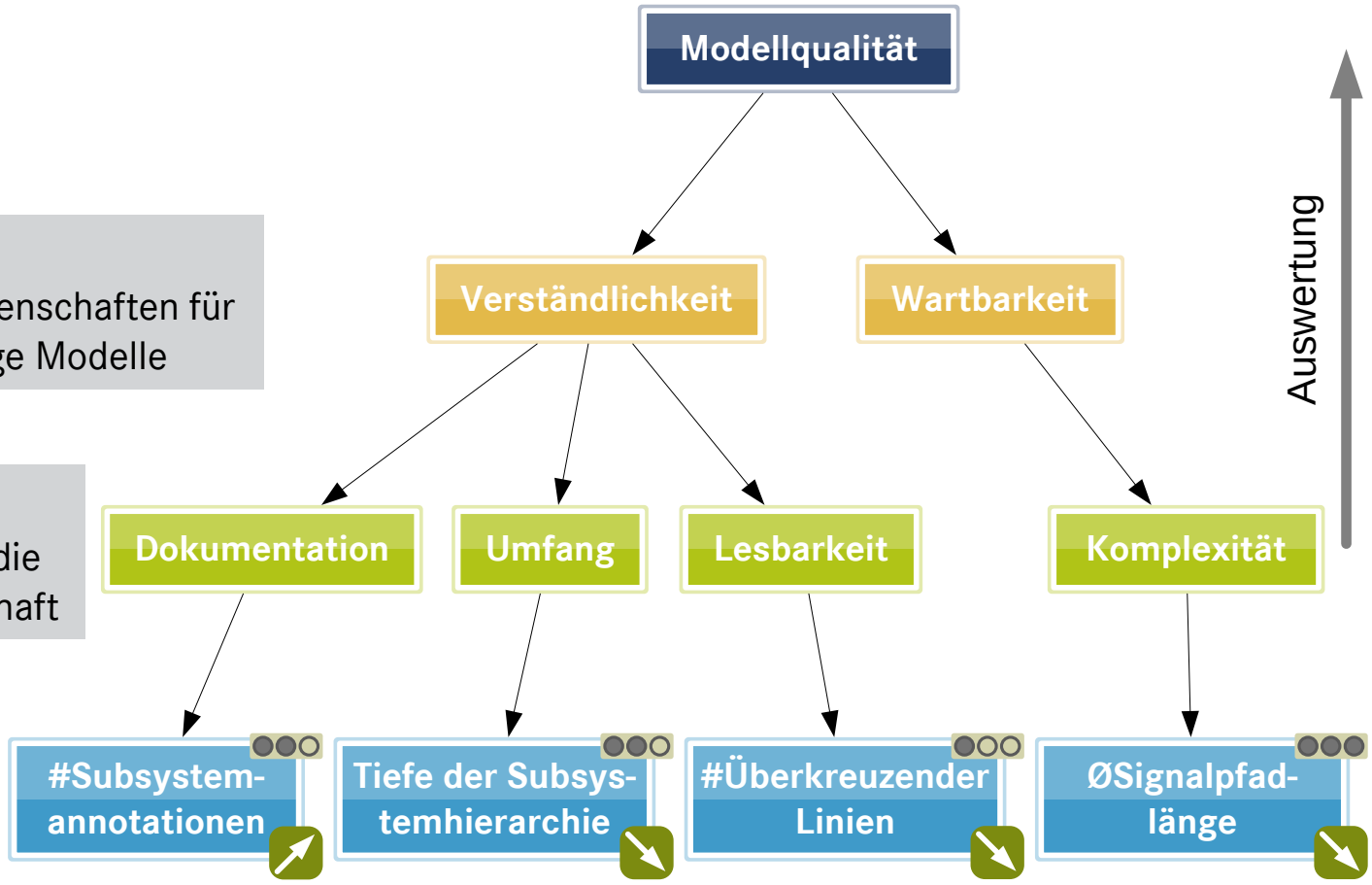


Einordnung der Metriken in das Qualitätsmodell

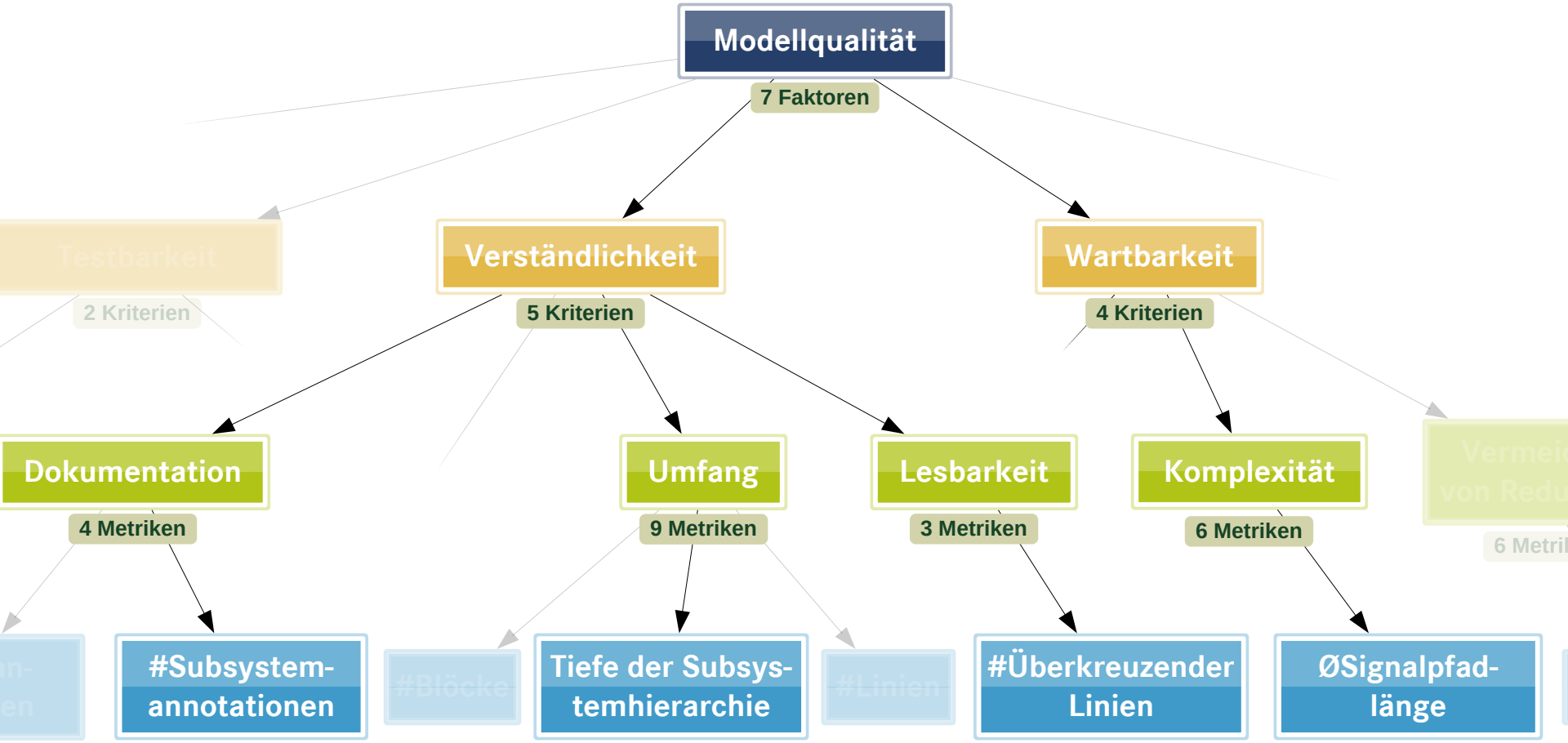
Faktoren
sind gewünschte Eigenschaften für qualitativ hochwertige Modelle

Kriterien
sind Indikatoren für die gewünschte Eigenschaft

Metriken
prüfen die Erfüllung der Kriterien

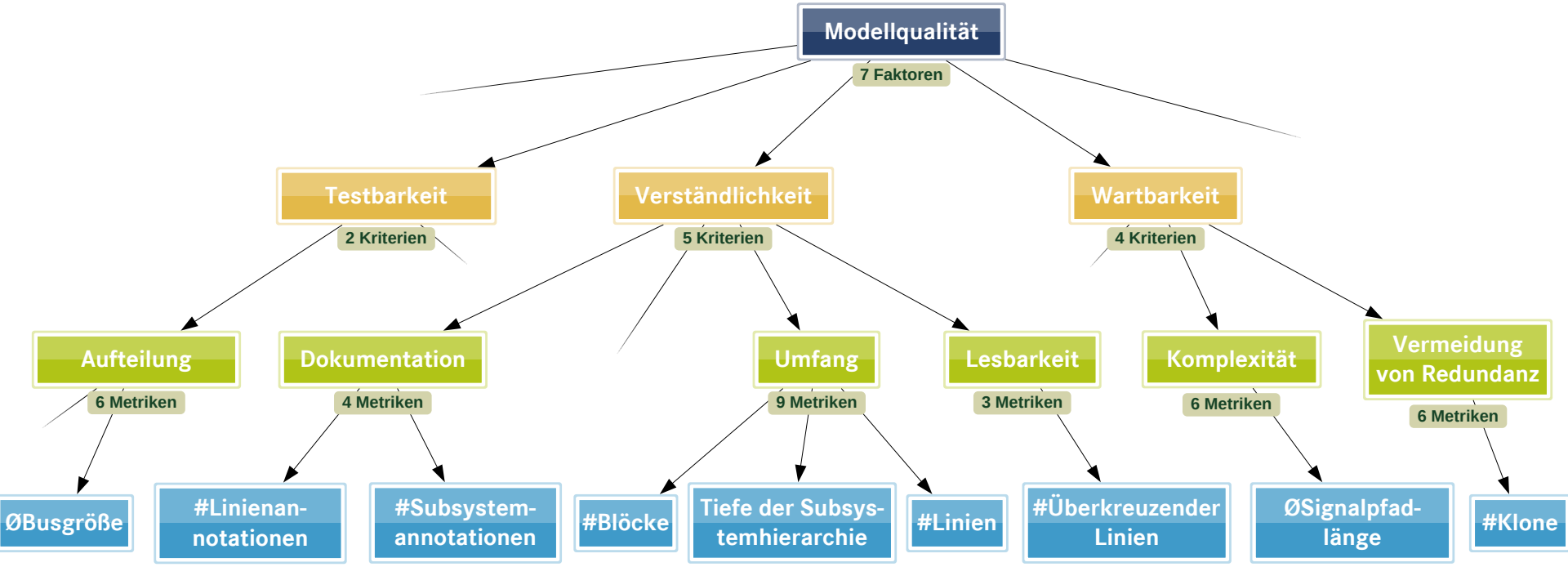


Überblick über das Qualitätsmodell



Überblick über das Qualitätsmodell

- momentan enthält das Qualitätsmodell 7 Faktoren, 24 Kriterien und 84 Metriken



[2] J. Scheible und I. Kreuz. „Ein Qualitätsmodell zur automatisierten Ermittlung der Modellqualität bei eingebetteten Systemen“. *In: Proc. of the 8. GI Workshop Automotive Software Engineering, Leipzig, Deutschland.* 2010.

