

Software Engineering für Software im Automobil.

Entwicklungswerkzeuge.

BMW Group



Entwicklungswerkzeuge.

Vortragsgliederung.

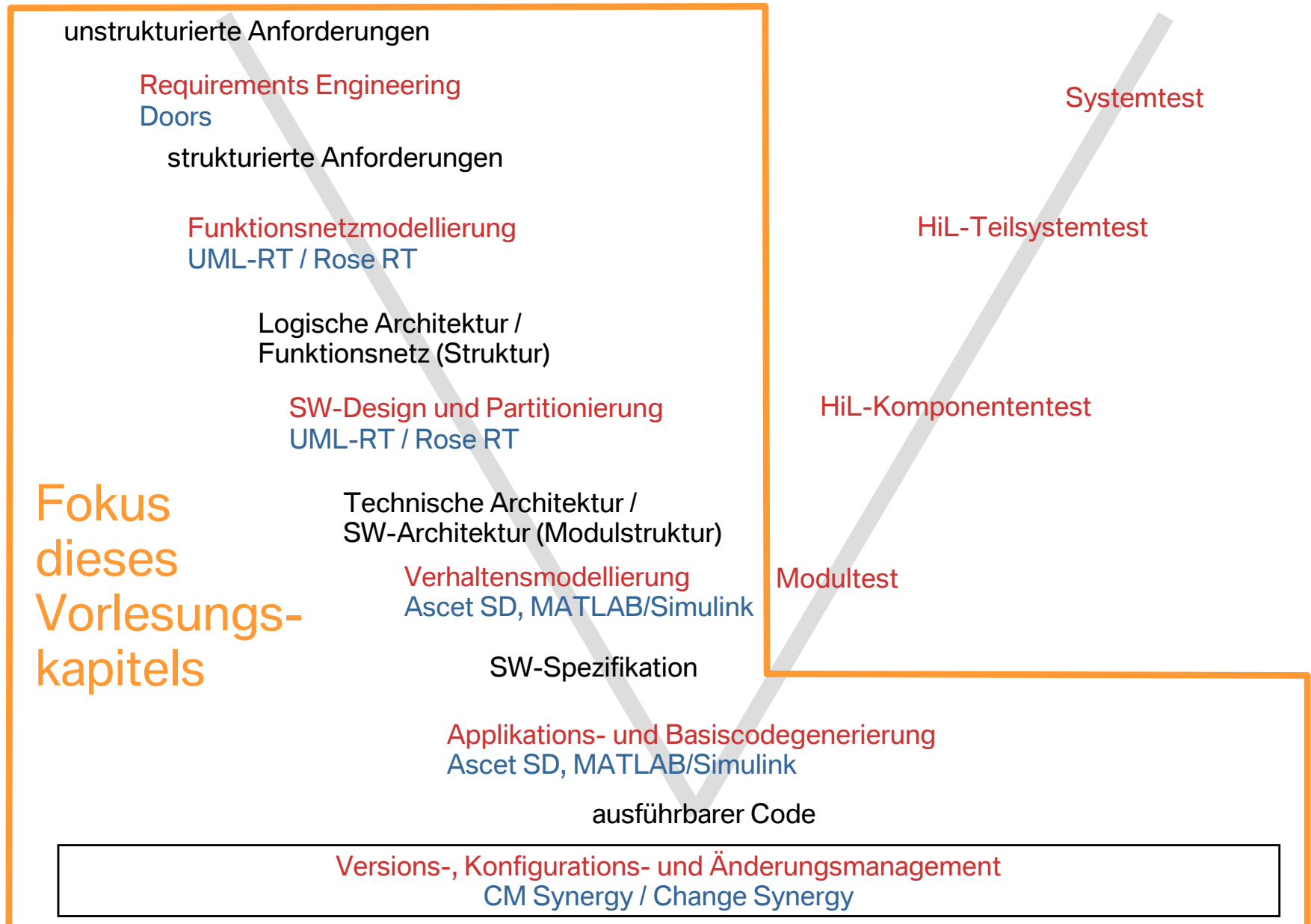
- Anforderungen an Methoden und Entwicklungswerkzeuge
- Eine Methoden- und Werkzeugkette für einen Entwicklungsprozess
- Präsentation einzelner Entwicklungswerkzeuge:
 - Requirements Engineering
 - DOORS
 - Funktionsnetzmodellierung
 - Rose RT
 - Verhaltensspezifikation:
 - MATLAB/Simulink
 - ASCET-SD
 - Versions-, Konfigurations- und Änderungsmanagement:
 - CM Synergy
 - Change Synergy
- Zusammenfassung

Anforderungen an Methoden und Entwicklungswerkzeuge.

