KNORR-BREMSE

## Der Knorr-Bremse Konzern

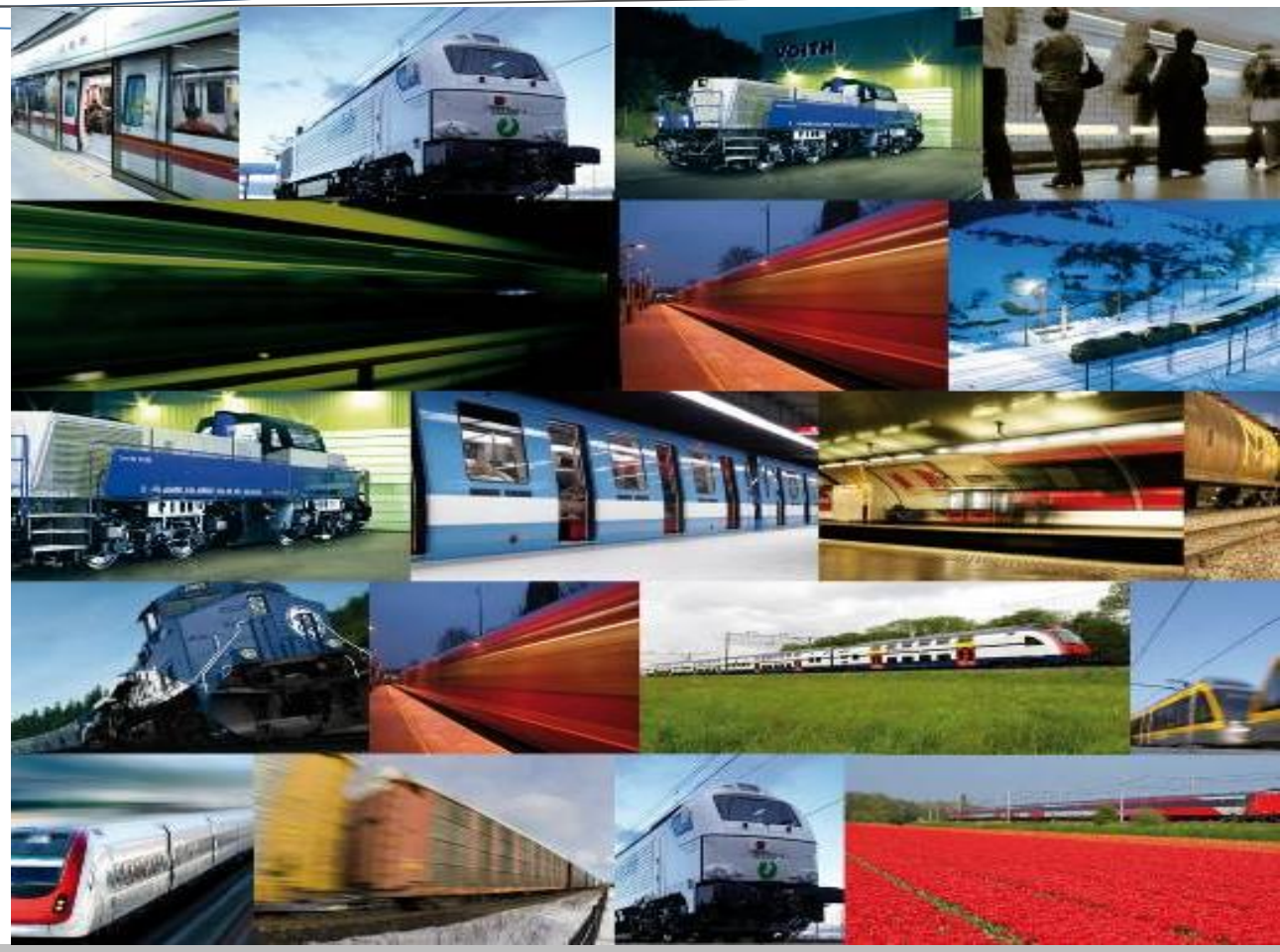
# Prozess zur Generierung einer sicherheitsrelevanten PLC-Applikation im Bahnbereich

MathWorks

Selectron

Knorr-Bremse GmbH

**Döbrössy Angelika MSc**
Team Technologie, Mödling
(Entwicklung/ Konstruktion)



# AGENDA

- Vorstellung Knorr Bremse
- Softwareentwicklung
  - Normatives Umfeld
  - Beispiel Jerk Control (Halteruckbegrenzung)
  - PLC Codegenerierung
  - Verifikation und Validierung
  - Statische Verifikation
  - Dynamische Verifikation
  - Testabdeckung – Model Coverage
  - PIL (Processor in the Loop)

# Knorr Bremse

# Mehr als eine Milliarde Menschen vertrauen täglich Systemen von Knorr-Bremse

## SYSTEME FÜR SCHIENENFAHRZEUGE

- Hochgeschwindigkeitszüge
- Regional & Nahverkehrszüge
- Metros
- Straßenbahnen
- Monorail
- Lokomotiven
- Reisezugwagen
- Güterwagen
- Off-Train

## SYSTEME FÜR NUTZFAHRZEUGE

- Lkw
- Trailer
- Busse
- Motoren
- Sonderfahrzeuge

# Knorr-Bremse Austria im Überblick

## Knorr-Bremse GmbH

- Vertrieb Systeme für Schienenfahrzeuge
- Magnetschienen-bremsen und Wirbel-strombremsen
- Sandungssysteme
- Scheiben-Wisch-Wasch-Systeme
- Flanschgeräte
- Führerbremsventile Bremsprobegeräte
- RailServices
- Vertrieb Systeme für Nutzfahrzeuge
- Trainings
- TruckServices

## Division IFE

- Türsysteme für Schienenfahrzeuge
- Antriebssysteme
- Türflügel
- Innentüren
- Einstiegshilfen
- Steuergeräte
- RailServices

## Dr. techn. Josef Zelisko GmbH

- Verkehrsmanagement-systeme
- Messwandler
- Signalsysteme

## Skach GmbH

- Handel mit Brems-komponenten und Verschleißteilen für Nutzfahrzeuge

## EKA d.o.o.

- Mobile Testgeräte
- Stationäre Testgeräte
- Zusatzausrüstung

## Kiepe Electric GmbH

- Elektrische Systeme und Integration
- Traktion
- Hilfsbetriebeumrichter
- Klimasysteme