Modellqualitätsbewertung

- macht das Qualitätsmodell ausführbar
- bildet die absoluten Messwerte der Metriken auf relative Bewertungen ab
  - Regeln geben gewünschte Zusammenhänge vor, wie z.B. die maximal erlaubte Anzahl an überkreuzenden Linien pro Subsystem
  - gibt es für eine Metrik keine Regel, wird ihr Messwert mit Messwerten anderer Modelle verglichen
- aggregiert die Bewertungen im Qualitätsmodell von unten nach oben
  → dadurch kommen die Bewertungen der Faktoren zustande

[3] J. Scheible und H. Pohlheim. „Automated Model Quality Rating of Embedded Systems“. *In: 4. Workshop zur Software-Qualitätsmodellierung und -bewertung (SQMB ’11).* 2011.

Handlungsempfehlungen

- Handlungsempfehlungen geben auf Basis der Modellqualitätsbewertung Hinweise, wie die Qualität eines Simulink-Modells verbessert werden kann
- dazu werden die Informationen aus dem Qualitätsmodell und die konkreten Messwerte eines Modells verknüpft
- Beispiele:
  - »Tiefe der Subsystem-Hierarchie« 
    - Qualitätsmodell: kleinere Messwerte sind besser als größere
    - Empfehlung: Sobald die Subsystem-Hierarchie zu tief ist, muss ein Modell-Refactoring durchgeführt werden
  - »Anzahl der Subsystem-Annotationen« 
    - Qualitätsmodell: größere Messwerte sind besser als kleinere
    - Empfehlung: Die Subsysteme müssen besser dokumentiert werden

Inhalt

Problemstellung

Lösungsansatz

Prototyp

Evaluation

Zusammenfassung und Ausblick

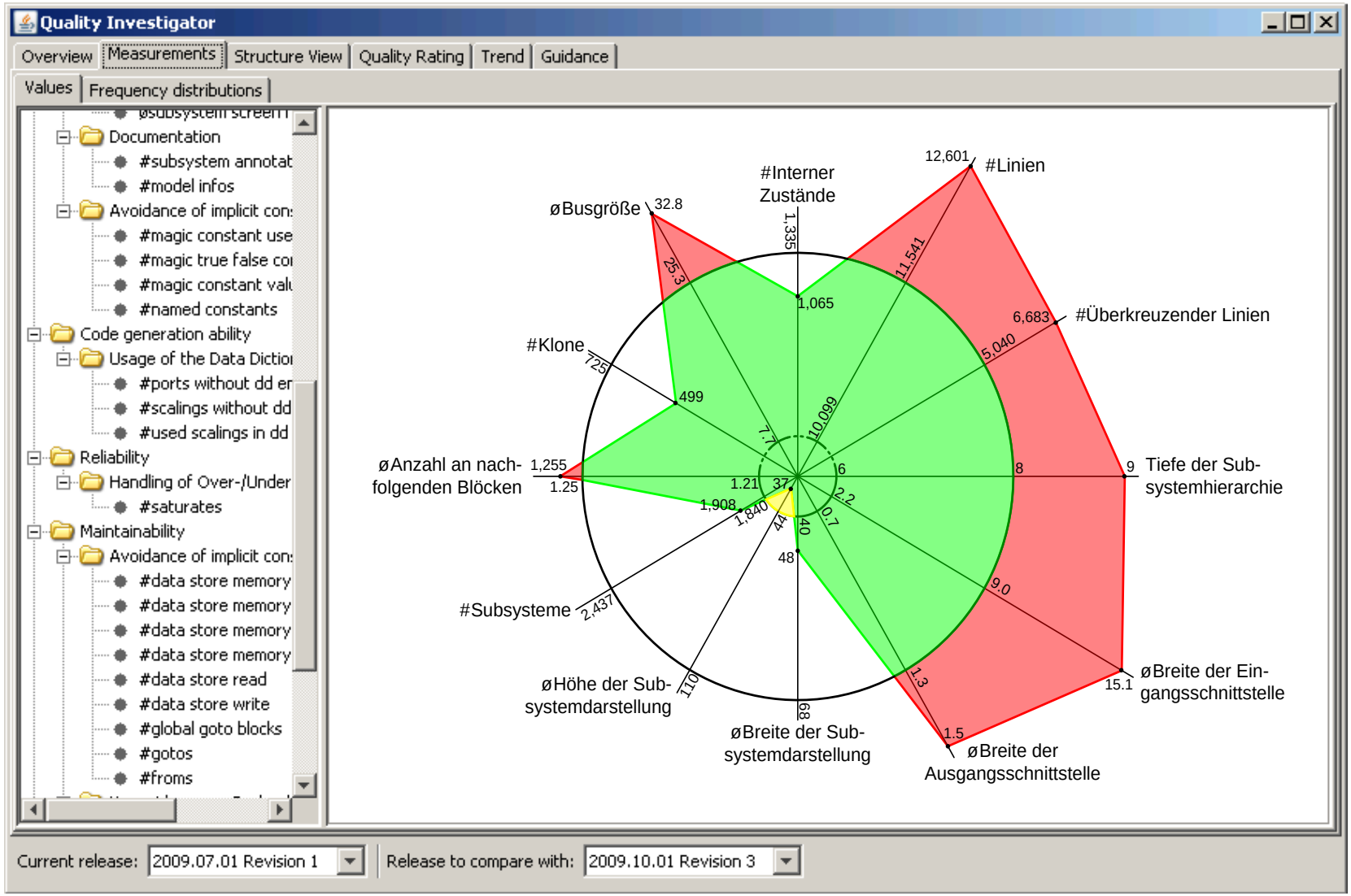
Prototyp

- enthält einen Modell-Exporter für Simulink (ca. 2.300 Zeilen MATLAB-Script)
- Implementiert die Modellmetriken und das Qualitätsmodell (ca. 20.000 Zeilen Java-Code in über 400 Klassen)
- verwendet ein Metamodell für den Zugriff auf die Simulink-Modelle in Java
- visualisiert die Ergebnisse der Modellqualitätsbewertung

