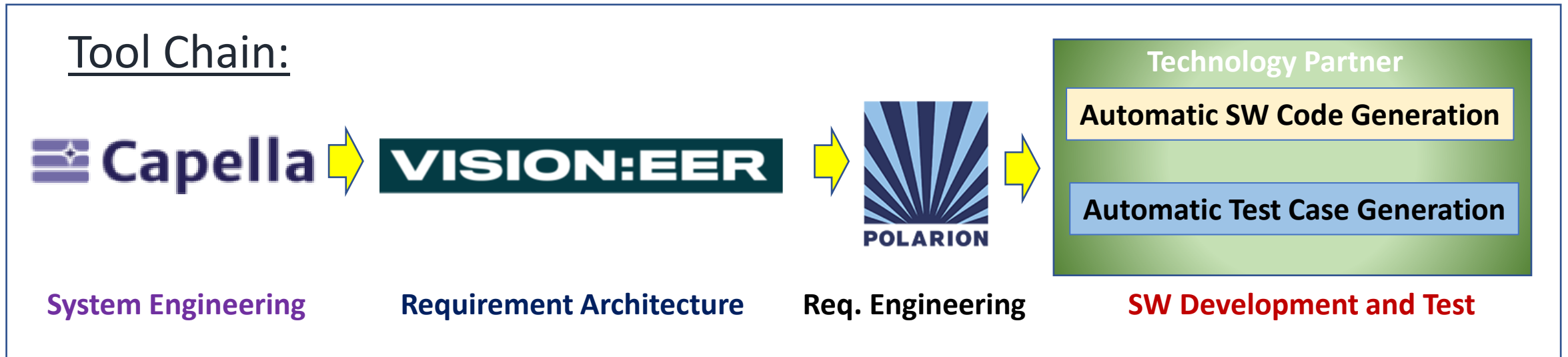


# Modelbasierte Entwicklung für Embedded Systems



*Single-Source-of-Truth: System → Subsystem*

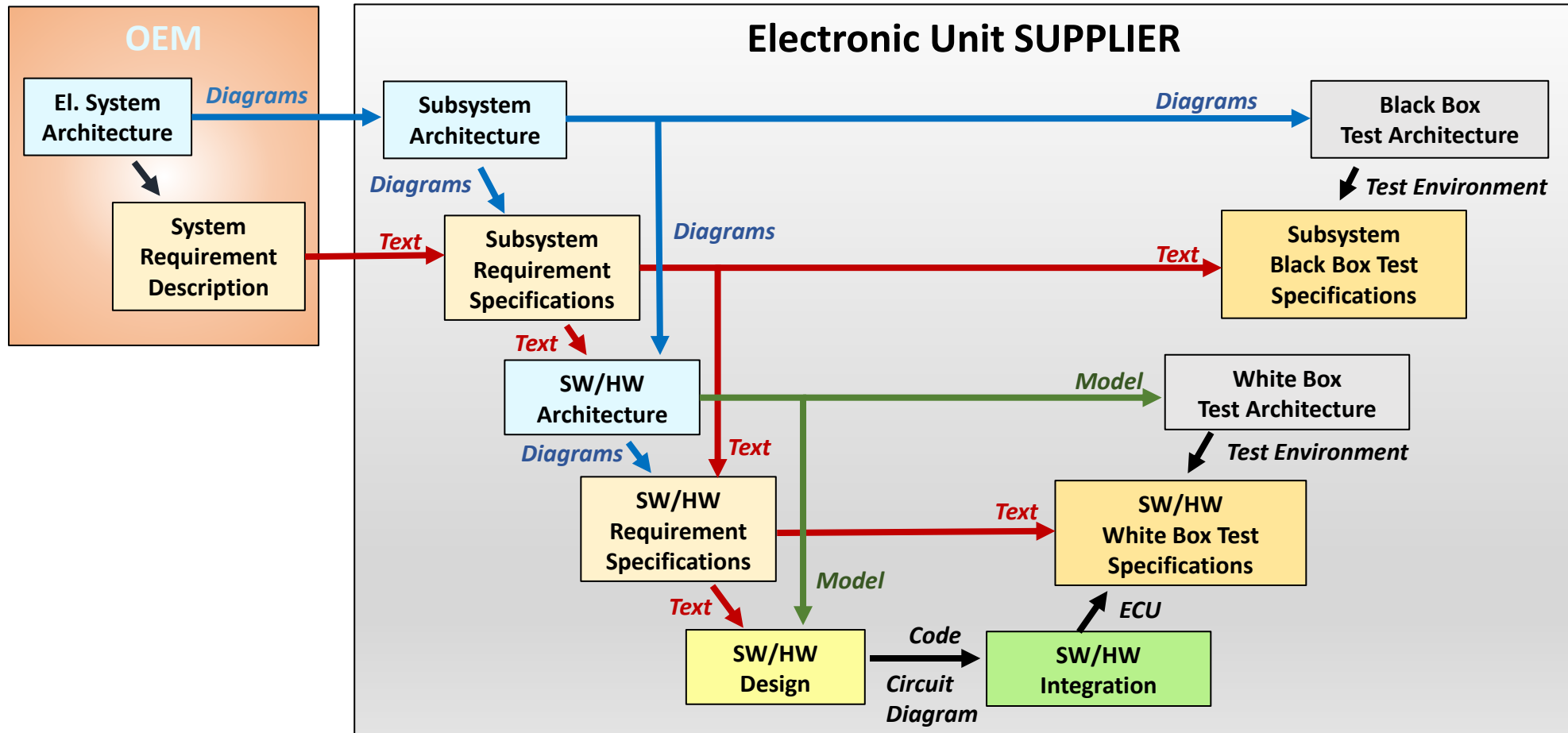
# Gelöste Probleme

1. Fehlende “Single Source of Truth” in V-Model Arbeitsprodukten (*Mehrdeutigkeit*)
  - ✓ Tool Chain für **Modelbasierte Entwicklung** und **System /Subsystem** Synchronisierung
  - ✓ **Objekt-Orientiertes** Requirement Engineering
2. Fehlende oder überzogene “Inhalte oder Detaillierung” der Arbeitsprodukte (*Inkonsistenz*)
  - ✓ **Generischer Baukasten** für Embedded Systems → **Vordefinierte** Anforderungs-Spezifikations-**Struktur**