# Entwicklung und Produktion Brems- & On-Board Systeme für Schienenfahrzeuge

## ENTWICKLUNG

### COC Bogie Equipment

- Magnetschienenbremse
- Wirbelstrombremse

### COC Brake Control

- Bremsprobegeräte

### COC Sandung/Wischer

- Sandungssysteme
- Scheiben-Wisch-Wasch-Systeme

### Prüffeld

- Produktprüfung
- Bauteilprüfung

## PRODUKTION

### COC Bogie Equipment

- Magnetschienenbremse
- Wirbelstrombremse

### Mechanische Fertigung

### COC Brake Control

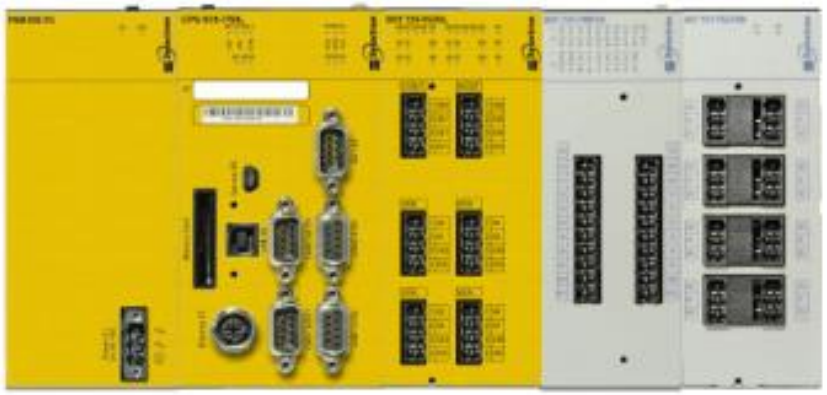
- Bremsprobegeräte
- Flanschbare Tafelgeräte
- Führerbremsventile
- Gleitschutzventile
- Luftfederungsventile
- Bedienungsventile
- Master Controller

### COC Sandung/Wischer

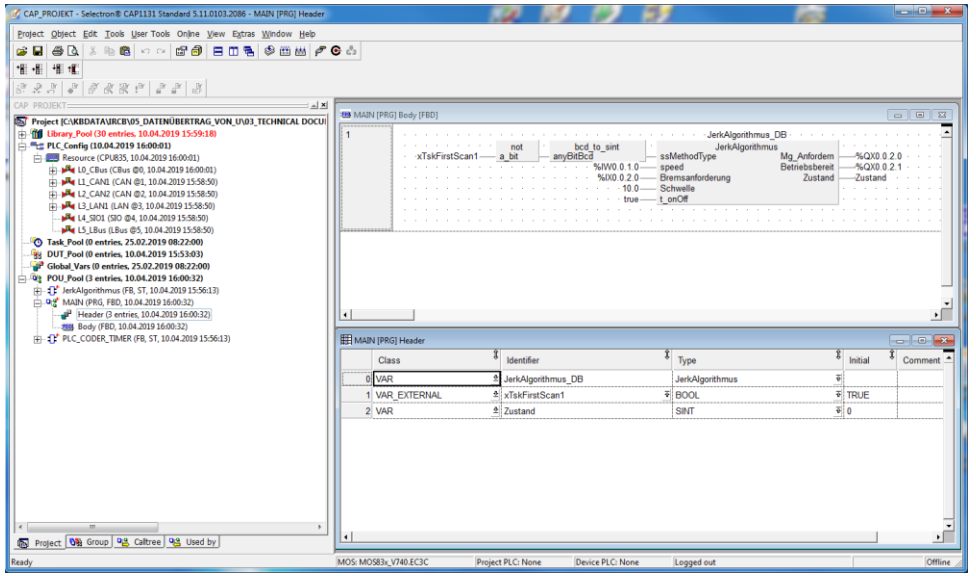
- Sandungssysteme
- Scheiben-Wisch-Wasch-Systeme

# Selectron

- HARDWARE
  - SIL2 Zertifizierte Hardware
    - Prozessormodule (zB.CPU835)
    - Erweiterungsmodule (zB. DDT, AIT)



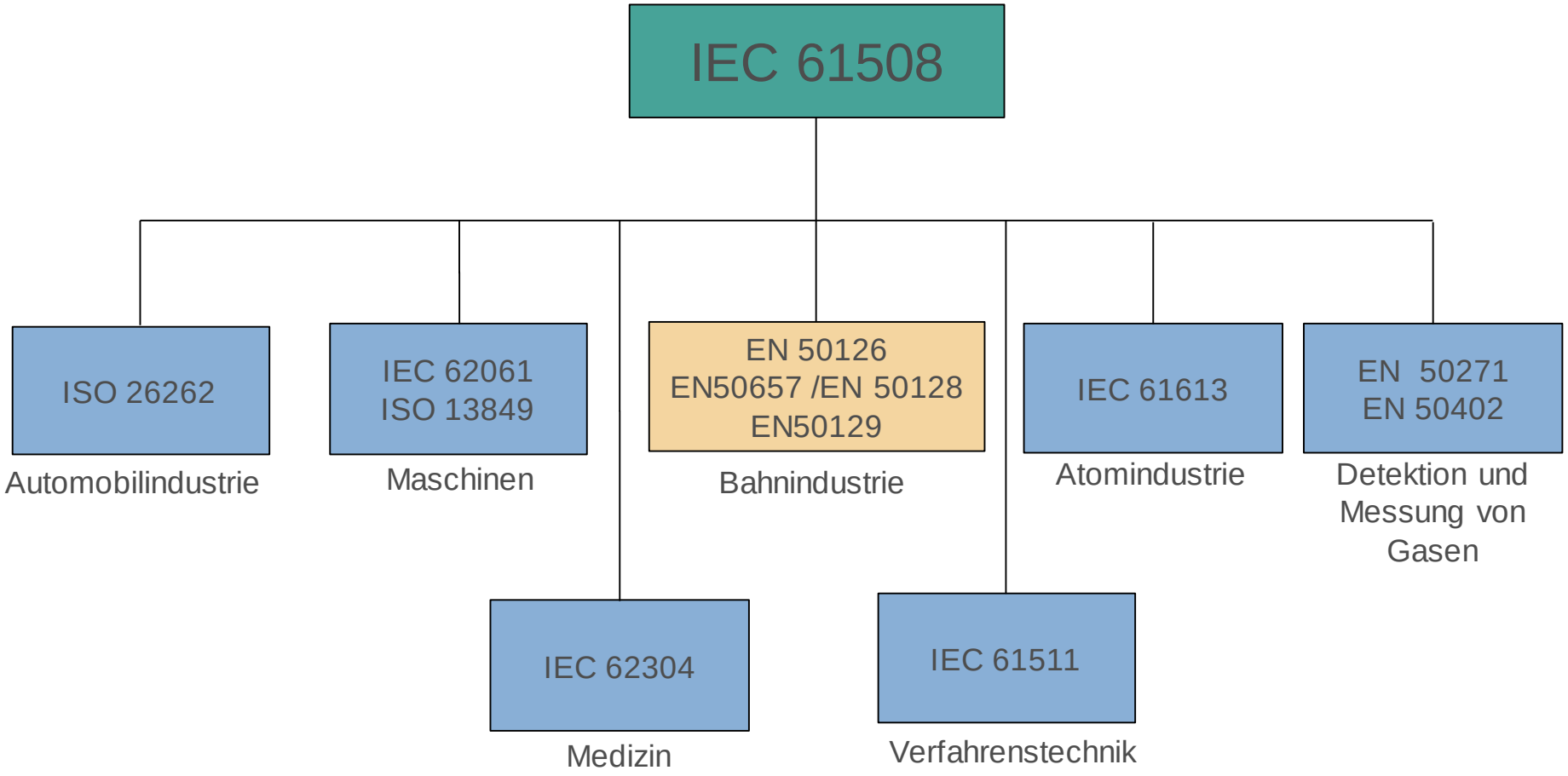
- SOFTWARE
  - Toolkette geeignet für zertifizierte SW Entwicklung bis SIL2
    - Programmierung auf Basis von IEC 61131-3
    - Verwendung von Strukturiertem Text (ST) möglich
  - Entwicklungsumgebung Symphony Suite




