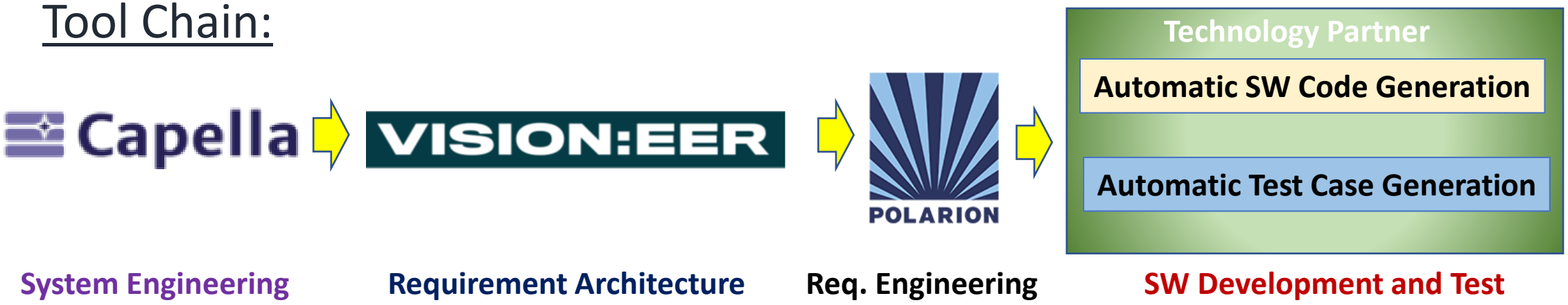
Model Based Development
for Embedded Systems

Tool Chain:



Single-Source-of-Truth: System → Subsystem

v1.1

Solved Problems

1. Missing **“Single Source of Truth”** in V-Model work products (*ambiguity)*

- ✓ Tool Chain for **Model Based Development** and **System** to **Subsystem** synchronization
- ✓ **Object-oriented** Requirement Engineering

2. Missing or exeeding **“Contents** or **Detail-Level”** of work products (*inconsistency*)

- ✓ **Generic System Kit** for embedded systems → **Fixed** requirement specification **structure**

3. Missing requirements for “**specific situations** or **signal combinations**” (*incompleteness*)

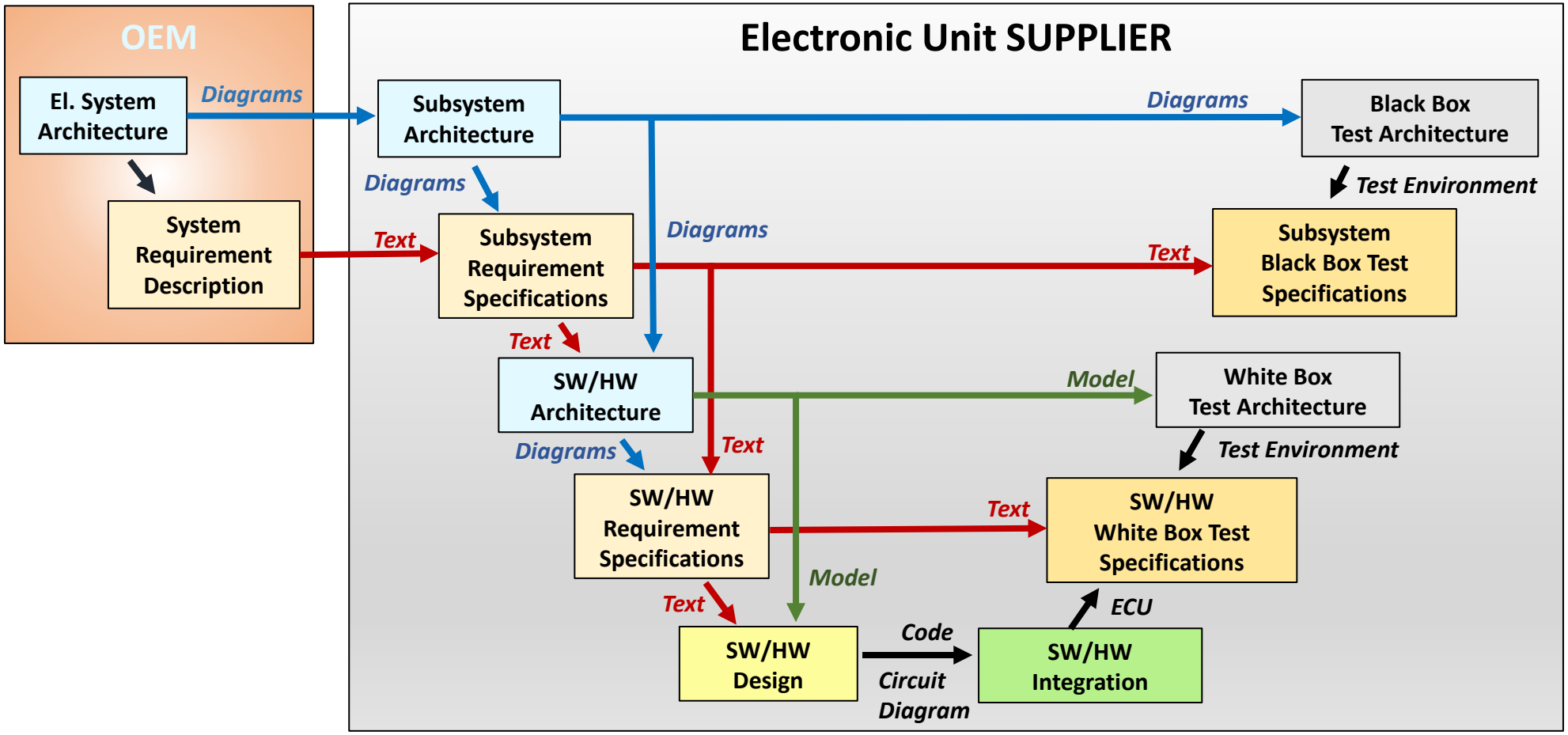
- ✓ Automatism for **Interface-Architecture**, **Modes** and **Signal-Handling**
- ✓ **Logic Tables** and **Completeness-Checker** for any potential situation and signal combination
- ✓ Automatic **Specification-**, **Code-** and **Test-Generation**