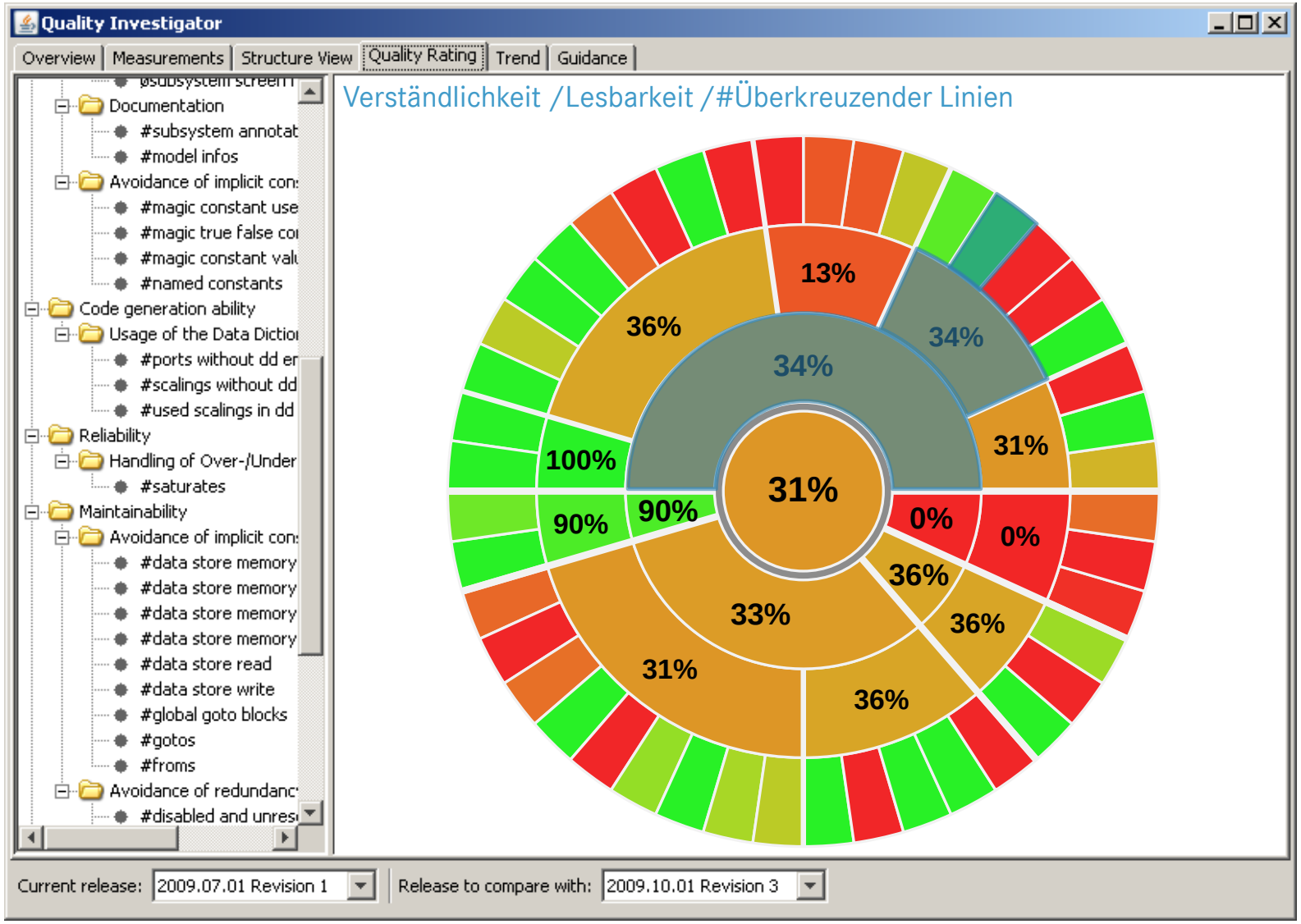
→ ermöglicht eine automatisierte Bewertung der Modellqualität

```
11:45:06.265 INFO  ModelMeasurer - Processing the file 'model.mdlx'.
11:45:06.265 DEBUG
* Created the measurement object - Starting the measurement...
    #crossed lines (1/84)... done!
    øsignal length (2/84)... done!
    #blocks (3/84)... done!
    #lines (4/84)... done!
              :
              :
    #subsystem annotations (83/84)... done!
    depth of subsystem hierarchy (84/84)... done!
* finished!
11:50:11.437 INFO Saved the results to 'model.measurements' (0 hours, 5 min, 11 sec).
```

Darstellung der Messwerte und ihrer Grenzen



Darstellung der aggregierten Bewertungen



Inhalt

Problemstellung

Lösungsansatz

Prototyp

Evaluation

Zusammenfassung und Ausblick

Evaluation (1/3)

- Verwendung des implementierten Prototyps
  - 74 Metriken wurden verwendet (10 Metriken mit Abhängigkeiten von nicht verfügbaren externen Tools oder Daten mussten weggelassen werden)
- es wurden 8 reale Modelle aus der PKW-Entwicklung untersucht
- Überprüfung folgender Eigenschaften:
  - **Validität** – Deckt sich die Modellqualitätsbewertung mit der Einschätzung von Experten?
  - **Diversifikation** – Können mithilfe der Modellqualitätsbewertung trotz Abstraktion verschiedene Modelle unterschieden werden?