Abbildungen Selectron , Screenshoot CAP1131

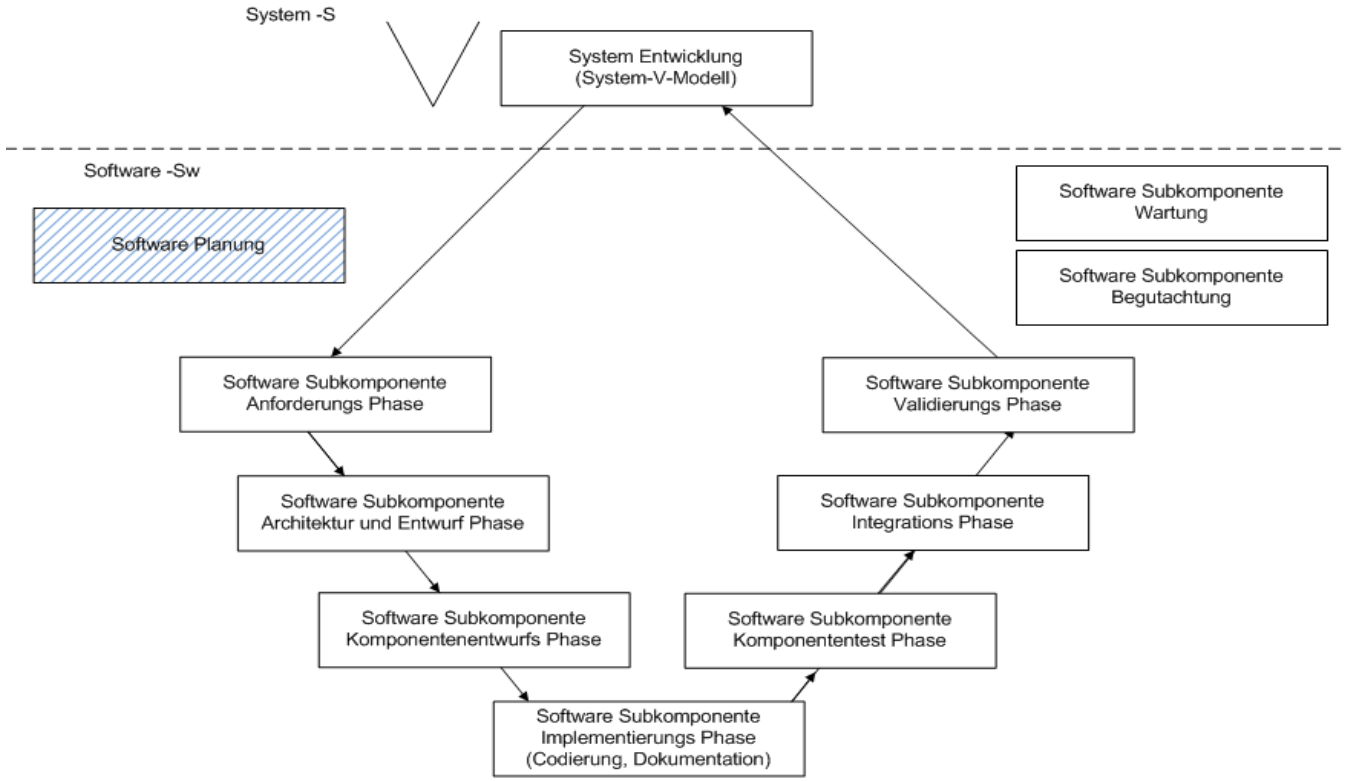
# Softwarentwicklung

# Entwicklungsumfeld - Normative Grundlagen

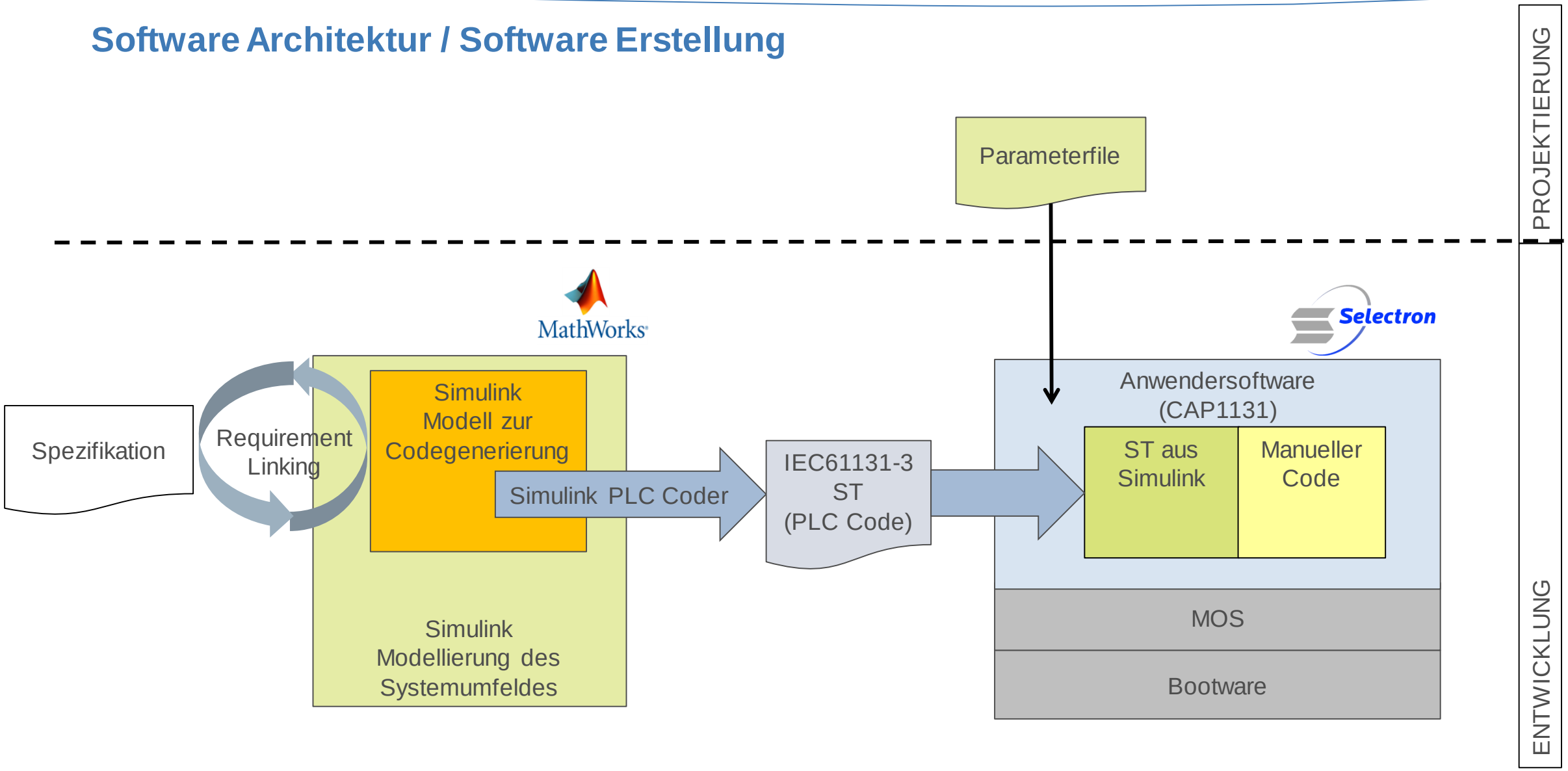


# Entwicklungsumfeld - Normative Grundlagen

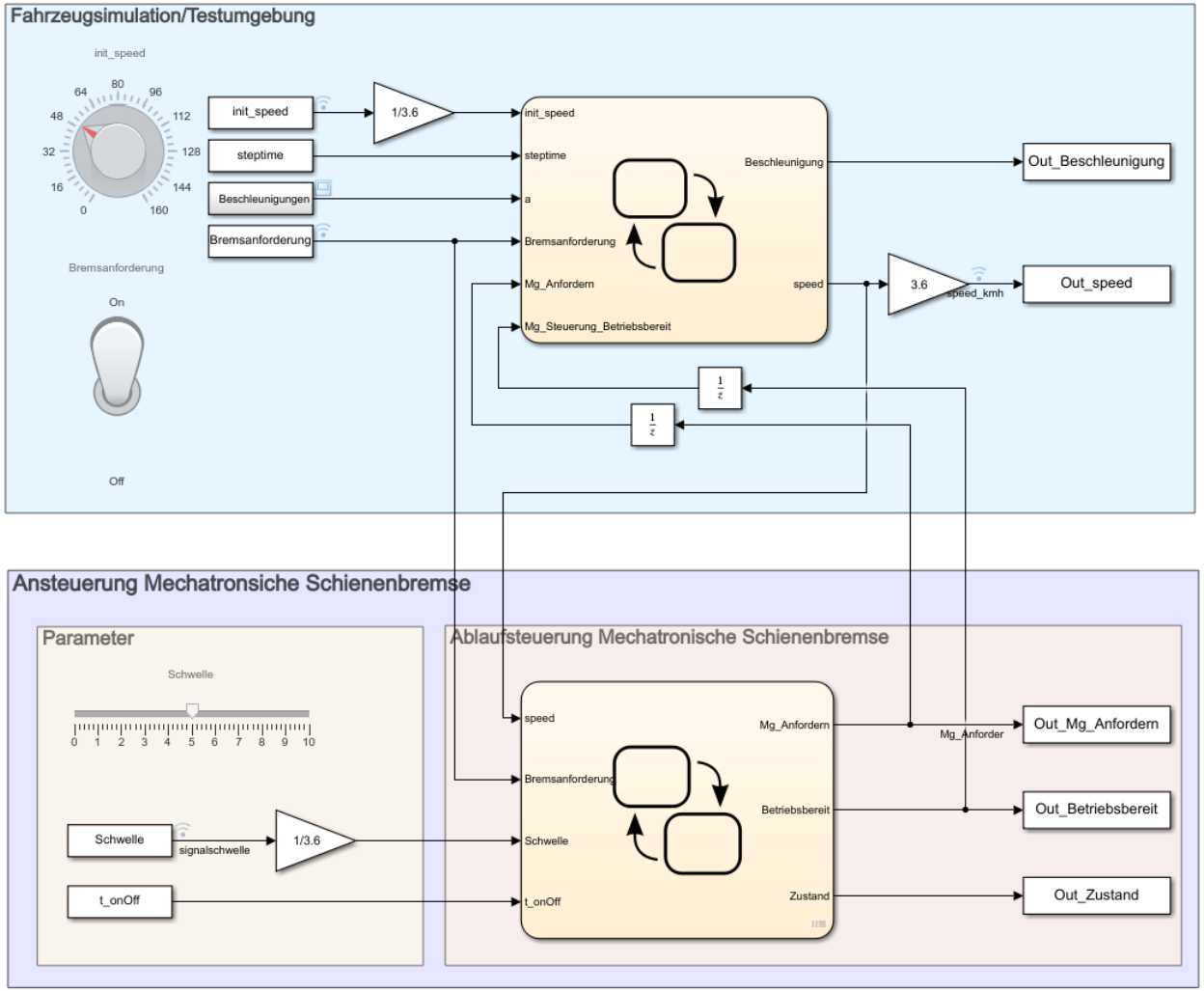
