



[disi: arti:]

Automatische Verlinkung von Testfällen und Anforderungen – Testplangesteuerte Filterung

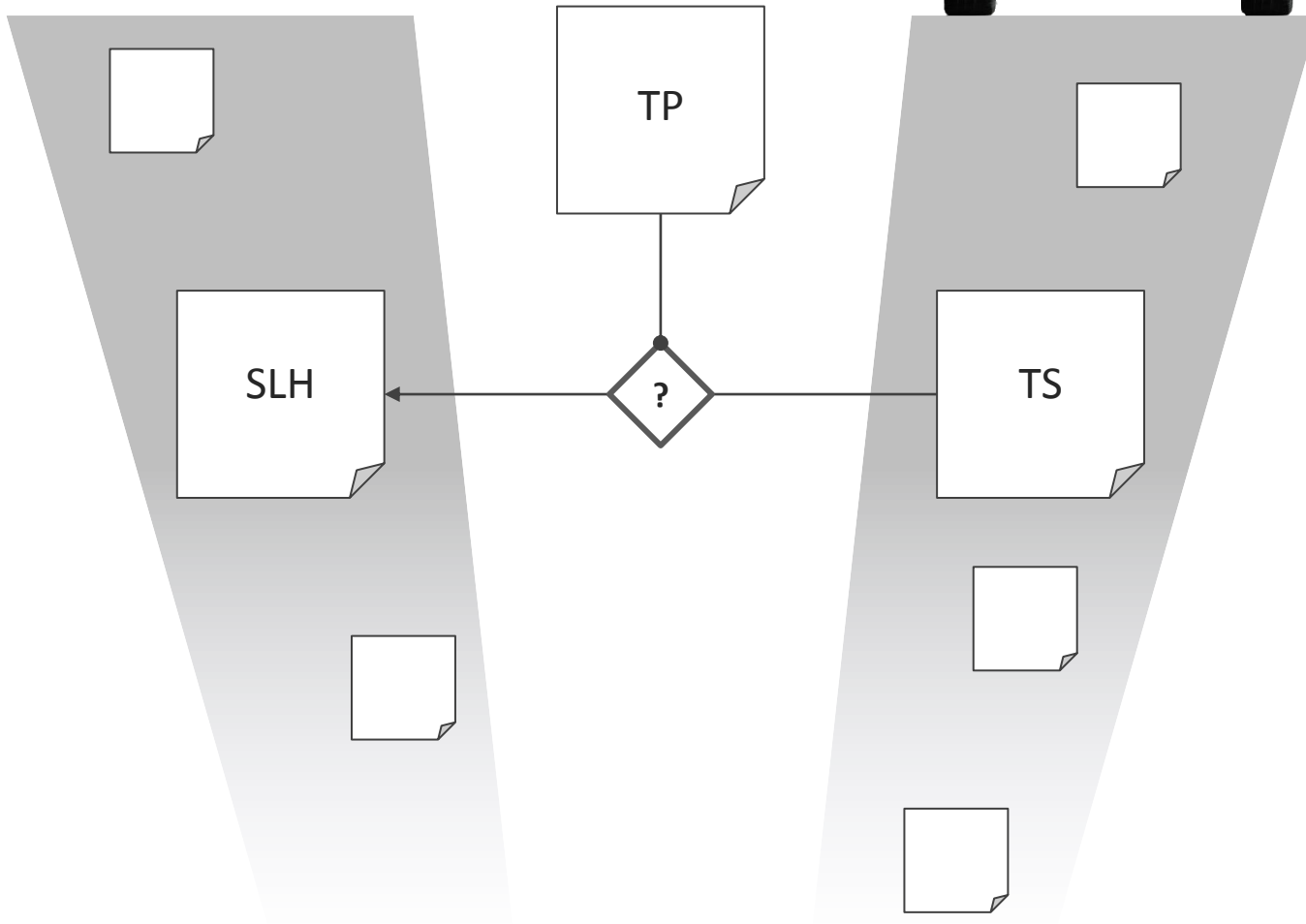
35. Fachgruppentreffen TAV, GI e.V.

Thomas Noack, TU Berlin, Daimler Center for Automotive IT Innovations (DCAITI)