Evaluation (2/3)
Validität

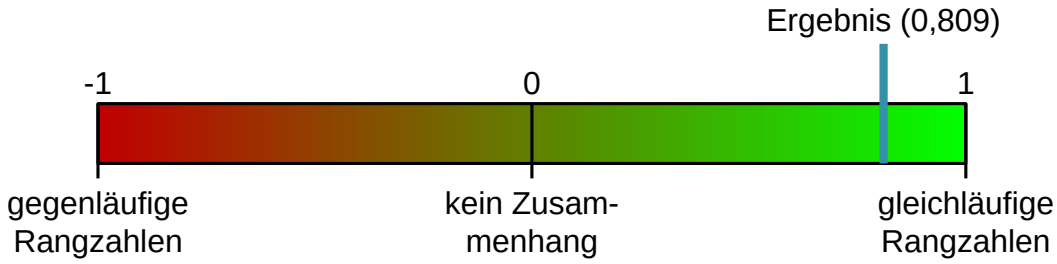
Modell	$rang(x_{mbq})$	$rang(y_{exp})$	d_i	d_i^2
Modell 2	3	3	0	0
Modell 3	2	1	1	1
Modell 1	4	2	2	4
Modell 5	1	4	-3	9
Modell 4	6	5	1	1
Modell 6	5	6	-1	1
Modell 8	7	7	0	0
Modell 7	8	8	0	0

H_A: Es besteht eine positive Korrelation

$r_s = 0,809$

→ Annahme von H_A mit einer Konfidenz von 99%

Interpretation von Spearmans Rangkorrelationskoeffizient:



Evaluation (3/3)

- die Evaluation des Ansatzes war mit den betrachteten Modellen erfolgreich
  - Vergleich der Modellqualitätsbewertung mit den Einschätzungen der Experten hat eine hohe Übereinstimmung gezeigt
  - Diversifikation der Modellqualitätsbewertung war gegeben
- allerdings wurde bei der Evaluation ein konkretes Qualitätsmodell evaluiert
  - sobald Änderungen an dem Qualitätsmodell vorgenommen werden, müssen die gewünschten Eigenschaften erneut geprüft werden
- weitere empirische Studien sind notwendig, die beispielsweise die tatsächlich im Feld aufgetretene Anzahl der Fehler betrachten