- EN50126 für Sicherheitsbetrachtung und EN50129 für Hardware
- EN50657 (bzw. EN50128) für Software

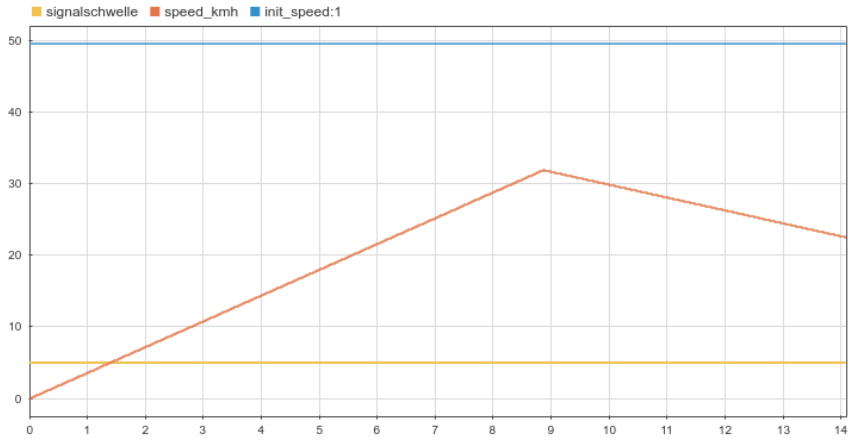


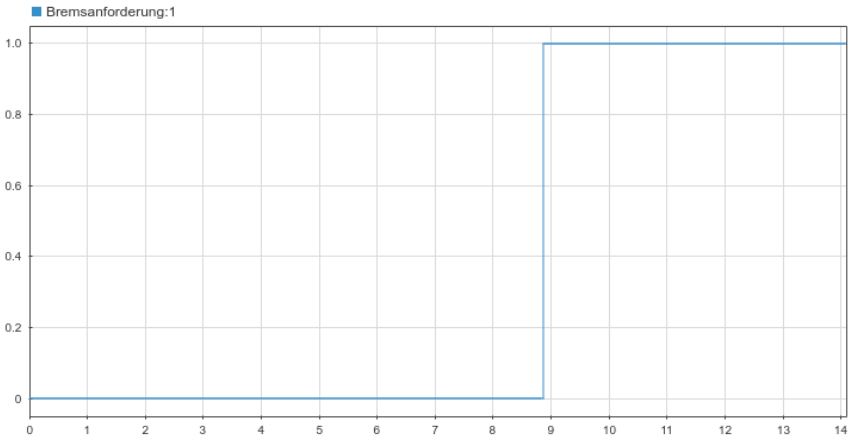
# Software Architektur / Software Erstellung