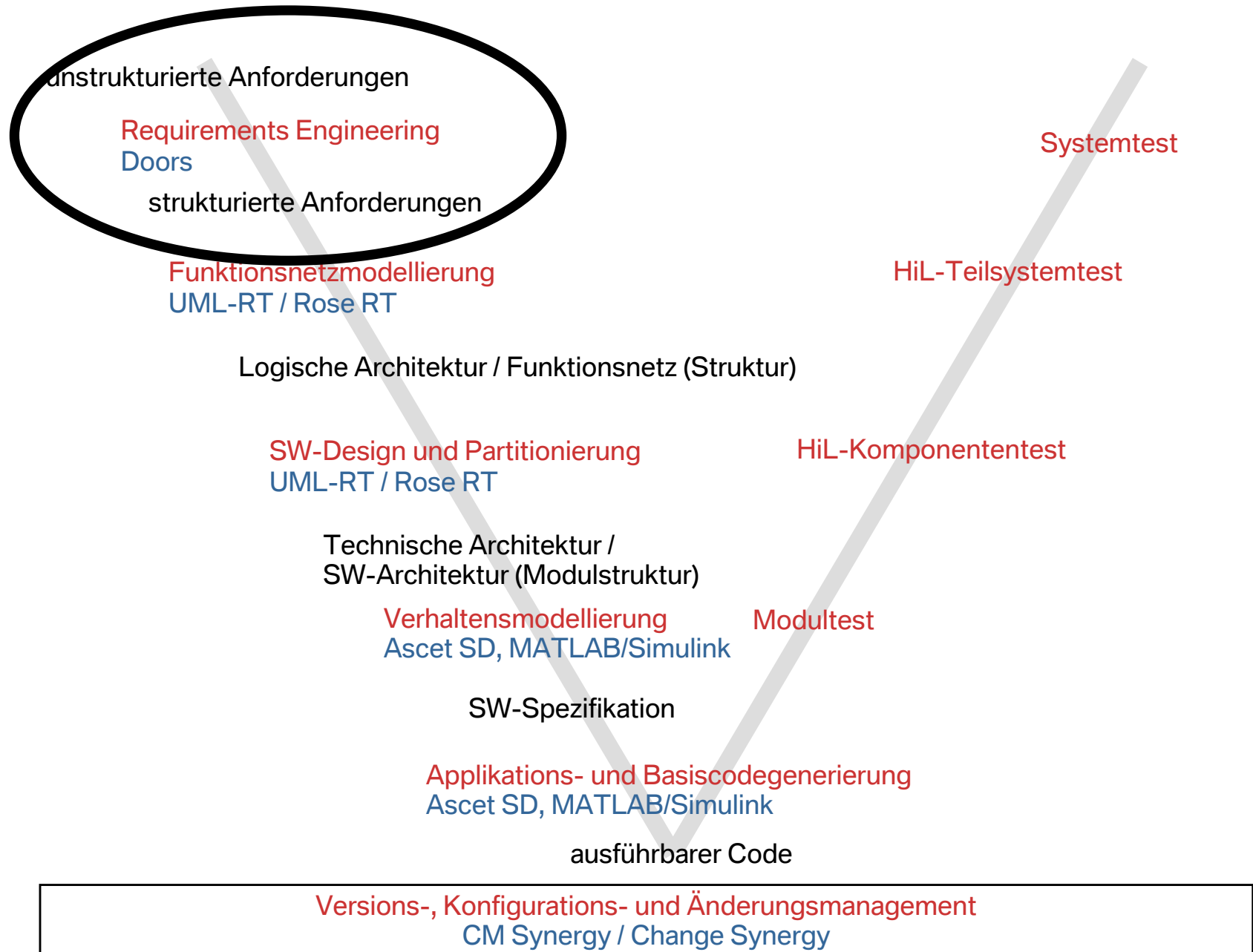
- Unterstützung bei Entwicklung, Konsistenzhaltung und Analysen
- Entlastung des Benutzers von Routine-Tätigkeiten
- Verwendung von Standards (Notationen, Schnittstellen)
- Modellbasierung
- Skalierbarkeit
- Durchgängigkeit zwischen „aufeinander folgenden“ Methoden bzw. Werkzeugen