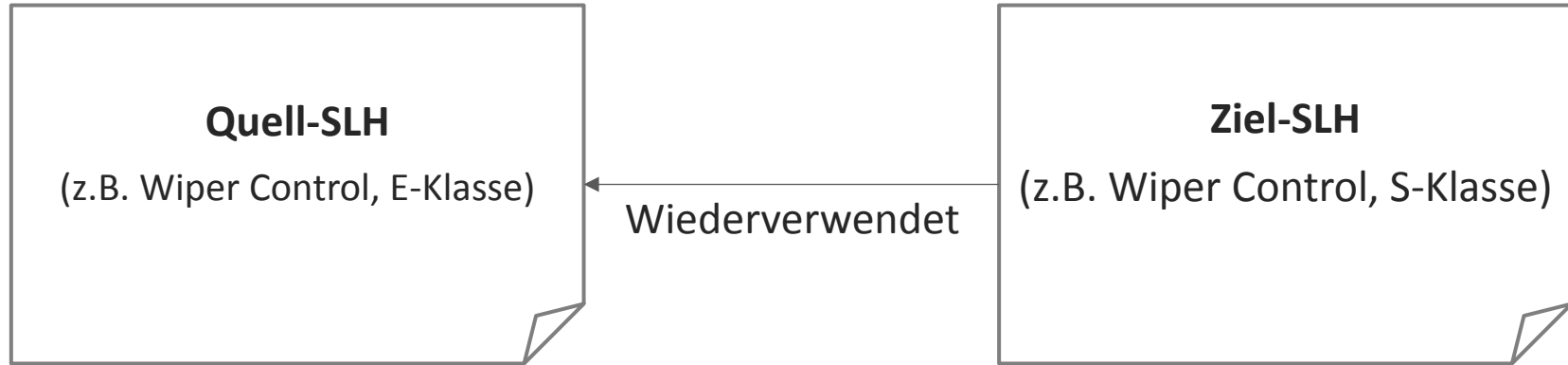
21. November 2013

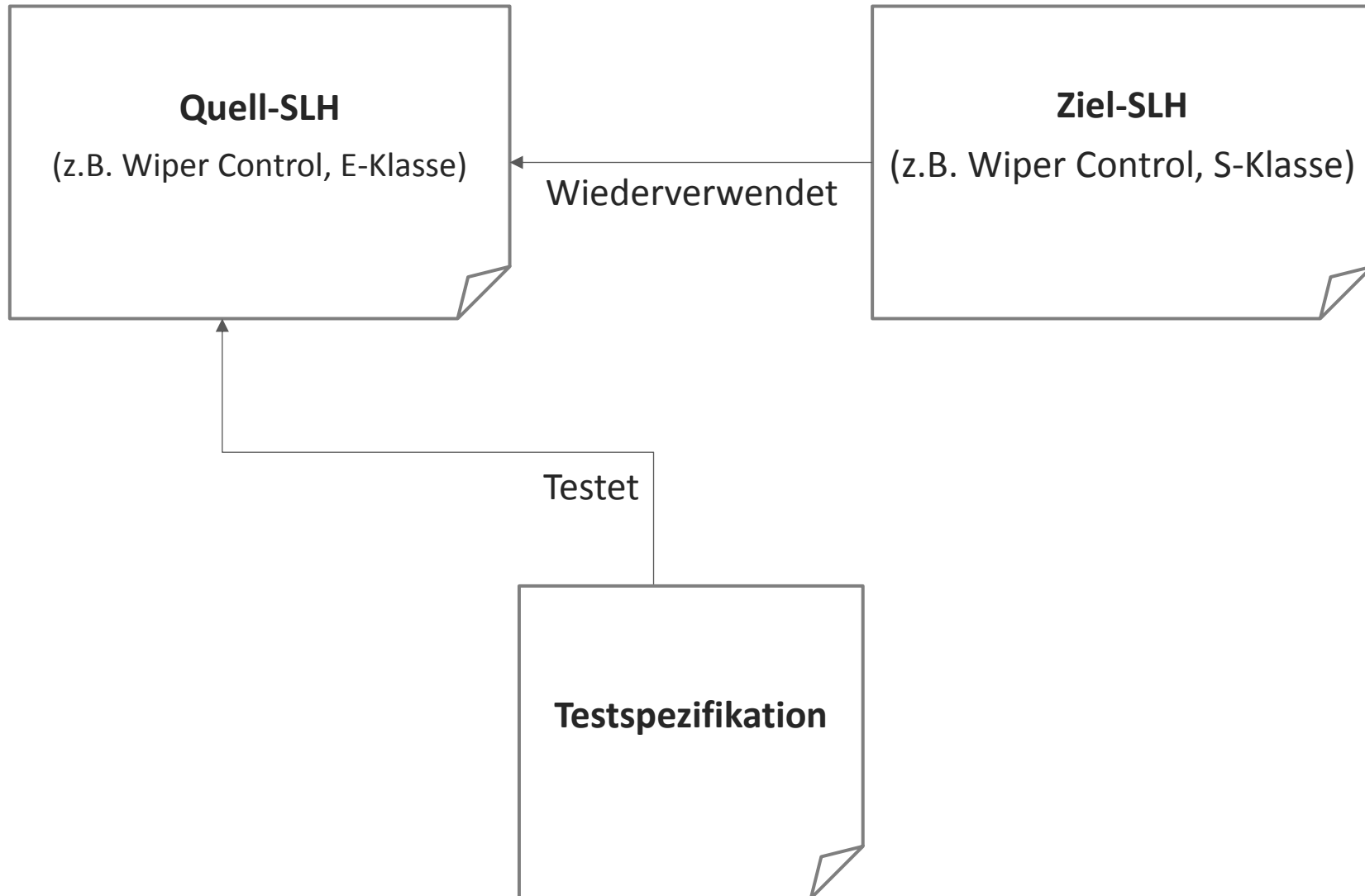
1

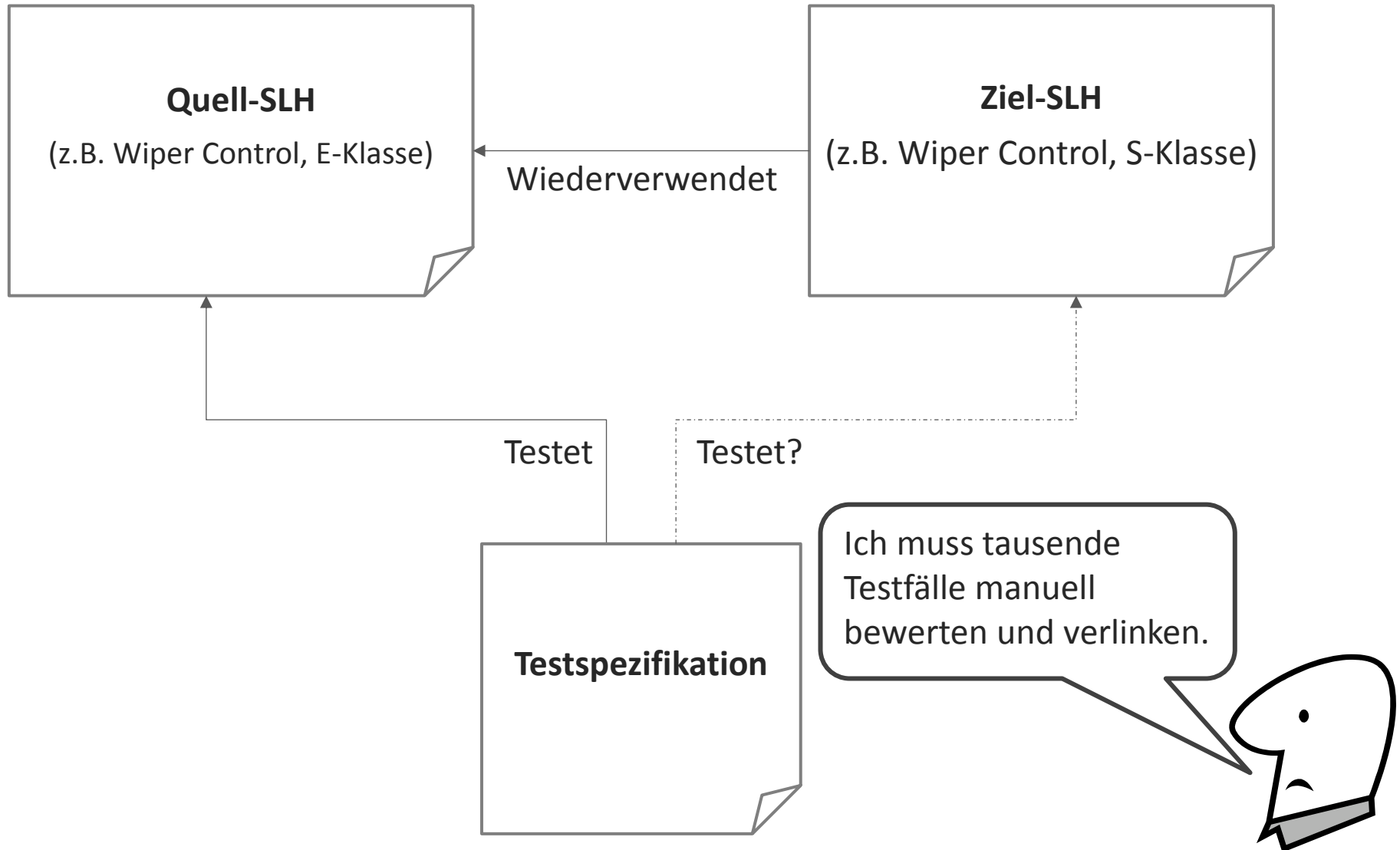
AUSGANGSSITUATION UND PROMOTIONSZIEL



SLH: Systemlastenheft, TS: Testspezifikation, TP: Testplan



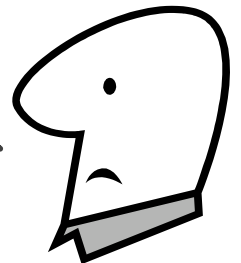




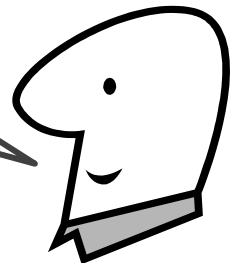
- Welche Anforderungen wurden (un)modifiziert wiederverwendet?
- Welche Anforderungen sind neu?
- Welche Testfälle verlinken auf welche Quell-Anforderungen?
- Welche Testfälle können (nicht) mit Ziel-Anforderungen verlinkt werden?
- Welche Inkonsistenzen entstehen durch die Wiederverwendung?
- Welche Anforderungen sind (nicht) ausreichend abgesichert?
- Welche Testfälle brauche ich für das aktuelle Testprojekt nicht?



Beantwortung und Dokumentation
wird durch ISO 26262 gefordert!

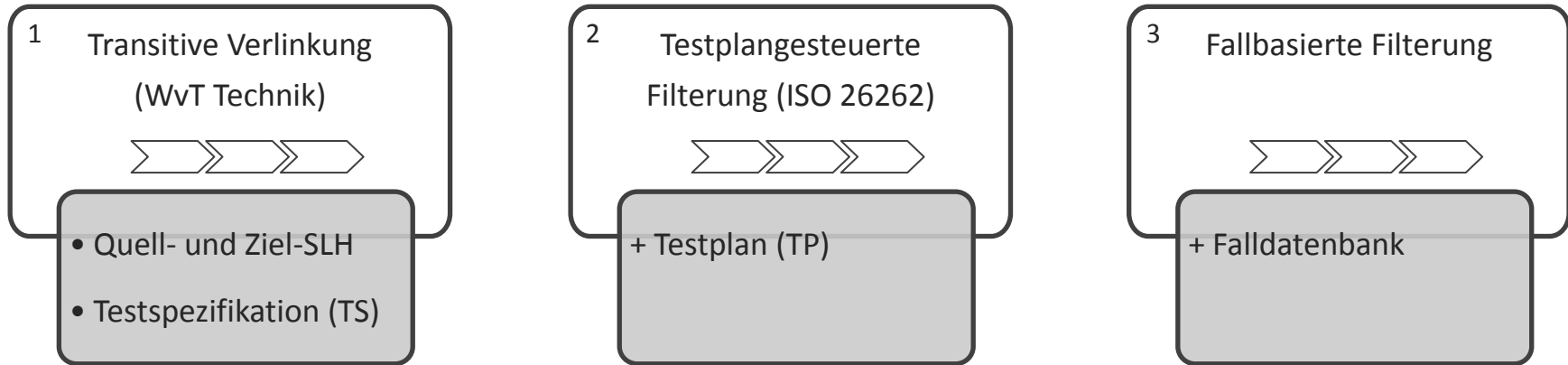


Ich ziehe Trace Links zwischen Testfällen und wiederverwendeten SLH-Anforderungen jetzt automatisch. Damit spare ich bei jedem Baureihenprojekt Zeit!