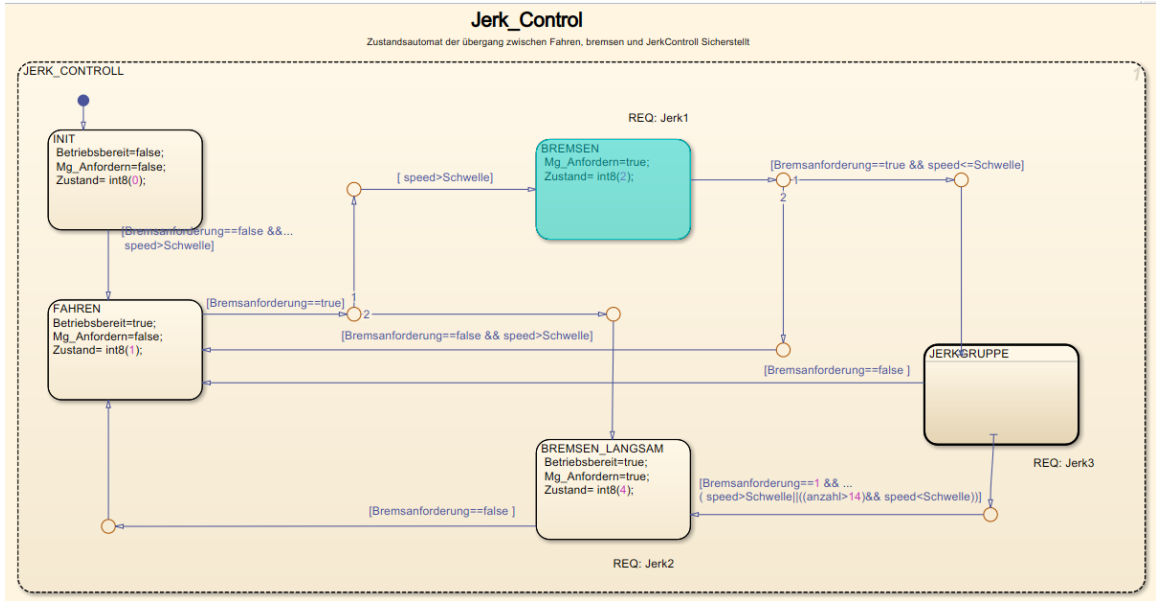
# BSP Jerk Control (Halteruckbegrenzung)



# BSP Jerk Control (Halteruckbegrenzung)




```
85              CASE is_JERKCONTROLL OF
86                  1:
87                      (* During 'BREMSEN': '<S1>:4' *)
88                      (* excluded object 4 for ev 0125 by doebroea on 2(
89                      (* Transition: '<S1>:166' *)
90                      (* '<S1>:8:1' sf_internal_predicateOutput = Bremsa
91                      IF Bremsanforderung AND (speed <= Schwelle) THEN
92                          (* Transition: '<S1>:8' *)
93                          (* Transition: '<S1>:132' *)
94                          is_JERKCONTROLL := 5;
95                          (* Entry 'JERKGRUPPE': '<S1>:66' *)
96                          (* '<S1>:66:1' anzahl= uint8(1); *)
97                          temporalCounter_i1(timerAction := 1, maxTime :
98                          (* Outport: '<Root>/Zustand' *)
99                          (* Entry 'JERK': '<S1>:5' *)
100                         (* '<S1>:5:1' Zustand=int8(3); *)
101                         Zustand := 3;
102                         (* '<S1>:5:1' while  anzahl<=uint8(14) && t_o
103                     ELSE
104                         (* Transition: '<S1>:9' *)
105                         (* '<S1>:144:1' sf_internal_predicateOutput =
106                         IF ( NOT Bremsanforderung) AND (speed > Schwel
107                             (* Transition: '<S1>:144' *)
108                             is_JERKCONTROLL := 3;
109                             (* Entry 'FAHREN': '<S1>:3' *)
110                             (* '<S1>:3:1' Betriebsbereit=true; *)
111                             Betriebsbereit := TRUE;
112                             (* '<S1>:3:1' Mg_Anfordern=false; *)
113                             Mg_Anfordern := FALSE;
114                             (* Outport: '<Root>/Zustand' incorporates
115                              *  Outport: '<Root>/Betriebsbereit'
116                              *  Outport: '<Root>/Mg_Anfordern' *)
117                             (* '<S1>:3:2' Zustand= int8(1); *)
118                             Zustand := 1;
```

- Code Erzeugung funktioniert mit EINEM KLICK!
- Verlinkung zwischen Code und Modell hilft bei Reviews

Screenshoot: Matlab

# PLC Codegenerierung (Strukturierter Text)

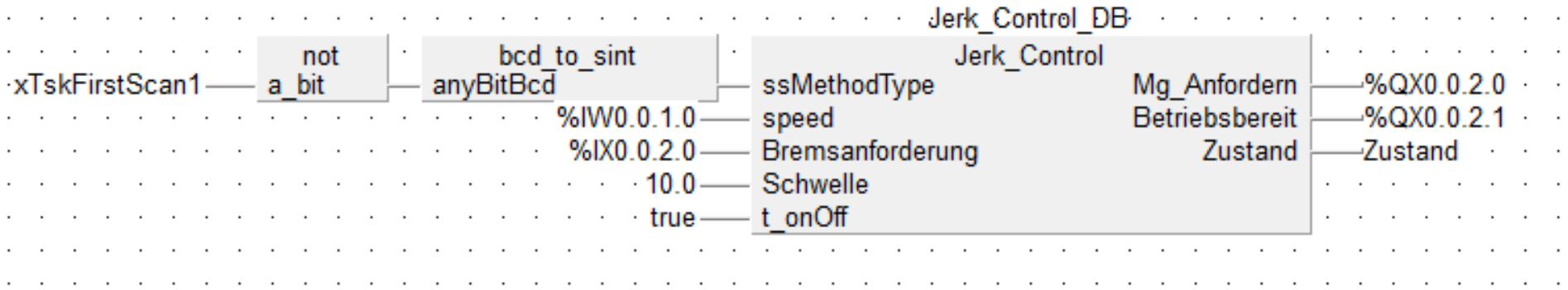
- Bei Modellierung muss Aspekt der Codegenerierung beachtet werden.
  - Kompatible Elemente (sichere Untermenge)
  - Architektur des Modells (Review-Vorlage)
  - Verwendung von Simulink.Parameter
  - Einhaltung der Namenskonventionen
  - Atomares Subsystem
- Es gibt für Selectron eine eigene Target IDE.
  - Testbench Generierung für PIL
  - steht zur Zeit als AddOn zu Verfügung