3. Fehlende Anforderungen für “spezifische Situationen oder Signal Kombinationen” (*Unvollständigkeit*)
  - ✓ Automatismen für die **Interface-Architektur**, **Modes** und **Signal-Handling**
  - ✓ **Logik Tabellen** und **Vollständigkeits-Checker** für potentielle Situationen und Signal-Kombinationen
  - ✓ Automatische **Spezifikations-**, **Code-** und **Test-Case Generierung**

⇒ **Reduktion der Fehler in elektronischen Baugruppen > 50% !**

# V-Model: Austausch der Arbeitsprodukte heute

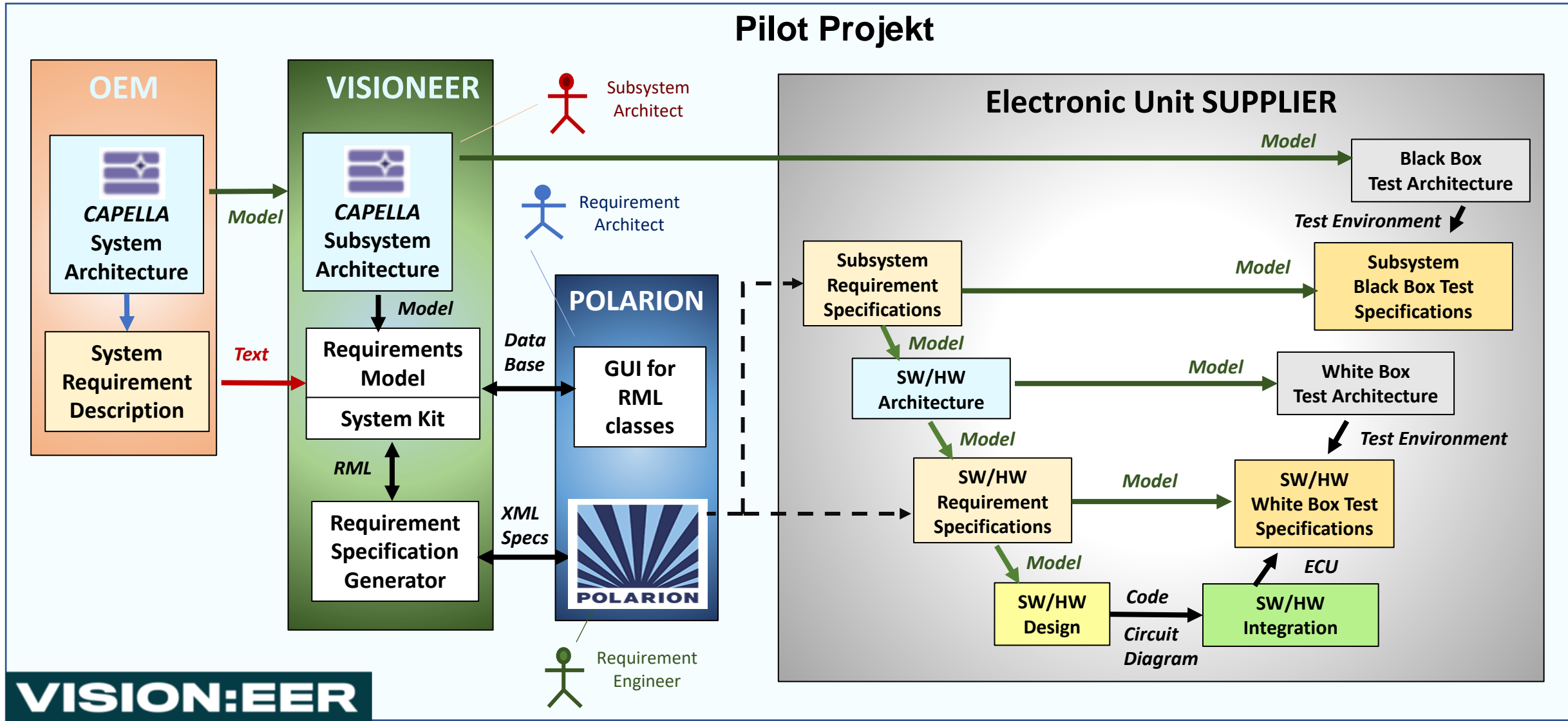
Problem: Struktur-Synchronisierung der abgeleiteten Arbeitsprodukte ist nur durch **Model-Austausch** möglich