Eine Methoden- und Toolkette für einen Entwicklungsprozess.

Prozessschritte, -resultate und Tools.



Präsentation einzelner Entwicklungswerkzeuge. Requirements Engineering mit DOORS.



Requirements Engineering mit DOORS.

Hersteller: Telelogic

Features: Single Source Datenbank
Multiuserfähig
Versionierung, History-Verwaltung
Linkkonzept
Attributierung
Sichten, Filter, Sortierungen
Programmierschnittstelle DXL
Web Interface: DOORSNet

Requirements Engineering mit DOORS. Strukturierung von Anforderungsdokumenten.

Projekt

Folder

Formales
Modul

Linkmodul

The screenshot displays the BMW Requirements Database (DOORS) interface. The main window is titled "BMW Requirements Database: /AGS - DOORS". It features a tree view on the left and a table view on the right. The tree view shows a hierarchy of folders and modules, with arrows pointing to specific elements labeled "Projekt", "Folder", "Formales Modul", and "Linkmodul". The table view shows a list of requirements with columns for "Name", "Type", "Description", and "Deleted?". A smaller window titled "Formales Modul 'AGS/AAC Umsetzungsnahe Doku' current 1.0 (initial) - DOORS" is open, showing a detailed view of a requirement with columns for "ID", "beinhaltet PDF-File", and "Aktion".

Name	Type	Description	Deleted?
EA-71 Inoffizieller Bereich	Folder	Für Versuche und Evaluierungen, kei...	
Prüfspezifikation	Folder	Testfälle, Testdaten, Prüfpläne	
SW-Änderungsanträge	Folder	Verwaltung der SW-Änderungsanträg...	
Word-Importe	Folder	Do Not Edit	
AAA_INHALTSVERZEICHNIS_AGS	Formal	Übersicht der verwendeten Module, V...	
AAB_REALISIERUNG_AGS	Formal	Zusammenfassung von Realisierung ...	
AAC_Umsetzungsnahe Doku	Formal	beinhaltet PDF-File	
ABA_Schnittstellen und Zugriffsdaten	Formal	SW-Schnittstellendokument; Gangsc...	
INFORMATIONSAUSTAUSCH	Formal		
Messgrößen	Formal	Mess-und Einstellgrößen, Variable un...	(Deleted)
Realisierung Schnittstellen	Link	von Realisierung nach Schnittstellen ...	
Schnittstellen und Zugriffsdaten_int...	Link	verbindet Objekte innerhalb des Moduls	
Schnittstellen_Umsetzungsnahe Doku	Link	von Schnittstellen_Zugriffsdaten nach...	
Umsetzungs-Beziehung	Link	von Realisierung zu Umsetzungsnahe...	
Umsetzungsnahe_Doku_intern	Link	verbindet Objekte innerhalb des Moduls	
von Inhaltsverzeichnis zu Module	Link	vom Inhaltsverzeichnis zu den einzeln...	
zu Historie	Link	zur Historie (im Urdokument Schnittste...	