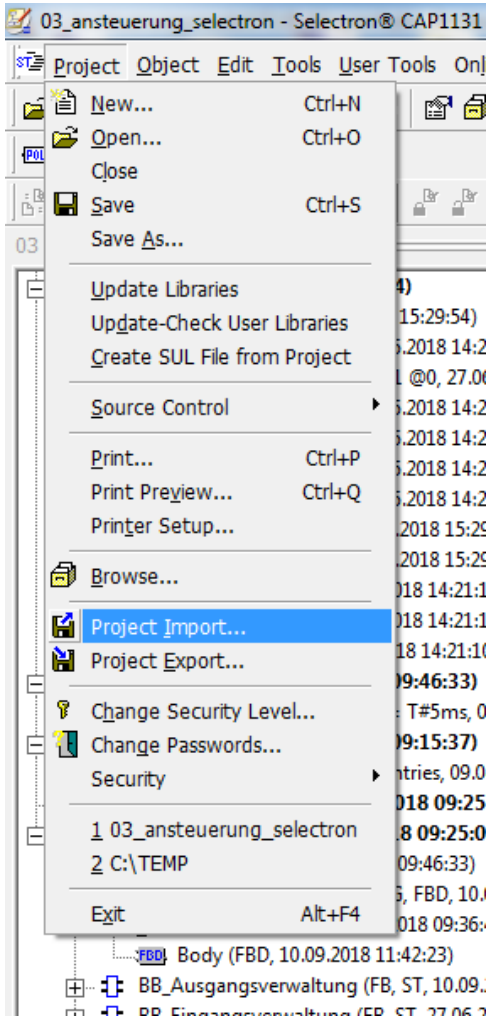
```
85          CASE is_JERKCONTROLL OF
86              1:
87                  (* During 'BREMSEN': '<S1>:4' *)
88                  (* excluded object 4 for ev_0125 by doebroea on 2019-05-20
89                  (* Transition: '<S1>:166' *)
90                  (* '<S1>:8:1' sf_internal_predicateOutput = Bremsanforderur
91                  IF Bremsanforderung AND (speed <= Schwelle) THEN
92                      (* Transition: '<S1>:8' *)
93                      (* Transition: '<S1>:132' *)
94                      is_JERKCONTROLL := 5;
95                      (* Entry 'JERKGRUPPE': '<S1>:66' *)
96                      (* '<S1>:66:1' anzahl= uint8(1); *)
97                      temporalCounter_i1(timerAction := 1, maxTime := 0);
98                      (* Outport: '<Root>/Zustand' *)
99                      (* Entry 'JERK': '<S1>:5' *)
100                     (* '<S1>:5:1' Zustand=int8(3); *)
101                     Zustand := 3;
102                     (* '<S1>:5:1' while  anzahl<=uint8(14) && t_onOff(anzal
103                 ELSE
104                     (* Transition: '<S1>:9' *)
105                     (* '<S1>:144:1' sf_internal_predicateOutput = Bremsanfc
106                     IF ( NOT Bremsanforderung) AND (speed > Schwelle) THEN
107                         (* Transition: '<S1>:144' *)
108                         is_JERKCONTROLL := 3;
109                         (* Entry 'FAHREN': '<S1>:3' *)
110                         (* '<S1>:3:1' Betriebsbereit=true; *)
```


Screenshoot: Matlab

# Generierten Strukturierten Text in die Entwicklungsumgebung einbinden

- Importieren der erstellten *.asc Datei über CAP1131->Project->Import Project



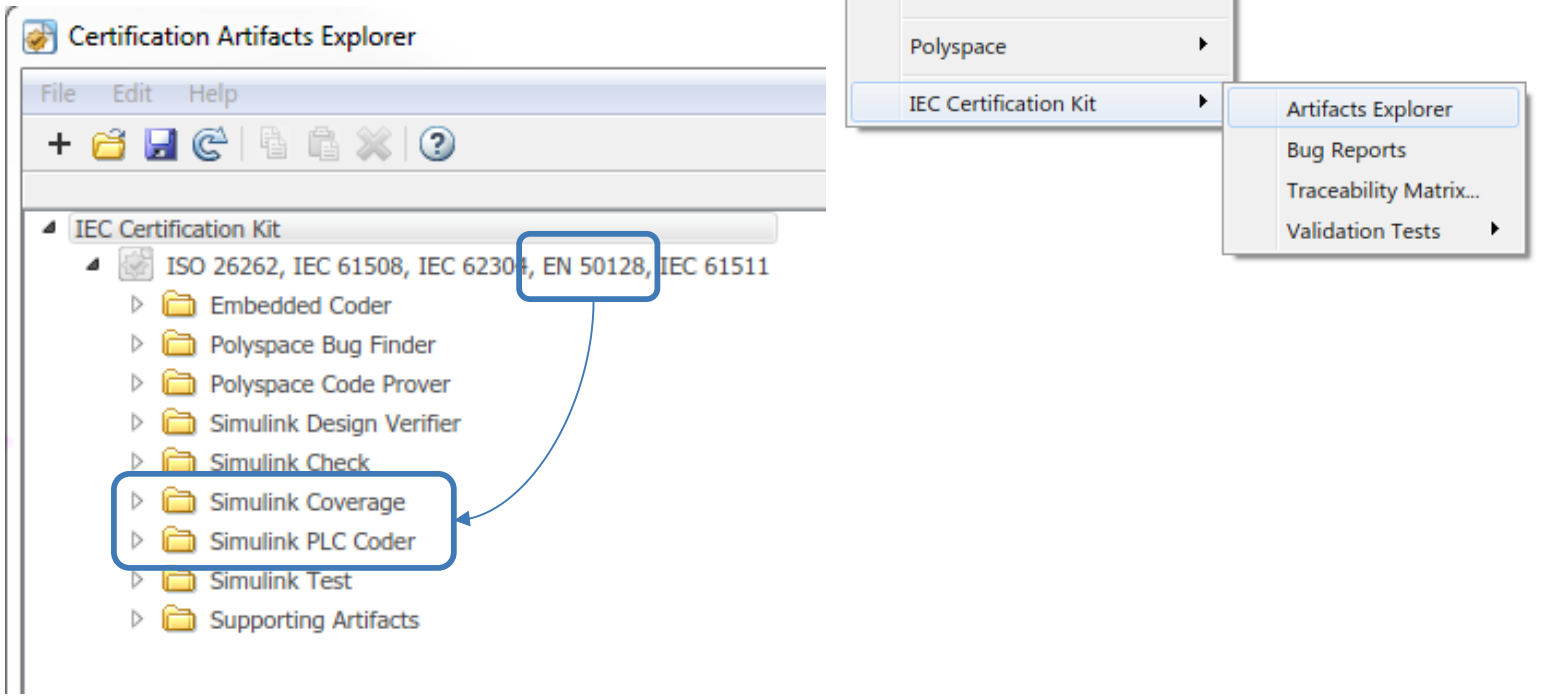

Screenshoot: Selectron CAP1131

# Verifikation und Validierung nach EN50657 (bzw. EN50128)

- Tool Qualification ist gefordert: Auswahl von Werkzeugen in den Klassen T2 und T3 ist zu begründen