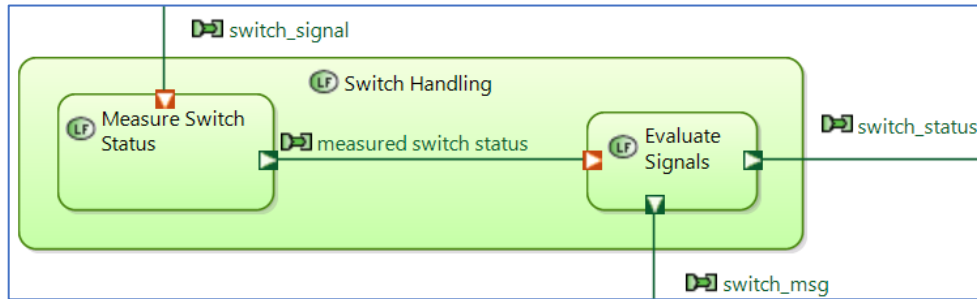
→ Keine SINGLE SOURCE OF TRUTH

# Pilot Projekt *Modellbasierte Entwicklung*

Modellbasierte Synchronisierung der Arbeitsprodukte → **SINGLE SOURCE OF TRUTH**



# Zusammenführung CAPELLA Model → RML Klassen



Automatische *Konvertierung* in RML (durch Kenntnis des IF-Typs jedes Signals)

```
Req_Block Merged_SubFs merge Switch_Handling and SF {
  INFO Scope;
  REQ Execution_Rate;
  MTHD Interface_Req = SF_its_IF_REQ();
  REQ_Block IN_sig_Conversion = Measure_Switch_Status:AI_CONV;
  REQ_Block (0..*) IN_msg_Handling;
  REQ_Block Signal_processing = Evaluate_Signals_SIG_PROC;
  REQ_Block OUT_msg_Handling = switch_msg: CAN_OUT; }
```

RML (Requirement Model Language)  
→ **Objectorientiertes Requirement Engineering**

## SRS\_P1\_SWITCH

3	Func_Req_dscr = P1_Switch_Handling	2
3.1	Scope	2
3.2	Execution_Rate	2
3.3	Interface_Req = Switch_its_IF_REQ	2
3.3.1	Input_Signals = Switch_its_Input_Signals	2
3.3.2	Input_SW_IFs	2
3.3.3	Input_Frames	3
3.3.4	Input_Messages	3
3.3.5	Output_Signals	3
3.3.6	Output_SW_IFs = Switch_its_Output_SW_IFs	3
3.3.7	Output_Frames = Switch_its_Output_Frames	3
3.3.8	Output_Messages = Switch_its_Output_Messages	3
3.3.9	Parameter = Switch_its_Parameter	3
3.3.10	Diag_Services	3
3.3.11	Events = Switch_its_Events	4
3.4	Measure_Switch_Status = switch_signal:AN_IN_CONV	4
3.5	Signal_Processing = Evaluate_Signals	4

Vererbung

Vererbung

```
Req_Block SF {
  INFO Scope;
  REQ Execution_Rate;
  MTHD Interface_Req = SF_its_IF_REQ();
  REQ_Block (0..*) IN_sig_Conversion = xx:xx_IN_CONV;
  REQ_Block (0..*) IN_msg_Handling = xx:xx_IN;
  REQ_Block (1..*) Signal_processing = xx_SIG_PROC;
  REQ_Block (0..*) OUT_sig_Conversion = xx:xx_OUT_CONV;
  REQ_Block (0..*) OUT_msg_Handling = xx:xx_OUT; }
```