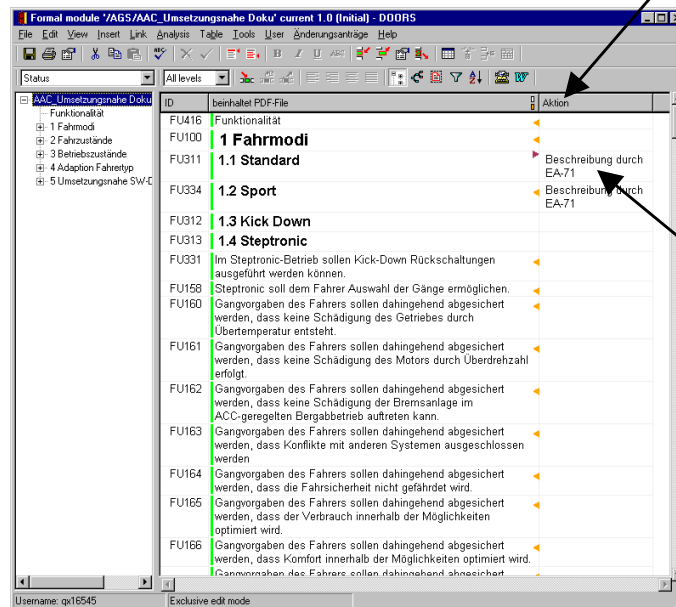
ID	beinhaltet PDF-File	Aktion
FU416		IFunktionalität
FU100		1 Fahrmodi
FU311		1.1 Standard
FU334		1.2 Sport
FU312		1.3 Kick Down
FU313		1.4 Steptronic
FU331		Im Steptronic-Betrieb sollen Kick-Down Rückschaltungen ausgeführt werden können.
FU158		Steptronic soll dem Fahrer Auswahl der Gänge ermöglichen.
FU160		Gangvorgaben des Fahrers sollen dahingehend abgesichert werden, dass keine Schädigung des Getriebes durch Überdrehzahl entsteht.
FU161		Gangvorgaben des Fahrers sollen dahingehend abgesichert werden, dass keine Schädigung des Motors durch Überdrehzahl erfolgt.
FU162		Gangvorgaben des Fahrers sollen dahingehend abgesichert werden, dass keine Schädigung der Bremsanlage im ACC-geregelten Bergabtrieb auftreten kann.
FU163		Gangvorgaben des Fahrers sollen dahingehend abgesichert werden, dass Konflikte mit anderen Systemen ausgeschlossen werden.
FU164		Gangvorgaben des Fahrers sollen dahingehend abgesichert werden, dass die Fahrsicherheit nicht gefährdet wird.
FU165		Gangvorgaben des Fahrers sollen dahingehend abgesichert werden, dass der Verbrauch innerhalb der Möglichkeiten optimiert wird.
FU166		Gangvorgaben des Fahrers sollen dahingehend abgesichert werden, dass Komfort innerhalb der Möglichkeiten optimiert wird.
FU167		Gangvorgaben des Fahrers sollen dahingehend abgesichert werden, dass die Fahrsicherheit nicht gefährdet wird.

Link

Requirements Engineering mit DOORS.

Attributierung von Anforderungen.

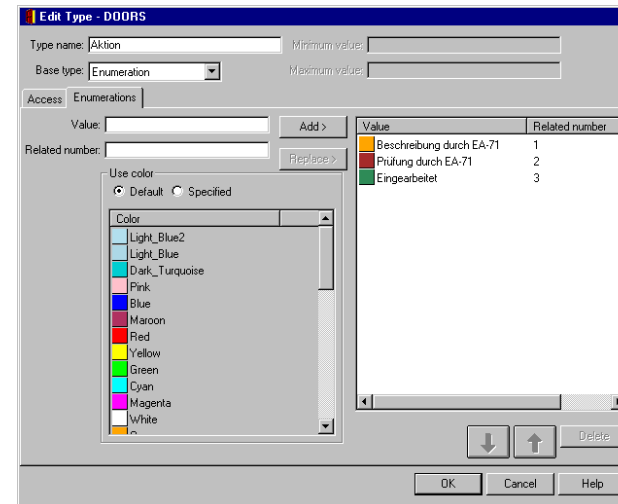
Attributierung über
 Enumerationstypen möglich