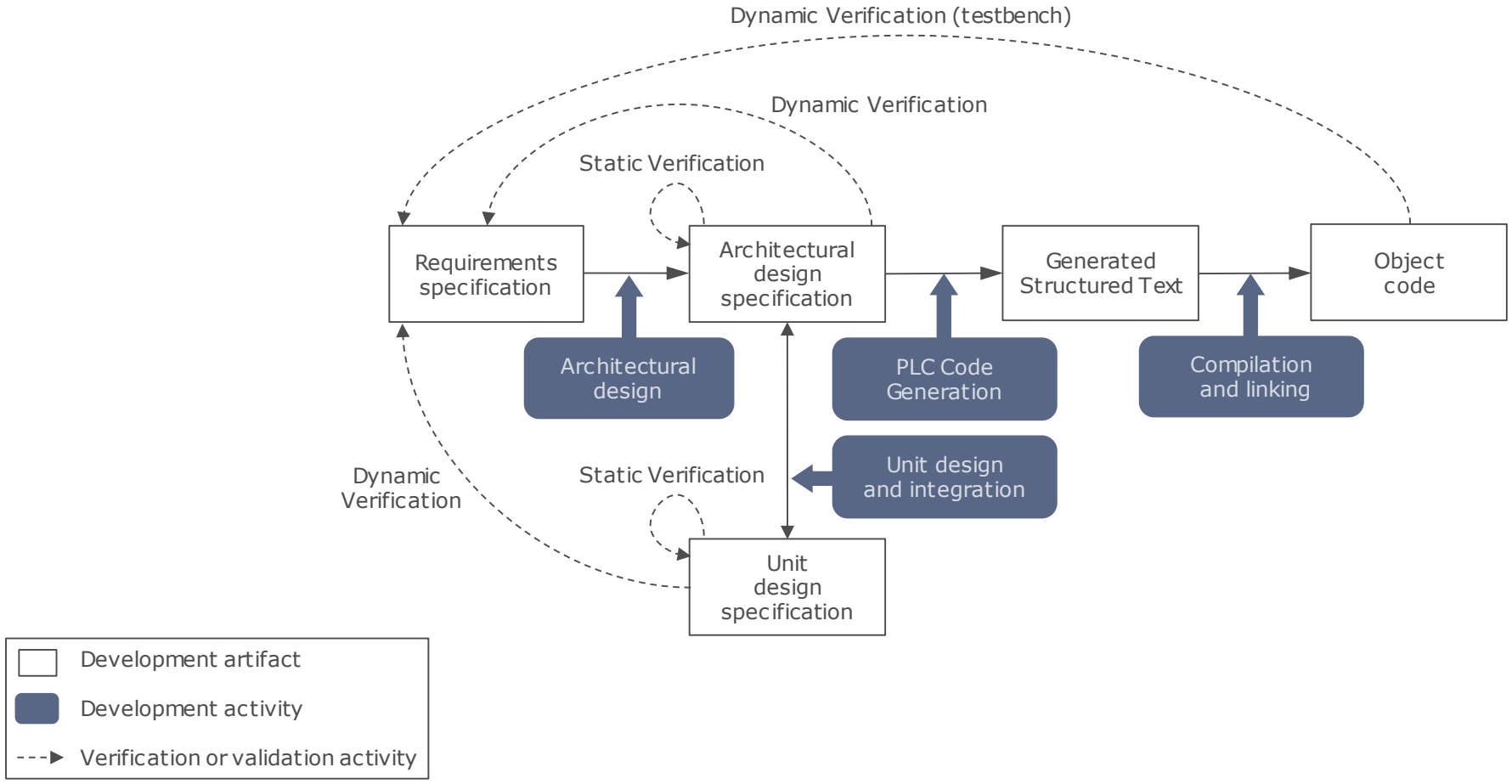
- IEC Certification Kit
  - Model-Based Design for EN 50128 (Normentabelle)
  - User‘s Guideline
  - Release Notes
  - Reference Workflow
  - Software Tool Inventory

- Für PLC-Coder & Coverage:
  - User's Guide
  - Release Notes
  - Conformance Demonstration Template
  - Certificate
  - Report to the Certificate
  - Tool Qualification Package
  - Reference Workflow




Zusammengefasst aus Certification Kit‘s der verwendeten Tools

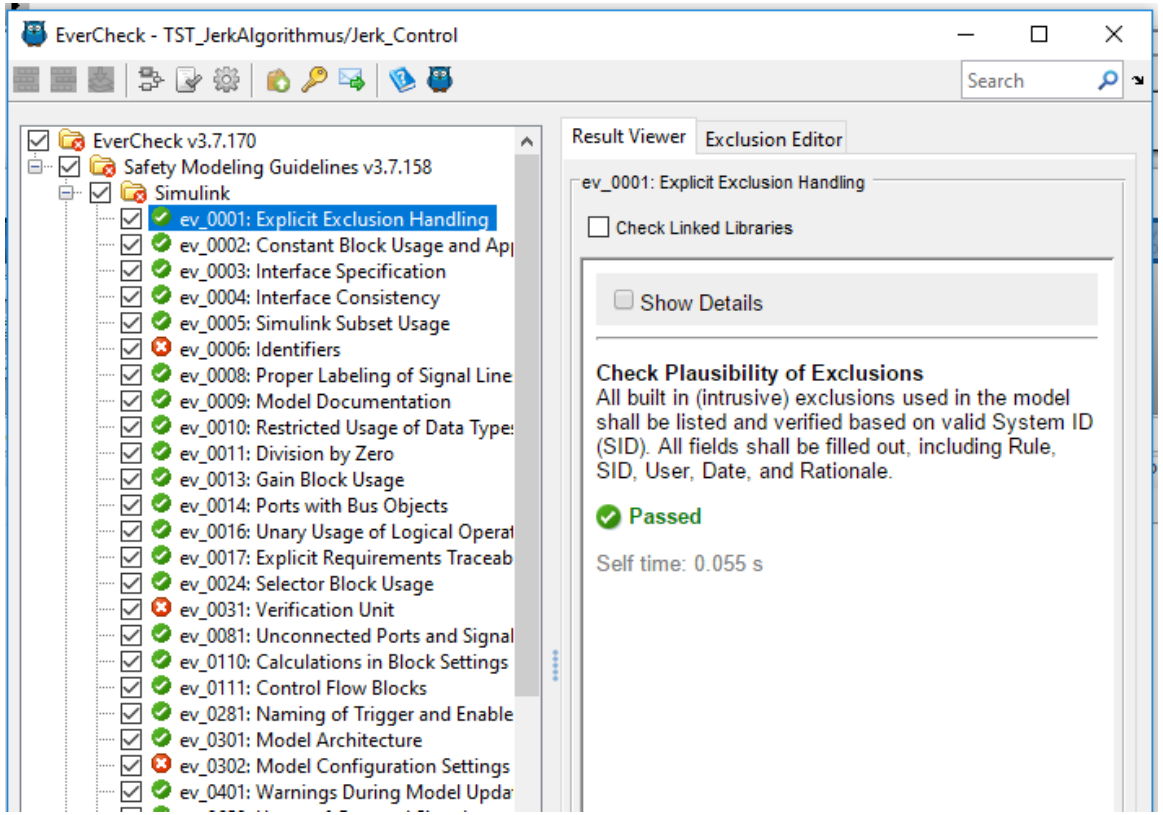
# Verifikation und Validierung nach EN50657 (bzw. EN50128)




EverTool Dokumentation

# Statische Verifikation der Modelle

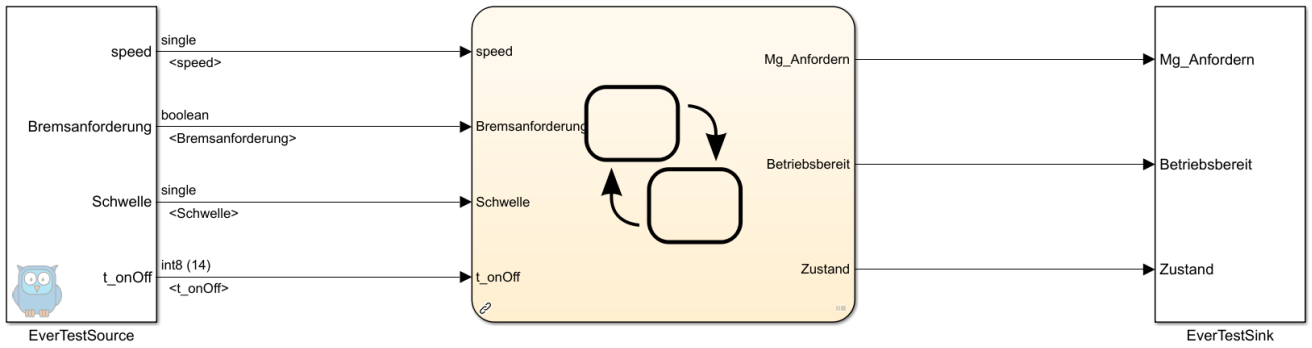
- Formale Überprüfung, dass Modelle nach Modellierungsrichtlinien entwickelt wurden
- Dokumentieren und Begründen von Abweichungen zu Modellierungsrichtlinien
- Automatisierte Modellüberprüfung von:
  - Verwendung einer definierte Untermenge von Funktionen
  - Prüfen, dass Modell geringe Komplexität aufweist
  - Strenge Definition von Schnittstellen (Datentypen, Ranges)
  - Einstellungen in MATLAB/Simulink
  - Defensive Programmierung
  - Graphische Darstellung
  - Erfassen von Modell Metriken