Inhalt

Problemstellung

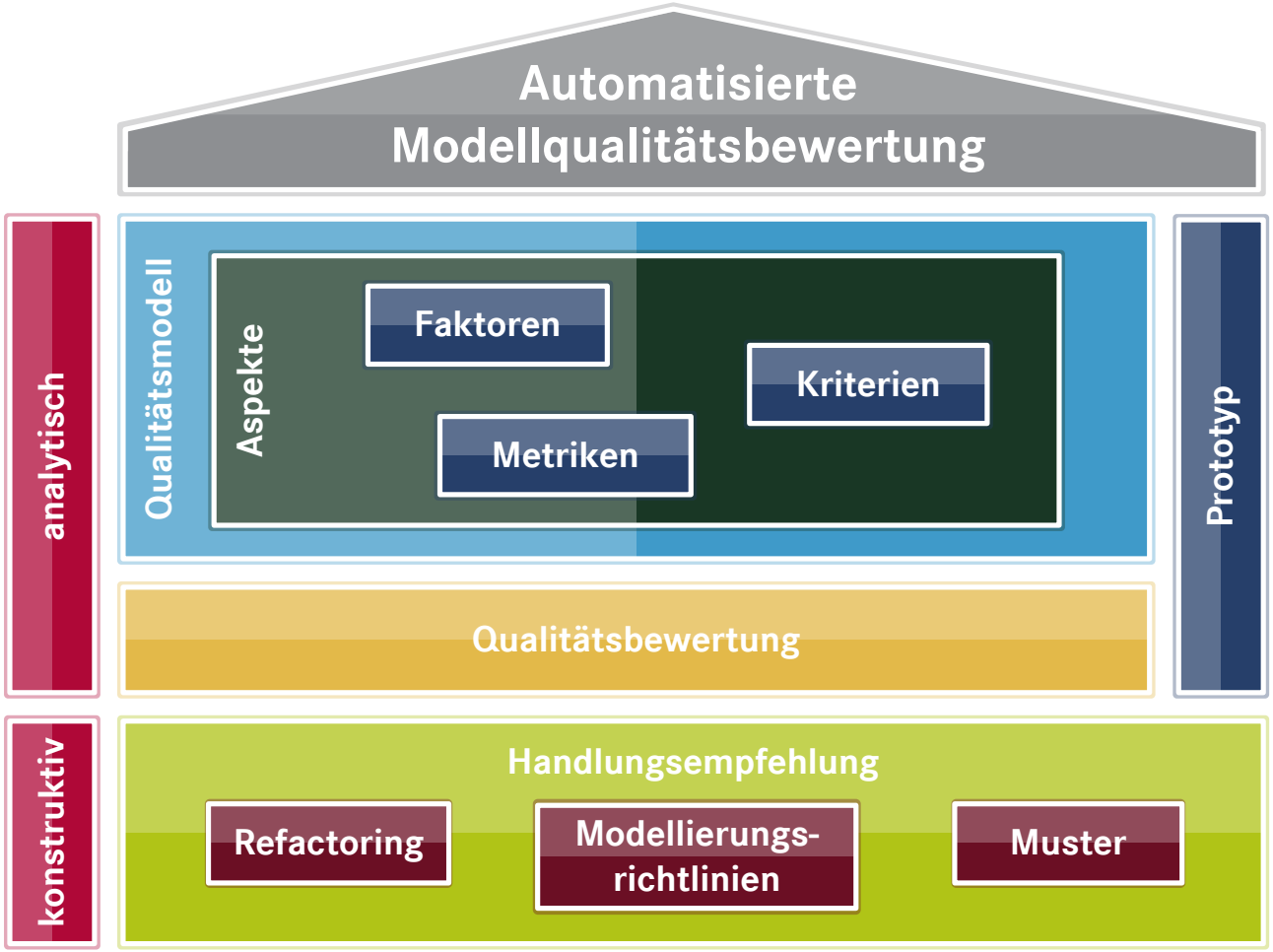
Lösungsansatz

Prototyp

Evaluation

Zusammenfassung und Ausblick

Zusammenfassung



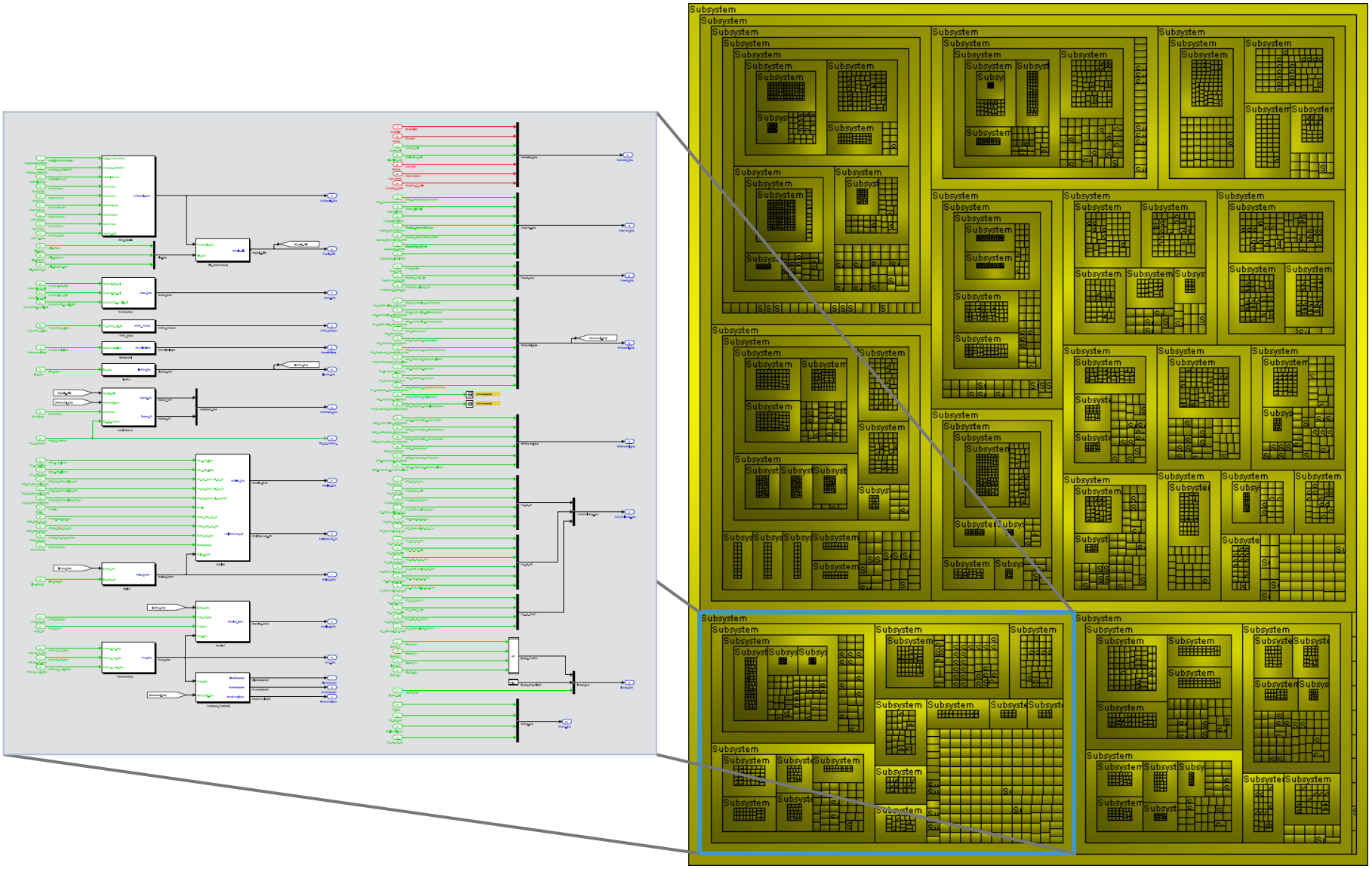
[1] J. Scheible. „Ein Framework zur automatisierten Ermittlung der Modellqualität bei eingebetteten Systemen“. In: *Proc. of the Dagstuhl-Workshop: Model-Based Development of Embedded Systems (MBEES), Schloss Dagstuhl, Deutschland*. 2010.