*System Kit*

```
Project File
Req_Block P1_Switch_Handling extends Merged_SubFs {
  INFO Scope = "Switch_Handling its SF specific requirements";
  REQ Execution_Rate;
  MTHD Interface_Req = SF_its_IF_REQ();
  REQ_Block IN_sig_Conversion = Measure_Switch_Status:AI_CONV;
  REQ_Block Signal_processing = Evaluate_Signals_SIG_PROC;
  REQ_Block OUT_msg_Handling = switch_msg: CAN_OUT; }
```

Automatische *Konvertierung* von RML in Anforderungsspezifikationen



**Ergebnis:** Das Capella-Modell wird um alle erwarteten Spezifikationselemente (Baukasten) für Embedded Systems erweitert

# Signalverarbeitung mittels Logik-Tabelle

Automatisch generierte Logiktablelle, welche alle funktionspezifischen Situationen u. Modes abdeckt

Manuelle Finalisierung der logischen Kombinationen

**X = don't care**

→ Zeigt die Dominanz eines Signals bei der Auswertung des Ausgangssignals an

Automatischer Test, ob jede mögliche Signal-Kombination definiert ist

SRDS-611 - switch\_status\_its\_Logic\_table

SWITCH_PARAM	LIFE_CYCLE_PHASE	CAR_MODES	ECU_MODES	Emergency_off_Mode	Disturbance_Modes	measured_switch_status	switch_status
enable	OPERATION or CAR_FACTORY	OCCUPANCY or KEY_POS_1 or DRIVING	WAKE_UP or DIAGNOSTIC	SAFE_OPERATION	NORMAL_VOLTAGE	OPEN	<b>OPEN</b>
enable	OPERATION or CAR_FACTORY	OCCUPANCY or KEY_POS_1 or DRIVING	WAKE_UP or DIAGNOSTIC	SAFE_OPERATION	NORMAL_VOLTAGE	FAULTY or CLOSED	<b>CLOSED</b>
enable	OPERATION or CAR_FACTORY	OCCUPANCY or KEY_POS_1 or DRIVING	WAKE_UP or DIAGNOSTIC	SAFE_OPERATION	LOW_VOLTAGE or HIGH_VOLTAGE	x	<b>FROZEN</b> (previous value)
x	x	x	x	FAIL_SAFE_MODE	x	x	<b>CLOSED</b>
x	x	x	START_UP or SLEEP or POWER_DOWN	x	x	x	<b>CLOSED</b>
x	x	PARKING or CRANKING or MOTOR_STOP	x	x	x	x	<b>CLOSED</b>
x	ECU_PRODUCTION_or CAR_TRANSPORT	x	x	x	x	x	<b>CLOSED</b>
disable	x	x	x	x	x	x	<b>CLOSED</b>

Windows aktivieren  
Wechseln Sie zu den Einstellungen, um Windows zu

# Automatische Generierung Funktionaler Anforderungen

Die Logik-Tabelle wird  
in funktionale  
Textanforderungen  
umgewandelt :

## 3.5.4.2 Functional\_Requirements = switch\_status\_its\_text\_conversion

(Note: Those textual requirements are automatically created out of the logic table)

**SRDS-577** - If SWITCH\_PARAM is enable

and LIFE\_CYCLE\_PHASE is OPERATION or CAR\_FACTORY

and CAR\_MODE is OCCUPANCY or KEY\_POS\_I or DRIVING

and ECU\_MODES is WAKE\_UP or DIAGNOSTIC

and Emergency\_off\_Mode is SAVE\_OPERATION

and Disturbance\_Modes is NORMAL\_VOLTAGE

and measured\_switch\_status is OPEN,

then **switch\_status** shall be OPEN.

**SRDS-575** - If SWITCH\_PARAM is enable

and LIFE\_CYCLE\_PHASE is OPERATION or CAR\_FACTORY

and CAR\_MODE is OCCUPANCY or KEY\_POS\_I or DRIVING

and ECU\_MODES is WAKE\_UP or DIAGNOSTIC

and Emergency\_off\_Mode is SAVE\_OPERATION

and Disturbance\_Modes is NORMAL\_VOLTAGE

and measured\_switch\_status is FAULTY or CLOSED,

then **switch\_status** shall be CLOSED.

Durch die **Formalisierung**  
können die **funktionalen**  
**Anforderungen** für die  
**automatische Generierung**  
von **Code-** und  
**Testspezifikationen**  
verwendet werden

# *Fragen?*

Bitte kontaktieren Sie mich:

Gerhard Schilling

Tel. +49 179 3245588

[schilling@visioneer.info](mailto:schilling@visioneer.info)

[www.visioneer.info](http://www.visioneer.info)