Screenshoot: EverCheck

# Dynamische Verifikation der Modelle

- Funktionale Überprüfung des Simulink Modells gegen die Anforderungen
- Erstellen der Testfälle und ausführen in der Simulationsumgebung
  - Test Driven Development, Debugging im Modell
  - Basis für Prozessor in the Loop (PIL) Tests
  - Generierung der Verifikationsberichte
- Auf Komponenten und Integrationsebene

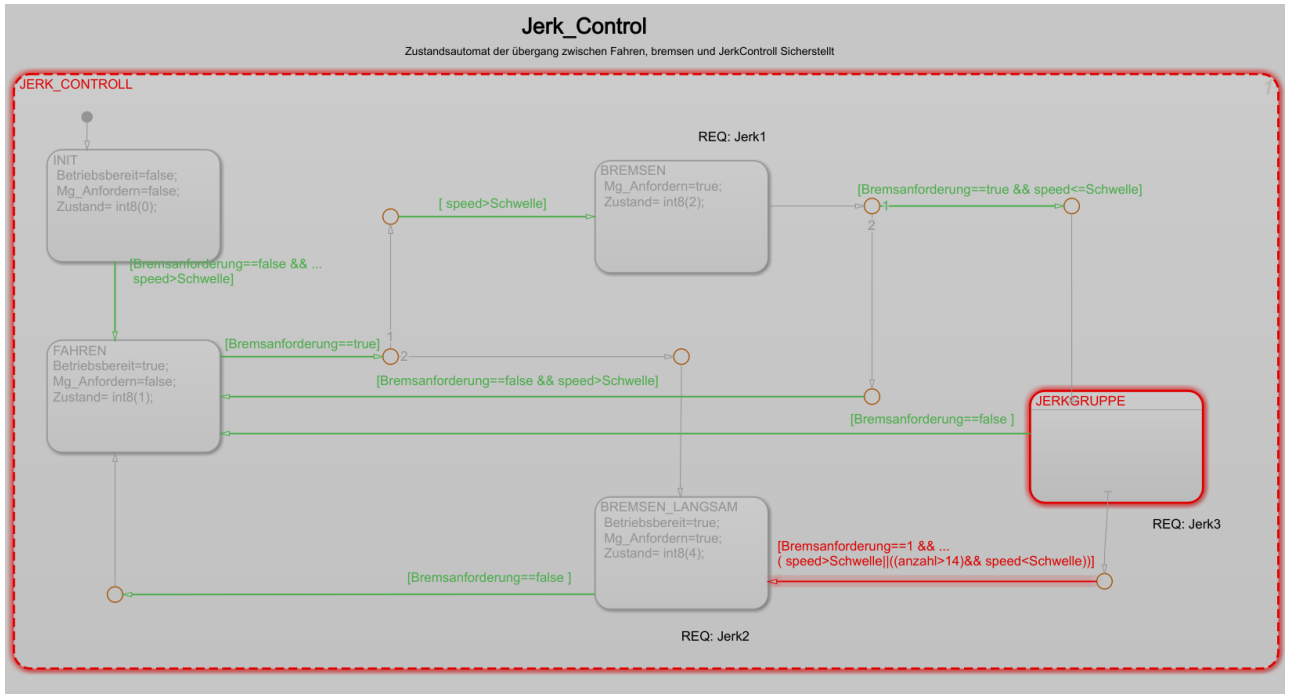



| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Description | | speed | Bremsanforderung | Schwelle | t_onOff | Mg_Anfordern_ref | Mg_Anfordern | Betriebsbereit_ref | Betriebsbereit | Zustand_ref | Zustand |
| 2 | | 0 | 0 | 0 | 10 | [50 30 40 50 30 80 10 20 30 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | | 0.005 | 20 | 0 | | | 0 | 0 | 1 | 1 | 1 | 1 |
| 4 | REQ: Jerk1 | 0.01 | 80 | | | | 0 | 0 | 1 | 1 | | 1 |
| 5 | | 2 | | 1 | | | 1 | 1 | 1 | 1 | 2 | 2 |
| 6 | | 3 | 5 | | | | 1 | 1 | 1 | 1 | 3 | 3 |
| 7 | REF: Jerk3 | 4 | | 0 | | | 0 | 0 | 1 | 1 | 1 | 1 |
| 8 | | 5 | | | | | 0 | 0 | 1 | 1 | | 1 |
| 9 | | 9 | 5 | 1 | | | 1 | 1 | 1 | 1 | 4 | 4 |
| 10 | | | | | | | | | | | | |

Screenshoot: Simulink / EverTest

# Testabdeckung - Model Coverage

- Prüfen, ob ausreichend Testfälle angewendet wurden
- Ergebnisse mehrerer Tests können zusammengefasst werden (Cumulative Coverage)
- Unterstützung geforderter Coverage Algorithmen:
  - Decision Converage
  - Modified Condition Decision Cover Age (MCDC)
- Grafische Darstellung der Testabdeckung
- Report über die Testabdeckung




Screenshoot: Matlab

# PIL (Processor in the Loop) – Äquivalenz-Test

- PLC Coder führt alle Testfälle aus und generiert PLC-Code für Modell + Testfälle
- PLC Code des Modells zusammen mit PLC Code der Testfälle werden auf Target ausgeführt
- Nachweis der Äquivalenz zwischen Modell-Simulation und PLC Code auf Target
- Prozess-Absicherung der Software-Entwicklung
- Ausführung auch auf SIM PLC möglich

# Zusammenfassung

- Klarer Entwicklungsprozess
- Hohe Testbarkeit ohne Einsatz von Hardware
- Funktionsentwicklung unabhängig von verwendeter Hardware möglich
- Einsatz von Messdaten zur Absicherung der entwickelten Funktionen
- Möglichkeit der System- bzw. Funktionssimulation
- Test können in Simulink durchgeführt werden und auf die HW übertragen werden (Testbench)
- Hohe Entwicklungsgeschwindigkeit / niedriger Personalaufwand

# VIELEN DANK FÜR IHRE AUFMERKSAMKEIT

# IHRE FRAGEN !!!

# Kontakt

Knorr-Bremse AG
Fr. Döbrössy Angelika
Beethovengasse 43-45
A-2340 Mödling

Tel.: +43 2236 409 2364
E-Mail: angelika.doebroessy@knorr-bremse.com
www.knorr-bremse.com