Ausblick

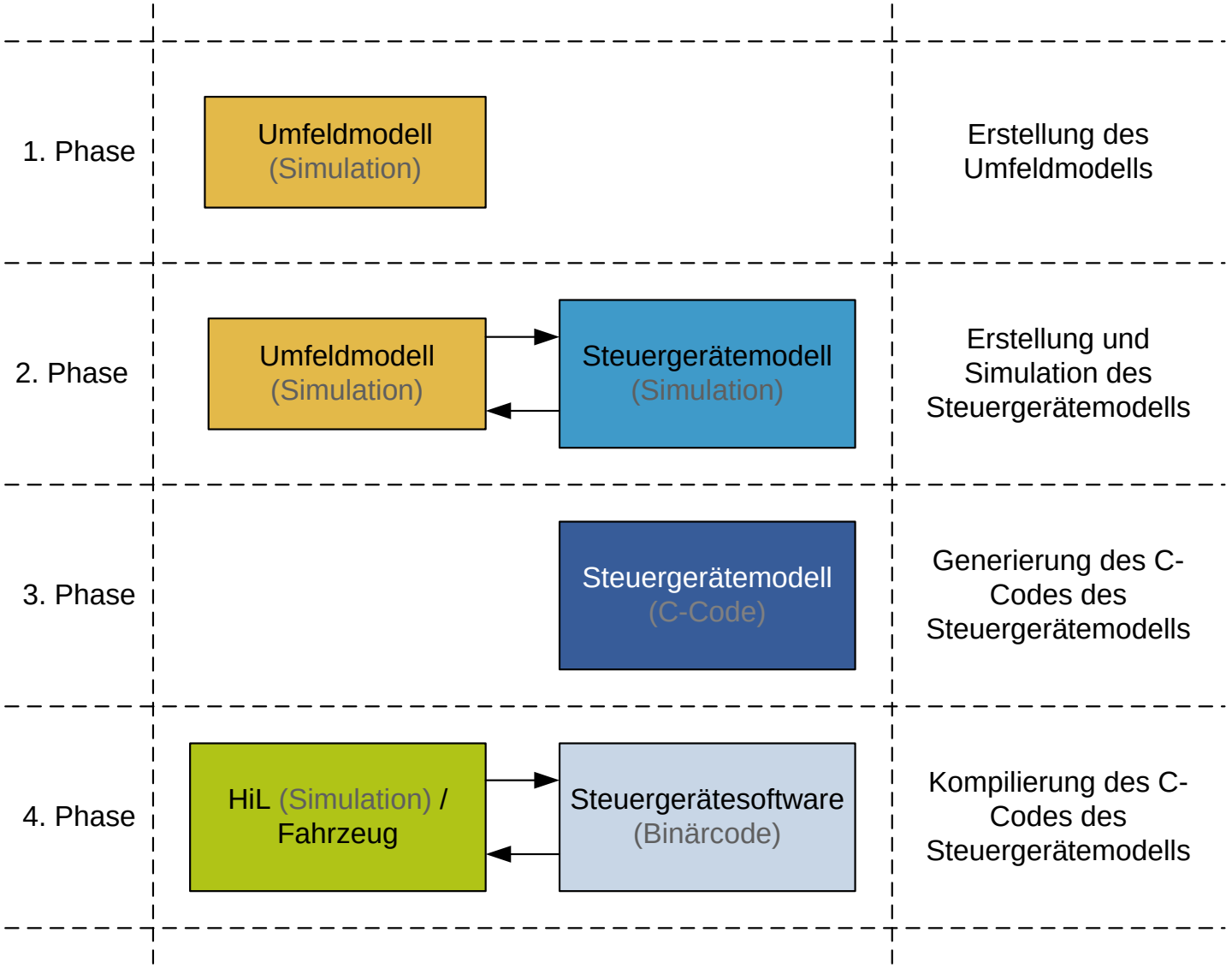
- Transfer der Arbeitsergebnisse in ein kommerzielles Produkt
- Betrachtung der Modellqualität pro Teilsystem bzw. pro Modul
- zusätzliche Metriken für Stateflow-Diagramme
- Verknüpfung der Modellqualitätsbewertung mit konkreten Vorschlägen für ein Modell-Refactoring als Ergänzung zu den Handlungsempfehlungen
- Übertragen der Modellqualitätsbewertung auf andere Modellierungssprachen (wie z.B. ASCET von ETAS)

Vielen Dank!