⇨ **Reduction** of **errors** in electronic units **> 50%** !

VISION:EER

V-Model: Work Product Exchanges today

Problem: **Structural Synchronisation** of **derived work products** is only possible by **Model-Exchange**

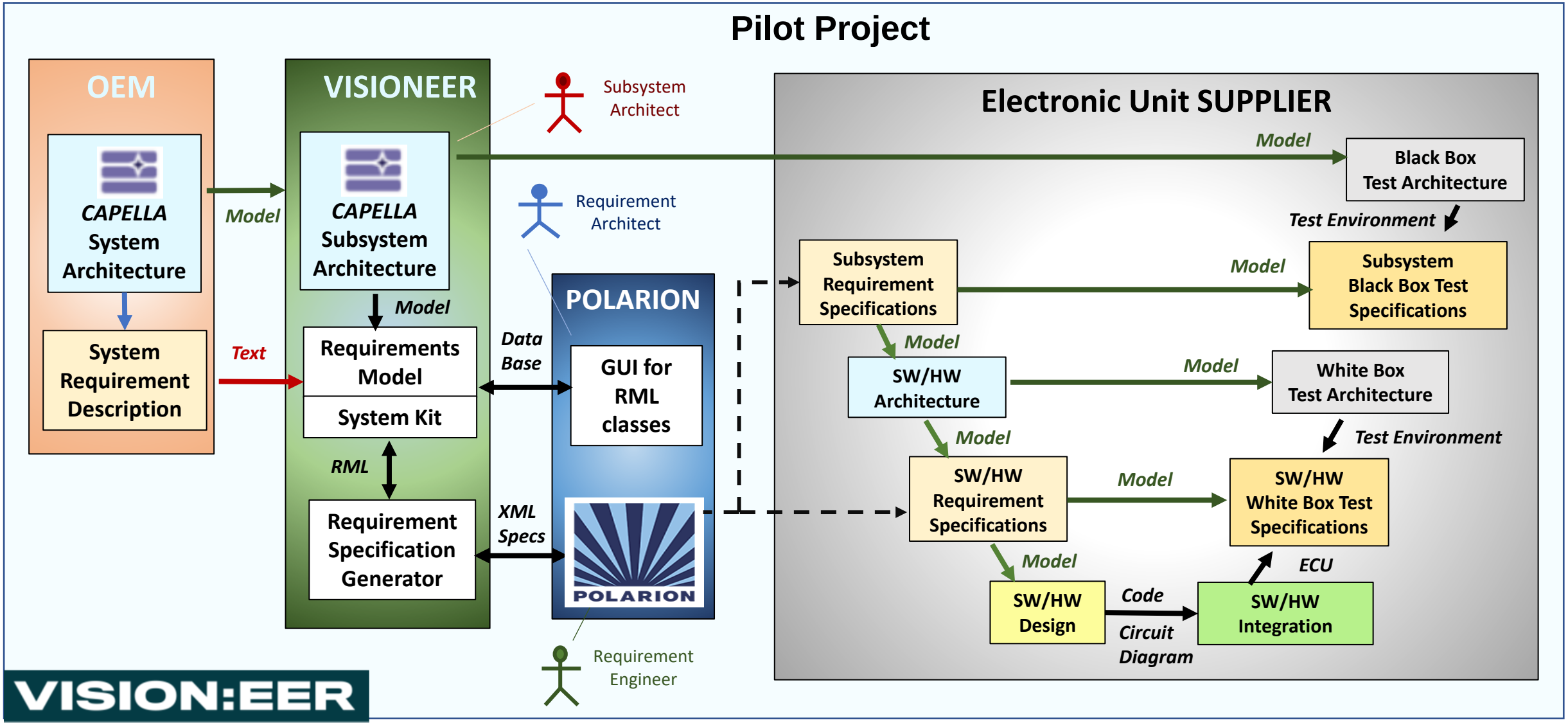


→ **NO SINGLE SOURCE OF TRUTH**

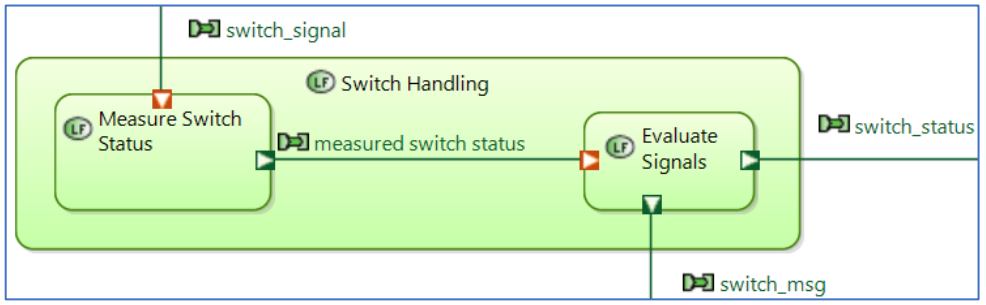
VISION:EER

Pilot Project *Model Based Development*

Model based synchronization of **work products** → **SINGLE SOURCE OF TRUTH**



Merging of CAPELLA Model → RML Classes



Automatic conversion into RML (by knowing the IF type of each signal)

```
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)
→**Object-Oriented** Requirement Engineering

SRS_P1_SWITCH

inherit

inherit

```
Req_Block  SF {                                   System Kit
 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; }
```

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; }
```

Automatic conversion of RML into Requirement Specifications

VISION:EER

Result: The Capella Model is extended with all expected specification elements (system kit) for embedded systems

Requirement Specification its Logic Table

Automatic generated logic table covering any *function specific situations*

Manually finalization of its logical combinations

X = don't care