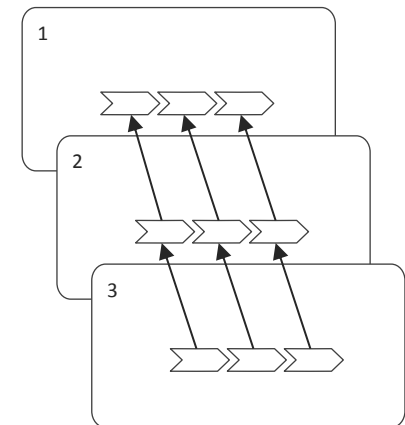


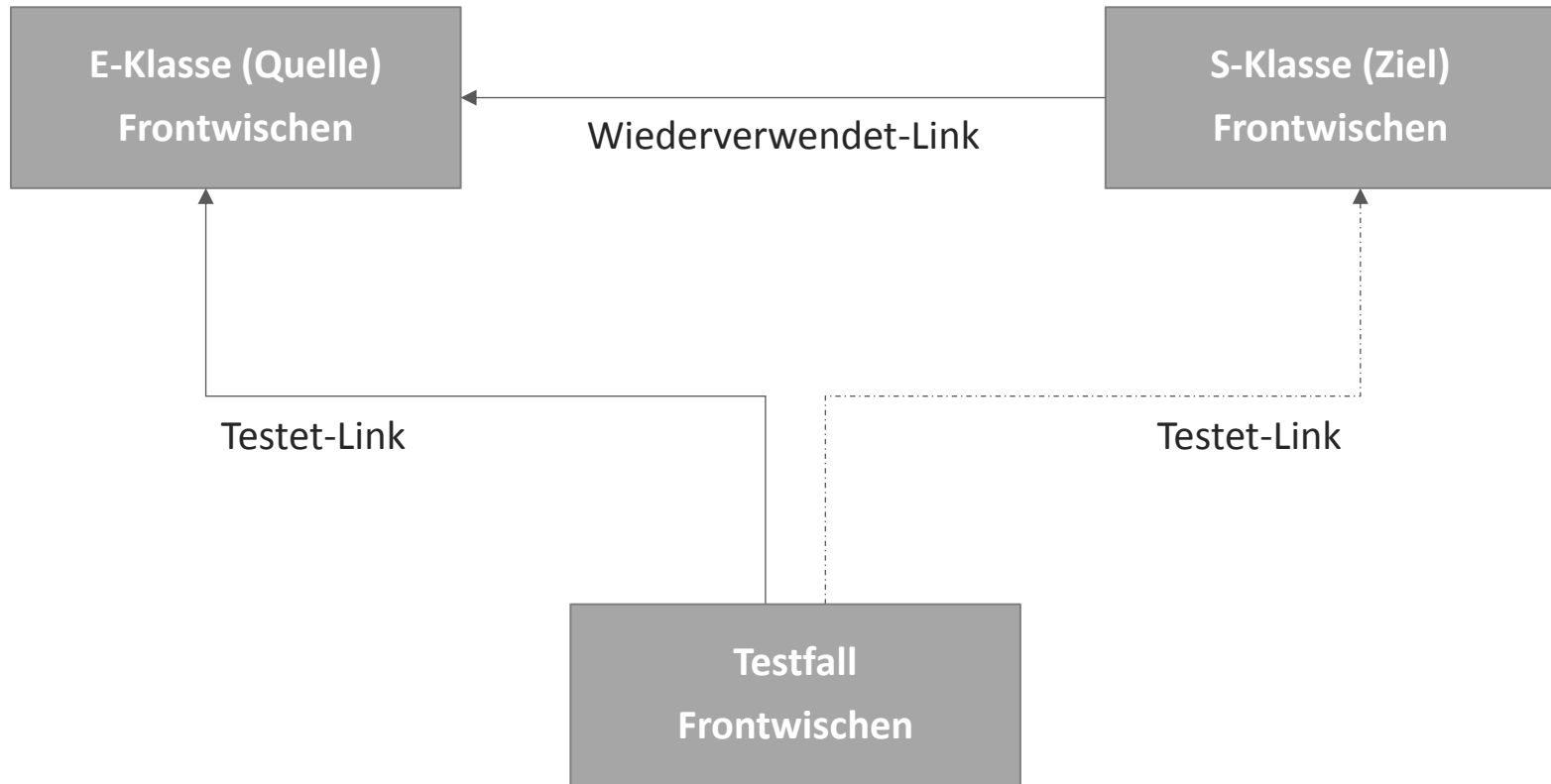
2




3-SCHICHTIGE METHODE: ÜBERBLICK

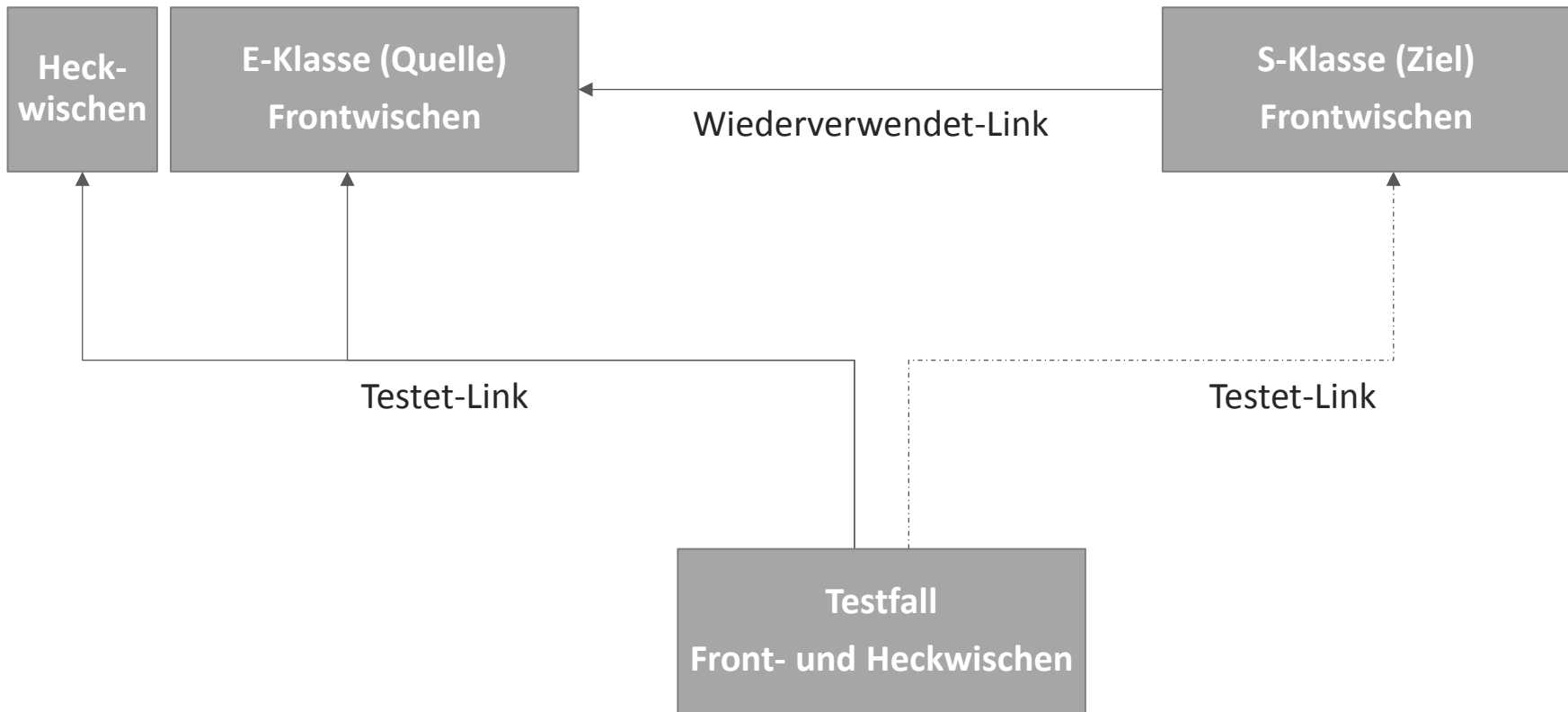


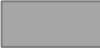


Jede Schicht hat 3 Phasen

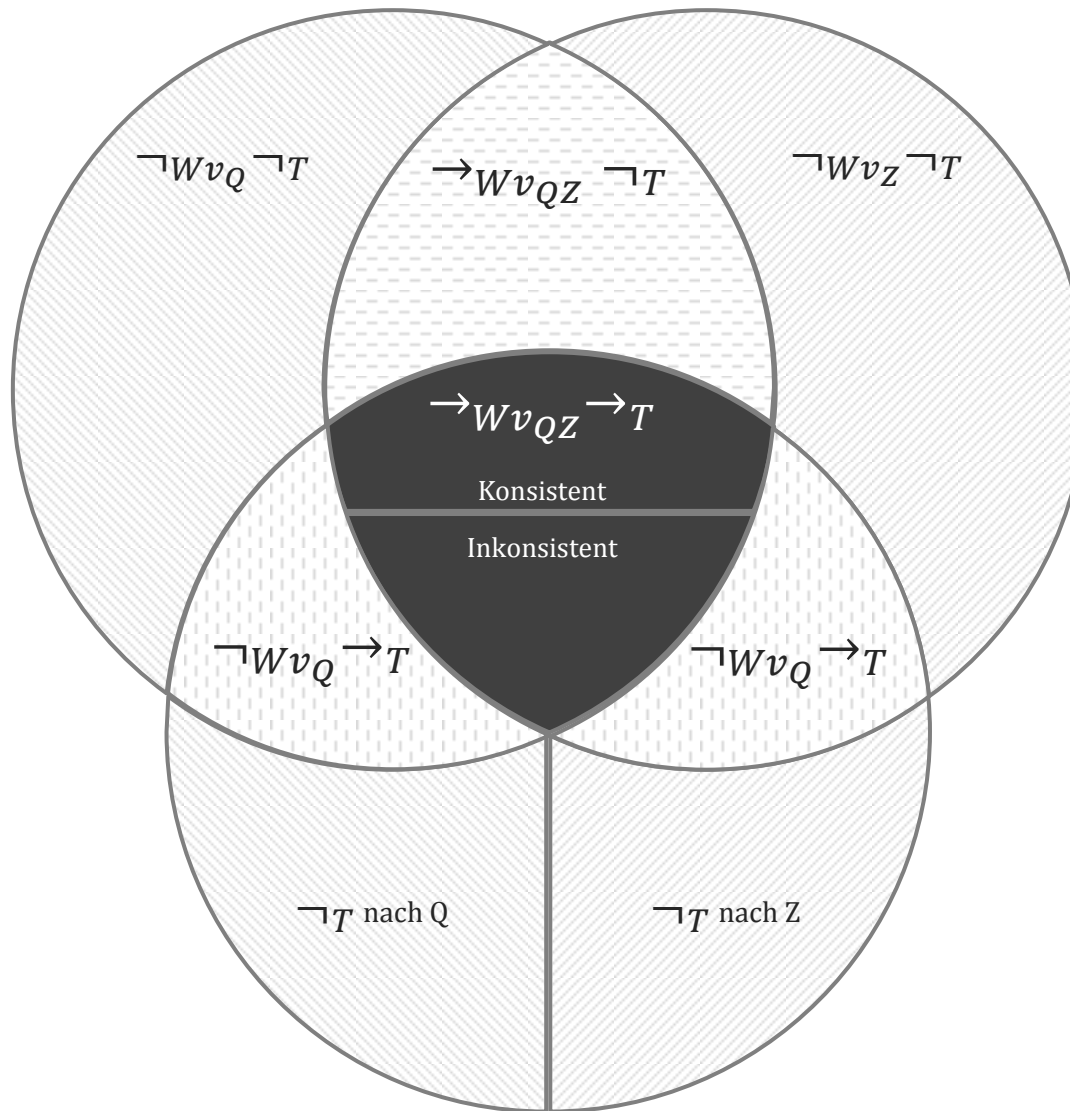


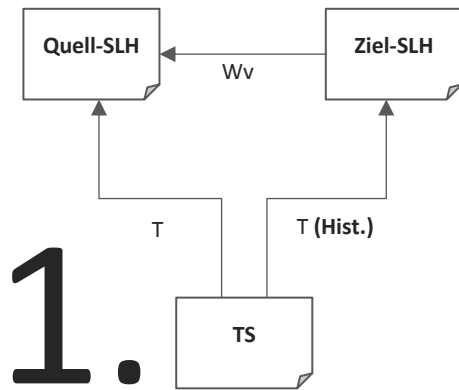


-  Schicht 1
-  Link bekannt
-  Link unbekannt

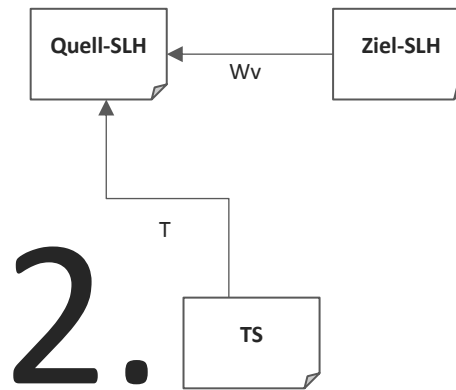


-  Schicht 1
-  Link bekannt