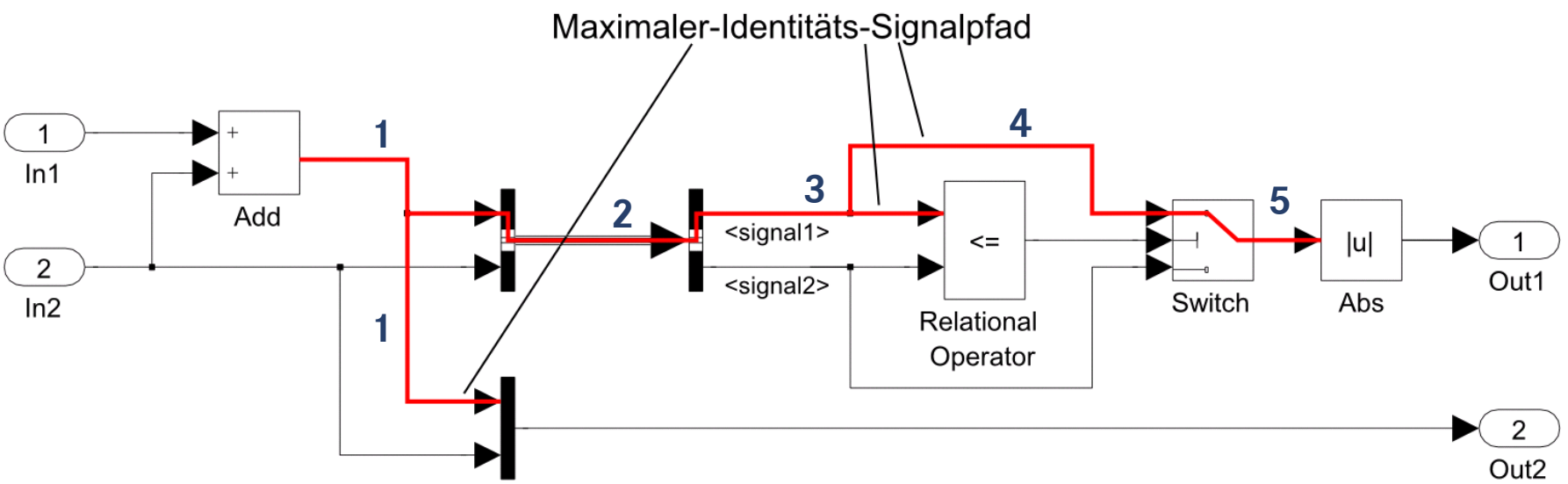
Beispiel eines großen Modells



Phasen des modellbasierten Entwicklungsprozesses

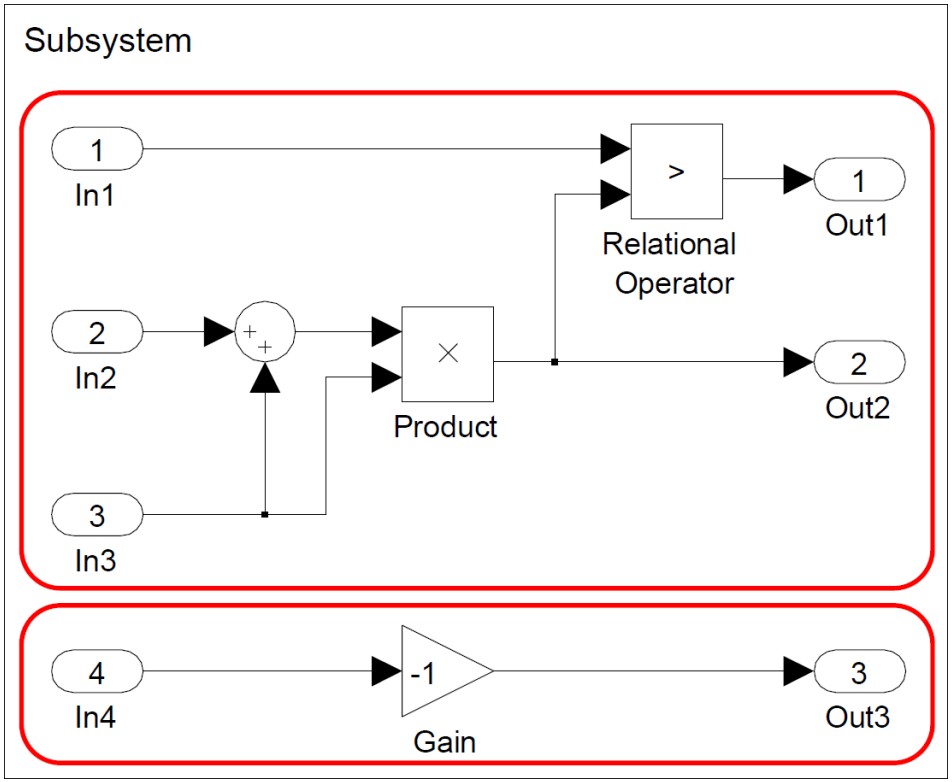


Metriken und Messwerte (1/2)



Durchschnittliche Signalpfadlänge = (1 + 3 + 5) / 3 = 3

Metriken und Messwerte (2/2)



Durchschnittliche Anzahl der unabhängigen Berechnungspfade = 2

Referenzmodell

- definiert, wie ein typisches Simulink-Modell aussieht
- enthält erlaubte Minimal- und Maximalwerte
- ist dank Normierung durch eine Referenzmetrik (z.B. »Anzahl der Blöcke« oder »Höhe der globalen Komplexität«) größenunabhängig

					Erlaubter Bereich		
	Modell 1	Modell 2		Modell n	Min.	Mittelw.	Max.
# Subsysteme	3,4	4,7		6,3	1,6	8,1	14,6
# Linien	134,4	179,3	…	174,5	141,3	167,1	192,9
# überkreuzender Linien	24,6	74,3		174,5	7,3	53,6	101,9

Regeln

- machen Aussagen über gewünschte Zusammenhänge von Messwerten
- werden verwendet, wenn sich keine sinnvollen Mittelwerte bilden lassen

$$0 \leq \frac{\text{Anzahl der UnitDelay-Blöcke}}{\text{Anzahl der Subsysteme}} \leq 2$$

Erlaubter Bereich	
Min.	Max.
0	2,0

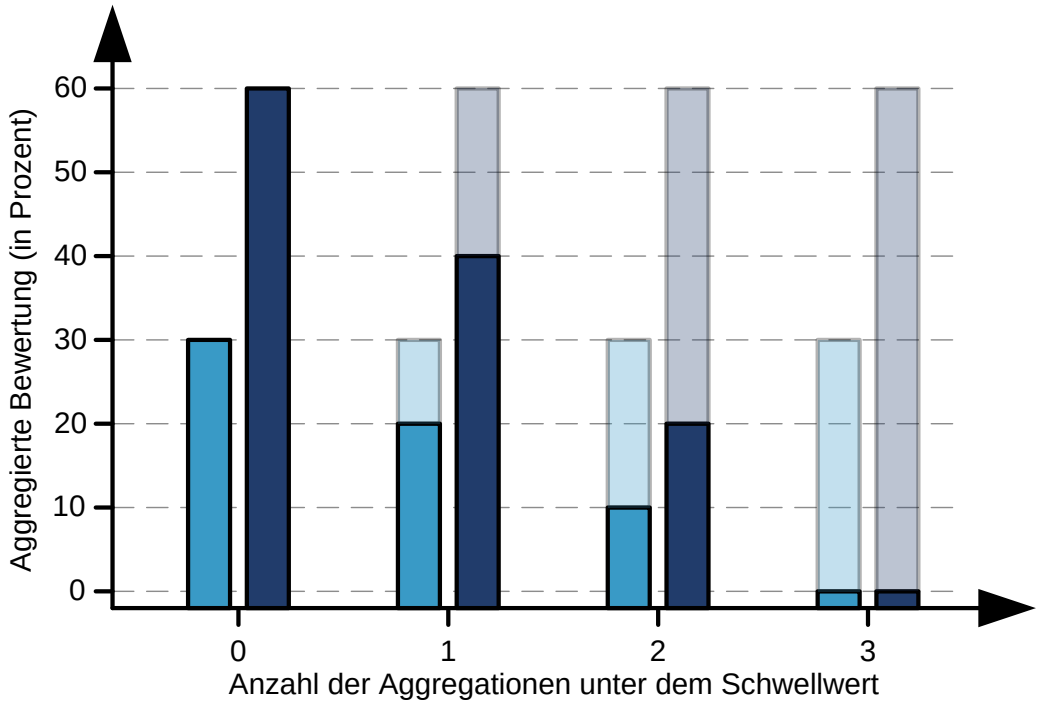
Aggregation der Bewertungen
Auswirkung von Schwellwertüberschreitungen

- aggregierte Werte erlauben einen einfachen Überblick über die Modellqualität
- Bildung des arithmetischen Mittels ist nicht ausreichend: Sind eine oder mehrere Bewertungen nicht erfüllt, muss das zu einer Abwertung führen

Bewertungen:

$e_1 = [25\%, 30\%, 35\%]$

$e_2 = [10\%, 80\%, 90\%]$



Evaluation
Diversifikation