→ Shows the dominance of a signal on the output signal evaluation

Automatic test if any potential logical combination is defined

SRDS-611 - switch_status__its_Logic_table

SWITCH_ PARAM	LIFE_CYCLE _PHASE	CAR_MODES	ECU_ MODES	Emergency_off _Mode	Distrubance _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	**OPEN**
enable	OPERATION or CAR_FACTORY	OCCUPANCY or KEY_POS_1 or DRIVING	WAKE_UP or DIAGNOSTIC	SAFE_OPERATION	NORMAL_VOLTAGE	FAULTY or CLOSED	**CLOSED**
enable	OPERATION or CAR_FACTORY	OCCUPANCY or KEY_POS_1 or DRIVING	WAKE_UP or DIAGNOSTIC	SAFE_OPERATION	LOW_VOLTAGE or HIGH_VOLTAGE	x	**FROZEN** (previous value)
x	x	x	x	FAIL_SAFE_MODE	x	x	**CLOSED**
x	x	x	START_UP or SLEEP or POWER_DOWN	x	x	x	**CLOSED**
x	x	PARKING or CRANKING or MOTOR_STOP	x	x	x	x	**CLOSED**
x	ECU_PRODUCTION_or CAR_TRANSPORT	x	x	x	x	x	**CLOSED**
disable	x	x	x	x	x	x	**CLOSED**

VISION:EER

Automatic Generation of Functional Requirements

The table is converted into textual functional requirements:

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.

Those functional requirements can be used for Automatic Code- and Test Specification-Generation

VISION:EER

Questions?

Please contact me:

Gerhard Schilling
Tel. +49 179 3245588
schilling@visioneer.info

www.visioneer.info

VISION:EER