-  Link unbekannt

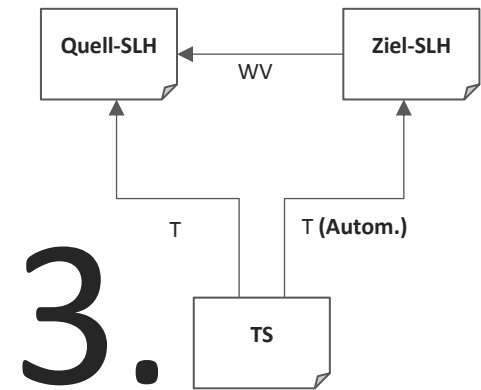




Historische Links analysieren

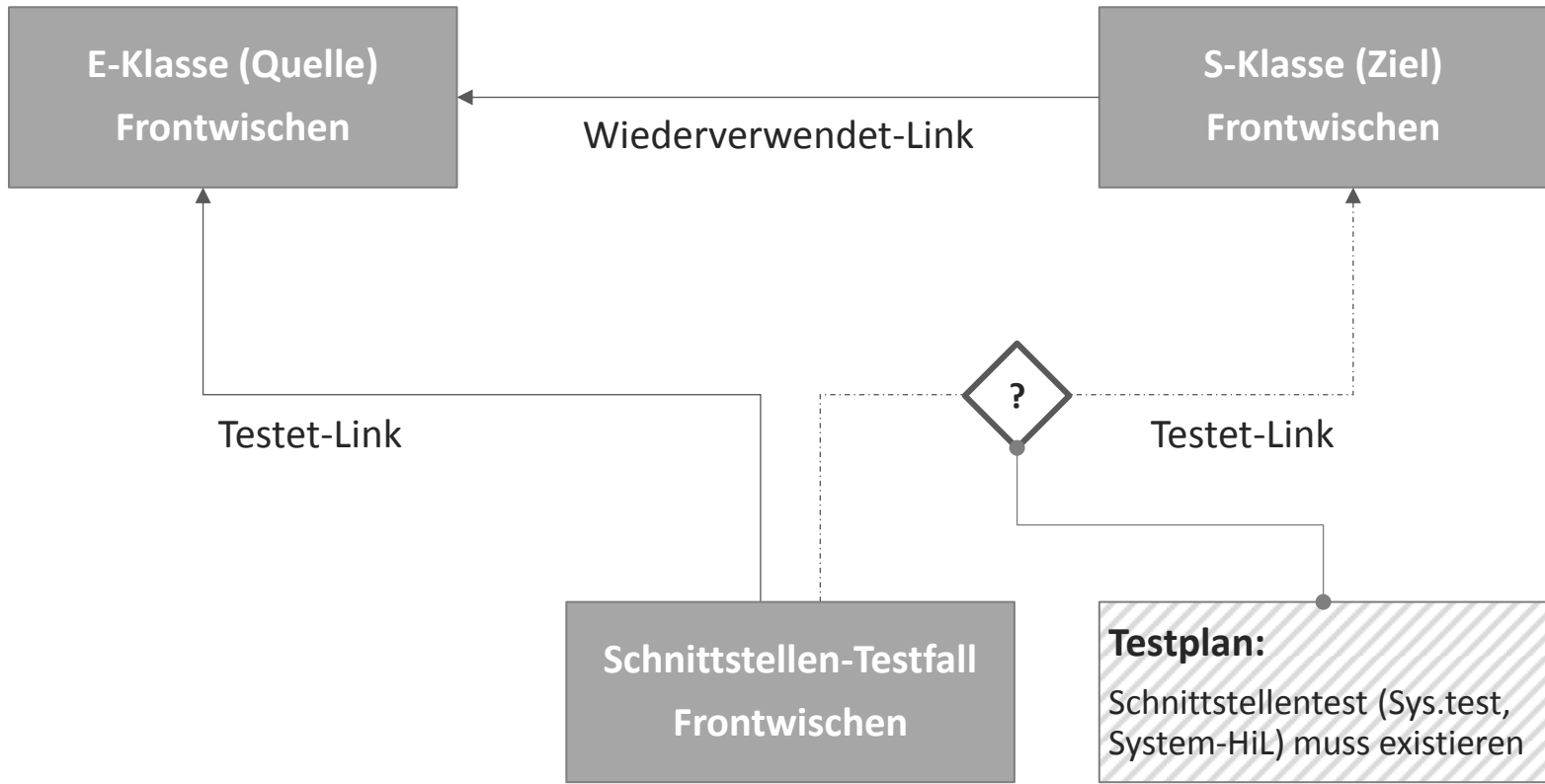


Historische Links löschen



Links automatisch setzen


- **Effektivität:**
 - Alle Links, die historisch existierten, existieren auch nach autom. Verlinkung
 - Mehr konsistente Links existieren nach automatischer Verlinkung
- **Effizienz:** Automatische Verlinkung dauert Minuten statt Tage




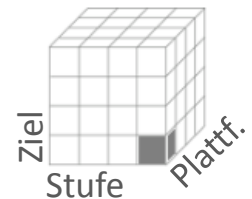
 Schicht 1

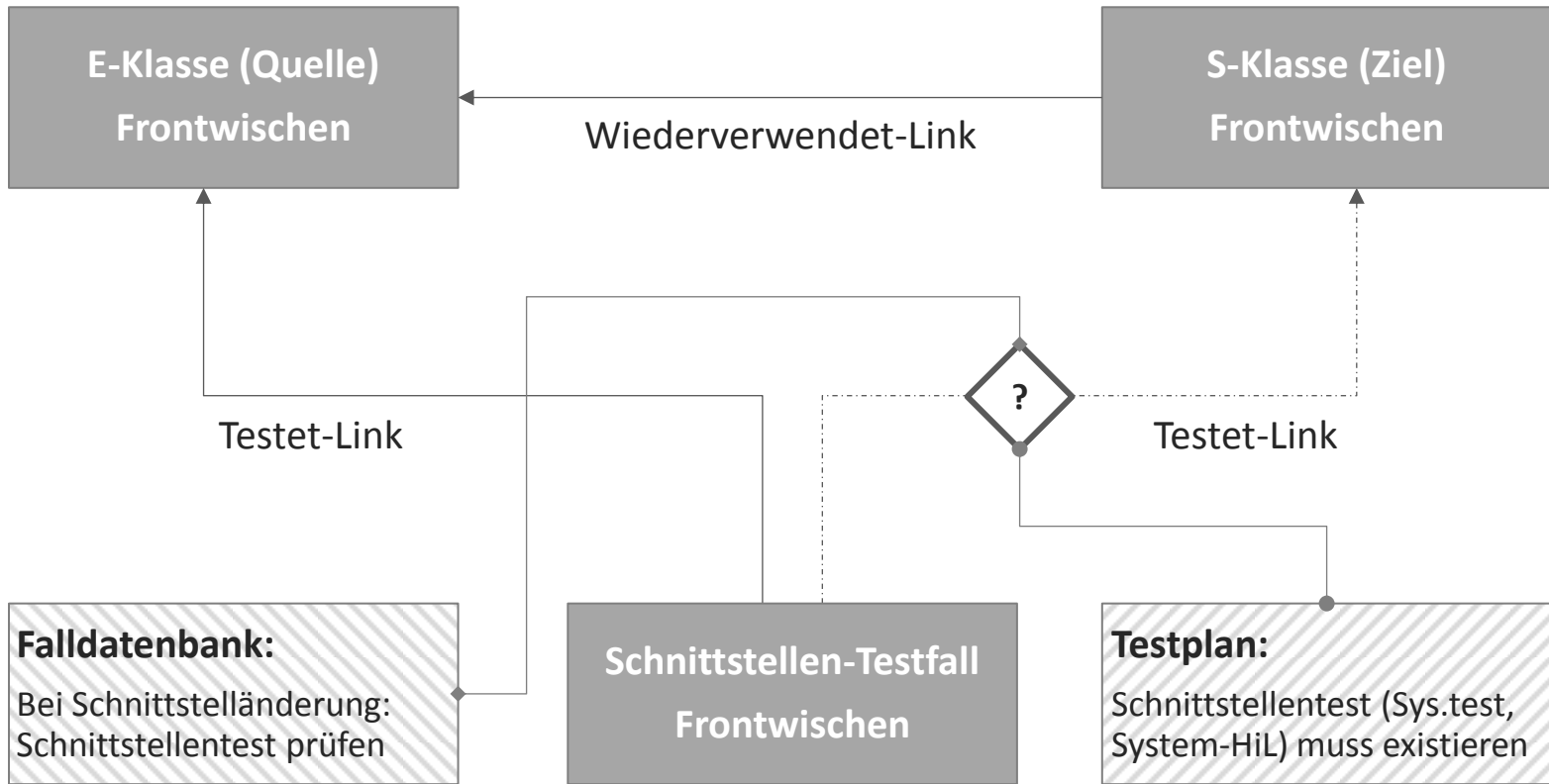
 Schicht 2

 Link bekannt

 3 Testplan-
dimensionen

 Link unbekannt





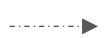



 Schicht 3

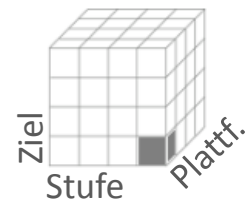
 Schicht 1

 Schicht 2

 Ähnliche
frühere WvT
Fälle

 Link bekannt
 Link unbekannt

 3 Testplan-
dimensionen

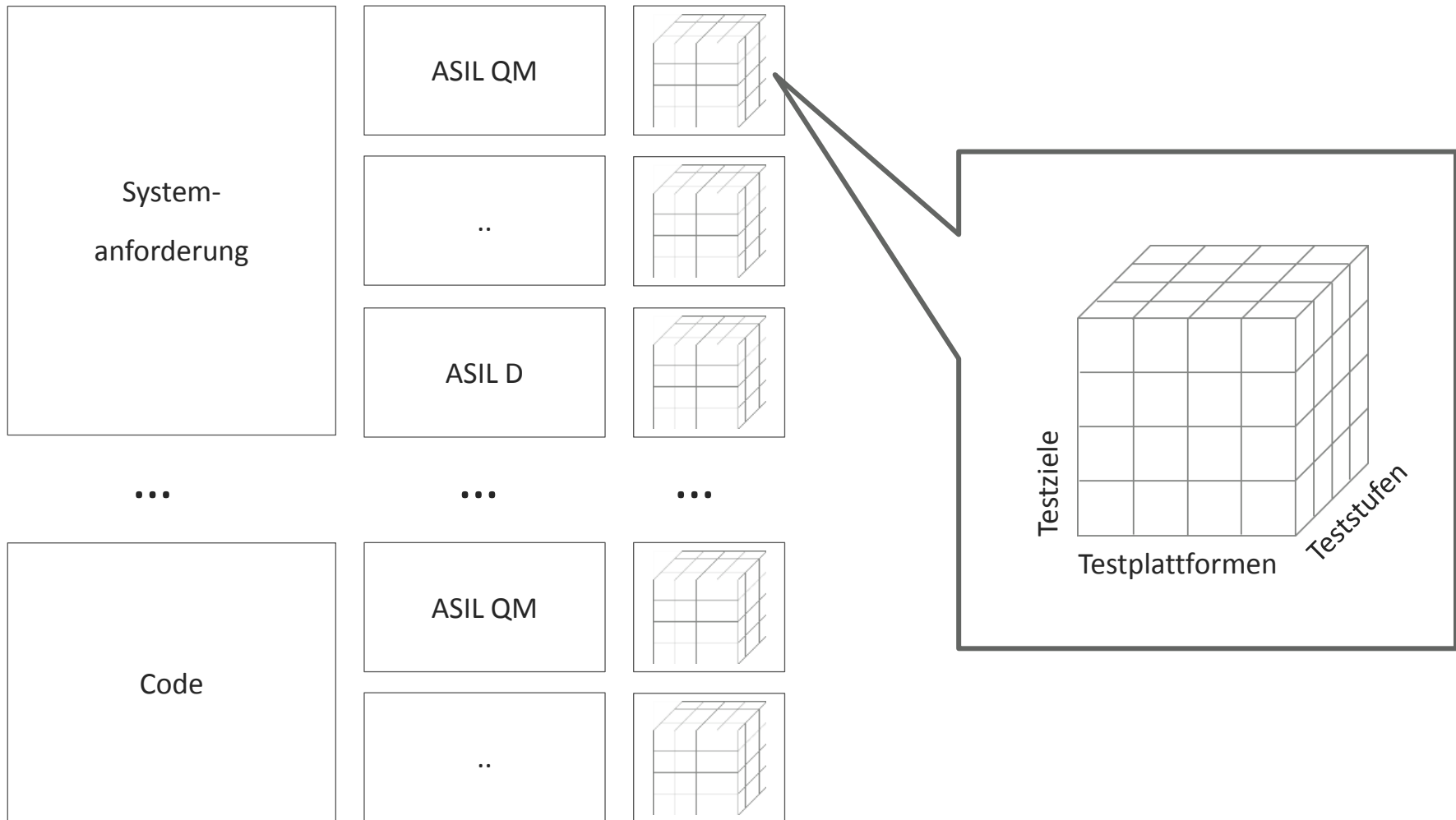


3

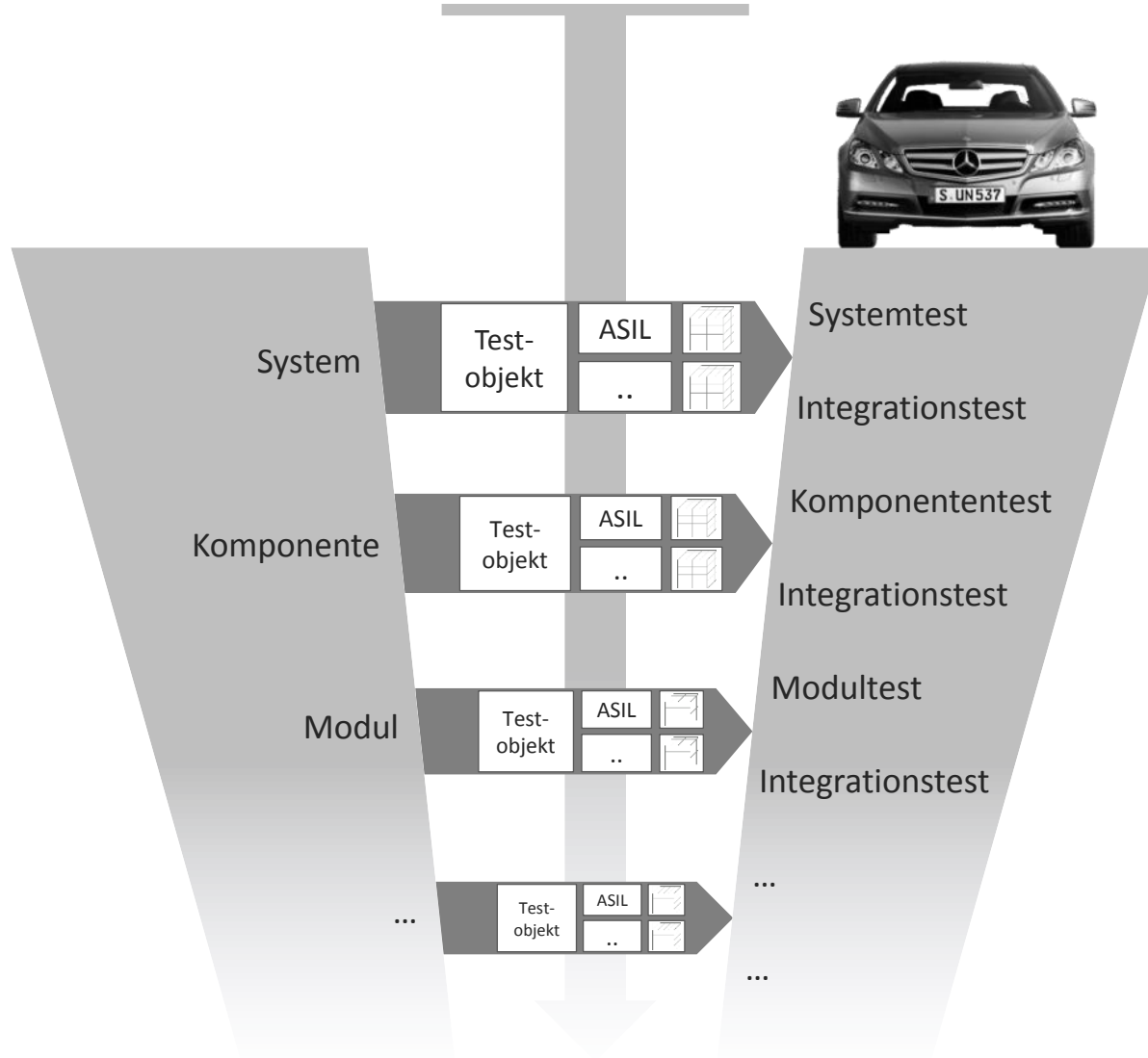
SCHICHT 2: TESTPLANGESTEUERTE FILTERUNG

- Rechtliche Grundlagen
 - Produzentenhaftung
 - Produkthaftung
 - Stand der Technik (vs. Branchenüblichkeit)
- Rückrufaktionen (ADAC Datenbank)
- Das Airbag-Urteil

Rechtliche Folgen für den Fahrzeugtest



Testkoordination



$$dim_{Objekt} = \{o \mid o \text{ ist Testobjekt-Typ}\}$$

$$dim_{ASIL} = \{a \mid a \text{ ist ASIL Stufe}\}$$

$$dim_{Stufe} = \{s \mid s \text{ ist Teststufe}\}$$

$$dim_{Plattform} = \{p \mid p \text{ ist Testplattform}\}$$

$$dim_{Ziel} = \{z \mid z \text{ ist Testziel}\}$$

$$V_{Links} = dim_{Objekt} \times dim_{ASIL}$$

$$V_{Rechts} = dim_{Stufe} \times dim_{Plattform} \times dim_{Ziel}$$

$$\text{Testplan-Hyperwürfel} = V_{Links} \times V_{Rechts}$$

$$\text{Testplan} \subseteq \text{Testplan-Hyperwürfel}$$

$$\text{Anforderung} = \text{UID} \times \text{Text} \times \text{Schnittstellen} \times \dots$$

$$\text{Anforderung}_{TP} = V_{Links} \times \text{Anforderung}$$

$$\text{Testfall} = \text{UID} \times \text{Vorbedingungen} \times \text{Testschritte} \times \dots$$

$$\text{Testfall}_{TP} = \wp(V_{Rechts}) \times \text{Testfall}$$

$$\text{vereine} : \text{Anforderung}_{TP} \times \text{Testfall}_{TP} \rightarrow \wp(\text{Testpl.} - \text{Hyperw.})$$

$$\text{vereine}(anf, test) = \{((o, a), (s, p, z)) \mid (s, p, z) \in test_{3D}\}$$

$$anf = (o, a, anf_{Rest})$$

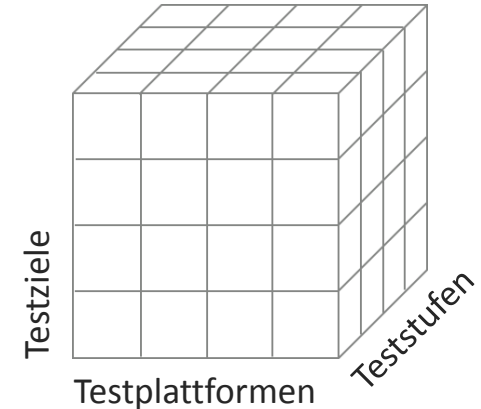
$$test = (test_{3D}, test_{Rest})$$

$$\text{verlinkeTransitiv}(a_Z, a_Q, t) = a_Z \rightarrow_{Wv} a_Q \wedge t \rightarrow_T a_Q \Rightarrow t \rightarrow_T a_Z$$

$$\text{koordiniereVerlinkung}(a_Z, t) \Rightarrow \text{verlinkeTransitiv}(a_Z, a_Q, t)$$

$$\text{koordiniereVerlinkung}(a_Z, t) = \exists at_{5D} \in \text{vereine}(a_Z, t) :$$

$$at_{5D} \in \text{Testplan}$$



vollständig(a_Z) = $\forall tp_{5D} \in Testplan : \exists t \in T(a_Z) :$

$\exists at_{5D} \in vereine(a_Z, t) : tp_{5D} = at_{5D}$

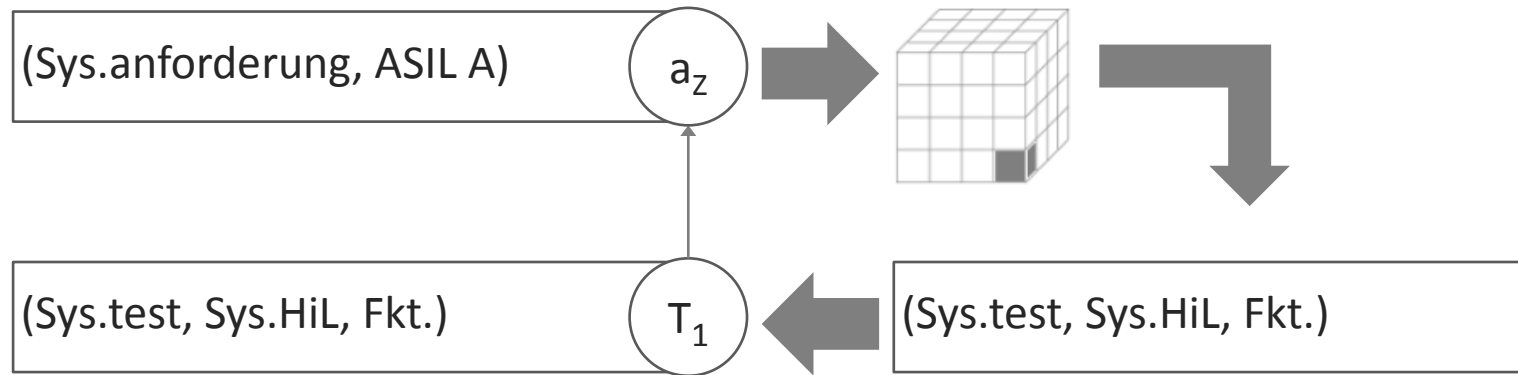
minimal(a_Z) = $\forall t \in T(a_Z) : \forall at_{5D} \in vereine(a_Z, t) :$

$\exists tp_{5D} \in Testplan : tp_{5D} = at_{5D}$

	\neg vollständig	\neg minimal
0: Konsistent	0	0
1: Vollständig und nicht minimal	0	1
2: Unvollständig und minimal	1	0
3: Unvollständig und nicht minimal	1	1

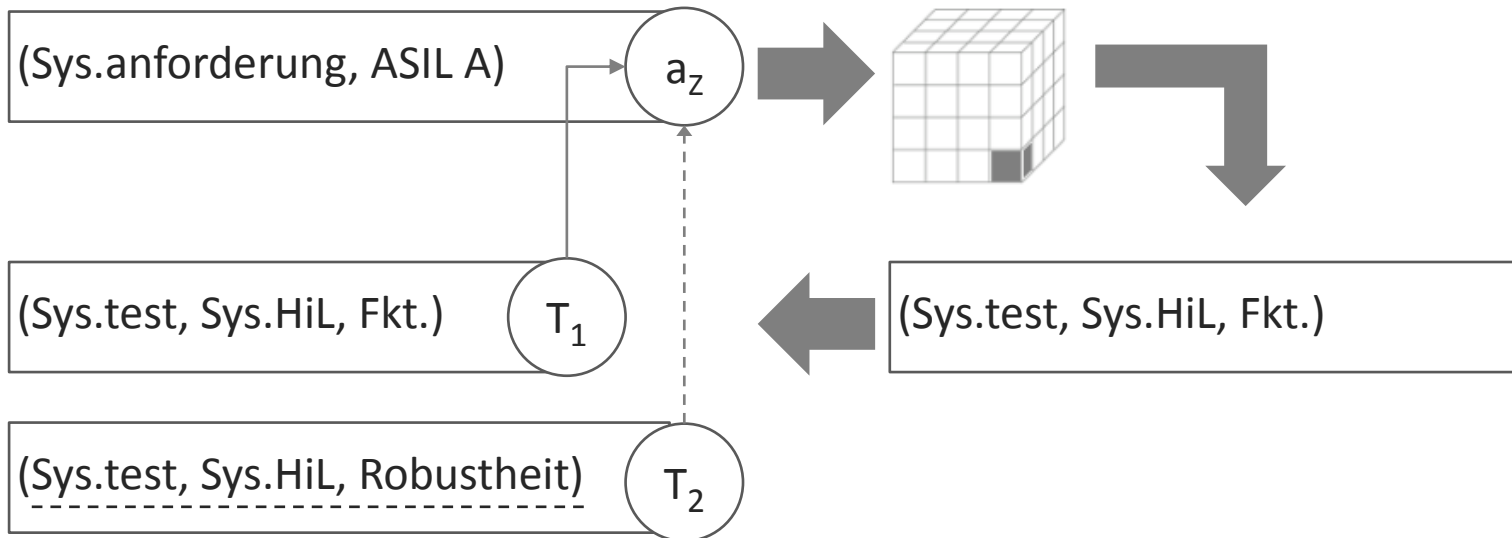
$\text{vollständig}(a_Z) = \forall tp_{5D} \in \text{Testplan} : \exists t \in T(a_Z) :$
 $\quad \exists at_{5D} \in \text{vereine}(a_Z, t) : tp_{5D} = at_{5D}$
 $\text{minimal}(a_Z) = \forall t \in T(a_Z) : \forall at_{5D} \in \text{vereine}(a_Z, t) :$
 $\quad \exists tp_{5D} \in \text{Testplan} : tp_{5D} = at_{5D}$

	\neg vollständig	\neg minimal
0: Konsistent	0	0
1: Vollständig und nicht minimal	0	1
2: Unvollständig und minimal	1	0
3: Unvollständig und nicht minimal	1	1



$\text{vollständig}(a_Z) = \forall tp_{5D} \in \text{Testplan} : \exists t \in T(a_Z) :$
 $\quad \exists at_{5D} \in \text{vereine}(a_Z, t) : tp_{5D} = at_{5D}$
 $\text{minimal}(a_Z) = \forall t \in T(a_Z) : \forall at_{5D} \in \text{vereine}(a_Z, t) :$
 $\quad \exists tp_{5D} \in \text{Testplan} : tp_{5D} = at_{5D}$

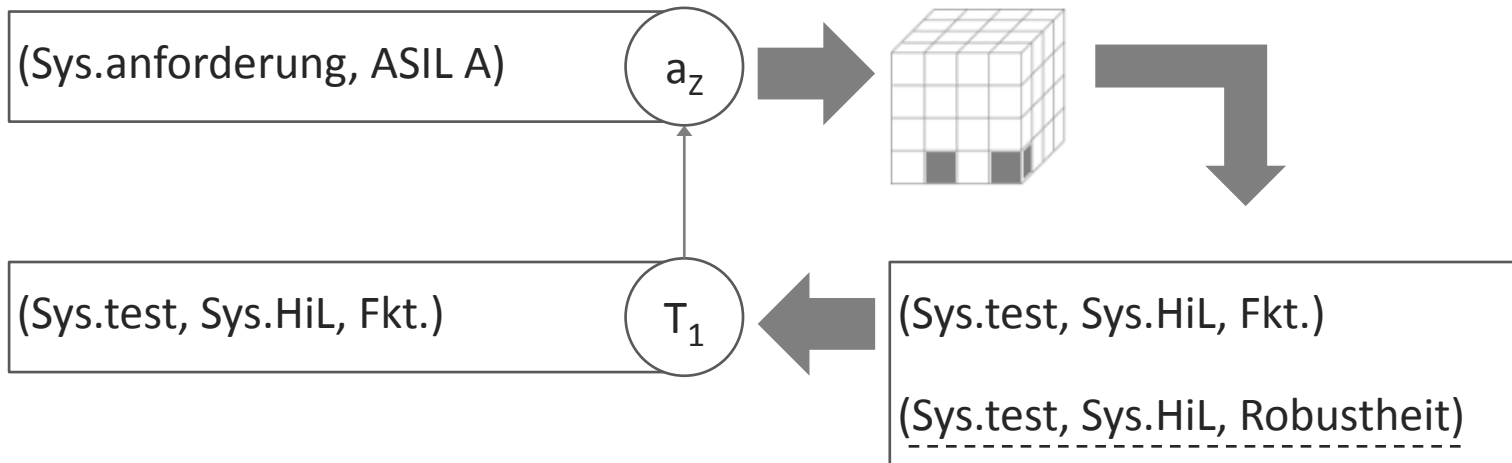
	\neg vollständig	\neg minimal
0: Konsistent	0	0
1: Vollständig und nicht minimal	0	1
2: Unvollständig und minimal	1	0
3: Unvollständig und nicht minimal	1	1



$$\text{vollständig}(a_Z) = \forall tp_{5D} \in \text{Testplan} : \exists t \in T(a_Z) : \\ \exists at_{5D} \in \text{vereine}(a_Z, t) : tp_{5D} = at_{5D}$$

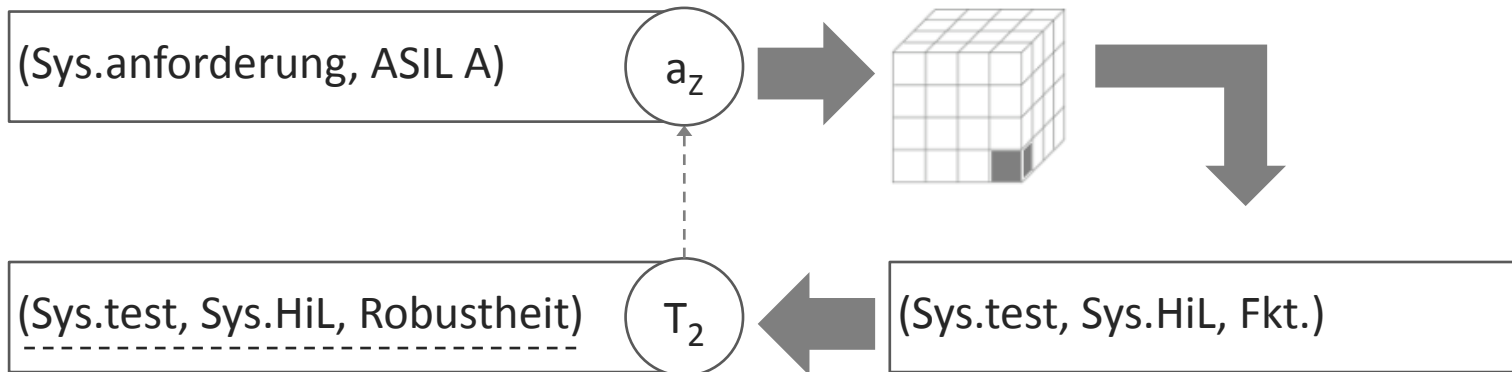
$$\text{minimal}(a_Z) = \forall t \in T(a_Z) : \forall at_{5D} \in \text{vereine}(a_Z, t) : \\ \exists tp_{5D} \in \text{Testplan} : tp_{5D} = at_{5D}$$

	\neg vollständig	\neg minimal
0: Konsistent	0	0
1: Vollständig und nicht minimal	0	1
2: Unvollständig und minimal	1	0
3: Unvollständig und nicht minimal	1	1



$\text{vollständig}(a_Z) = \forall tp_{5D} \in \text{Testplan} : \exists t \in T(a_Z) :$
 $\quad \exists at_{5D} \in \text{vereine}(a_Z, t) : tp_{5D} = at_{5D}$
 $\text{minimal}(a_Z) = \forall t \in T(a_Z) : \forall at_{5D} \in \text{vereine}(a_Z, t) :$
 $\quad \exists tp_{5D} \in \text{Testplan} : tp_{5D} = at_{5D}$

	\neg vollständig	\neg minimal
0: Konsistent	0	0
1: Vollständig und nicht minimal	0	1
2: Unvollständig und minimal	1	0
3: Unvollständig und nicht minimal	1	1



Testplan

Testplan		
Testobjekt-Typ:	Teststufe	
Fahrzeugfunktion	Int	Sys
Testziele: Prüfe ..		
.. Funktionalität	X	X
.. Schnittstelle	X	
.. Konfigurierbarkeit	X	
Testplattformen:		
HiL	X	
Fahrzeug		X

Quell-SLH

Anforderungen	Ähnlichkeit	Wv?	T?	Inkonsistent	Kommentar
1 Funktion Q					
Q 1	▶	Ja	Ja		
Q 2	▶	Ja	Prüfen		- Textuelle Änderung
Q 3	▶	Ja	Prüfen	I_Q	- Textuelle Änderung - Test 3 verlinkt mit Q3 aber nicht mit Z3
Q 4		Nein	Nein		

Testspezifikation

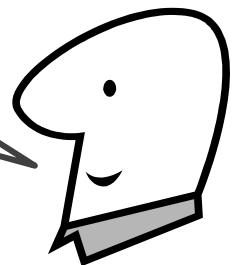
Testfälle	Verlinkt zu	Testziel	Teststufe	Testplattform
1 Tests				
Test 1	▶ SLH-Q SLH-Z	Funktionalität	Int Sys	HiL Fzg
Test 2	▶ SLH-Q SLH-Z	Schnittstelle	Int	HiL
Test 3	▶ SLH-Q SLH-Z	Konfigurierbarkeit	Sys	HiL

Ziel-SLH

Anforderungen	Ähnlichkeit	Wv?	T?	Inkonsistent	Kommentar
1 Funktion Z					
Z 1: Gleich	▶▶▶ 100	Ja	Ja		
Z 2: Fast gleich	▶▶▶ 80	Ja	Prüfen		- Textuelle Änderung
Z 3: Ungleich	▶▶ 60	Ja	Prüfen	I_Q	- Textuelle Änderung - Test 3 verlinkt mit Q3 aber nicht mit Z3
Z 4: Sehr ungleich		Nein	Nein		
Z 5: Neu		Nein	Nein		

- 3-schichtige Methode zur Verlinkung von Testfällen und Anforderungen
- **1. Schicht:** Transitive Verlinkung **implementiert** und **Feldstudien durchgeführt**
- **2. Schicht:** Testplangesteuerte Filterung **implementiert**
- **3. Schicht:** Fallbasierte Filterung **implementiert**

Ich ziehe Trace Links zwischen Testfällen und wiederverwendeten SLH-Anforderungen jetzt automatisch. Damit spare ich bei jedem Baureihenprojekt Zeit!





Vielen Dank