Attribut

Attributwert

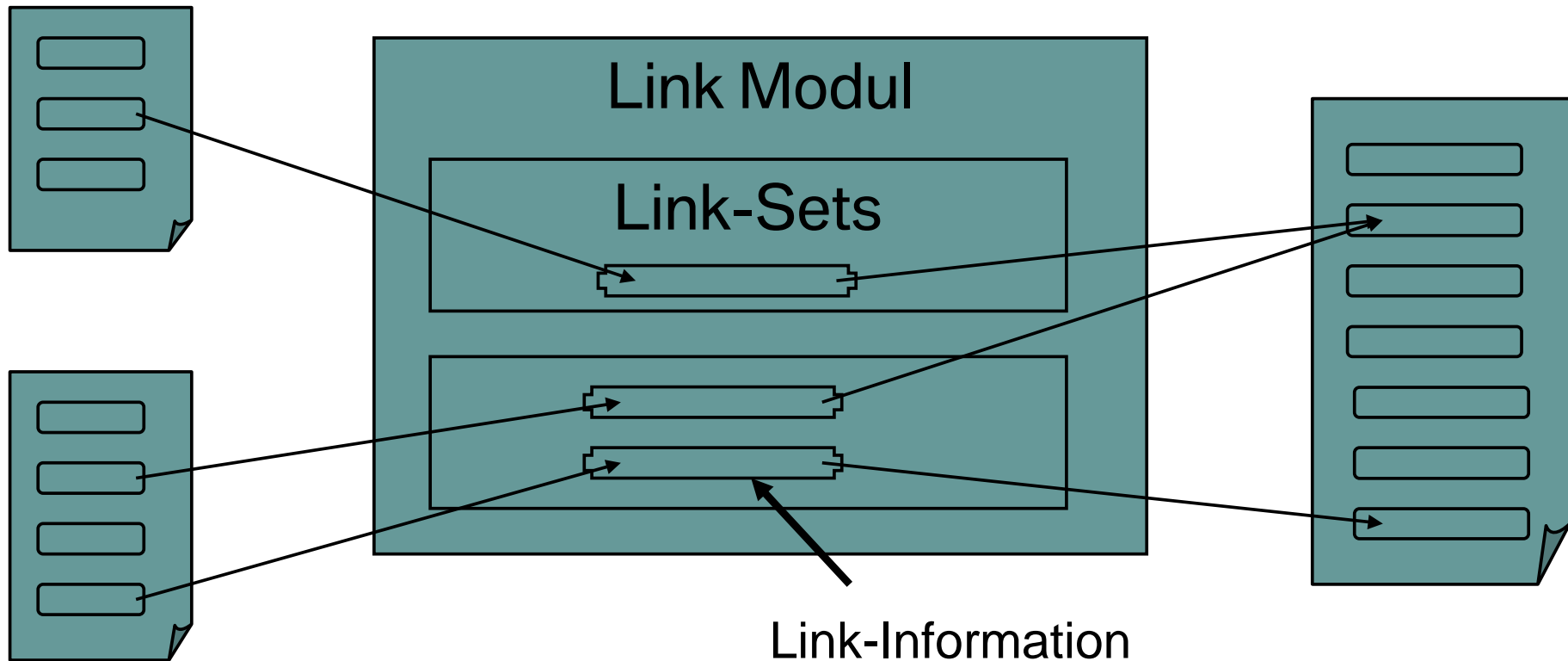
Name	Type	Default value	Inherent value	Exists for	Multi valued
Absolute Number	Integer		No	Object	No
Aktion	Aktion		No	Object	No
Created By	String		No	Module & Object	No
Created On	Date		No	Module & Object	No
Created Thru	Created Thru	Manual Input	No	Object	No
Description	String		No	Module	No
Font Table	Text		No	Module	No
Last Modified By	String		No	Module & Object	No
Last Modified On	Date		No	Module & Object	No
Name	String		No	Module	No
Object Heading	String		No	Object	No
Object Short Text	String		No	Object	No
Object Text	Text		No	Object	No
Paragraph Style	String		No	Object	No
Prefix	String		No	Module	No



Requirements Engineering mit DOORS. Verlinkung von Anforderungen.

Formales Modul

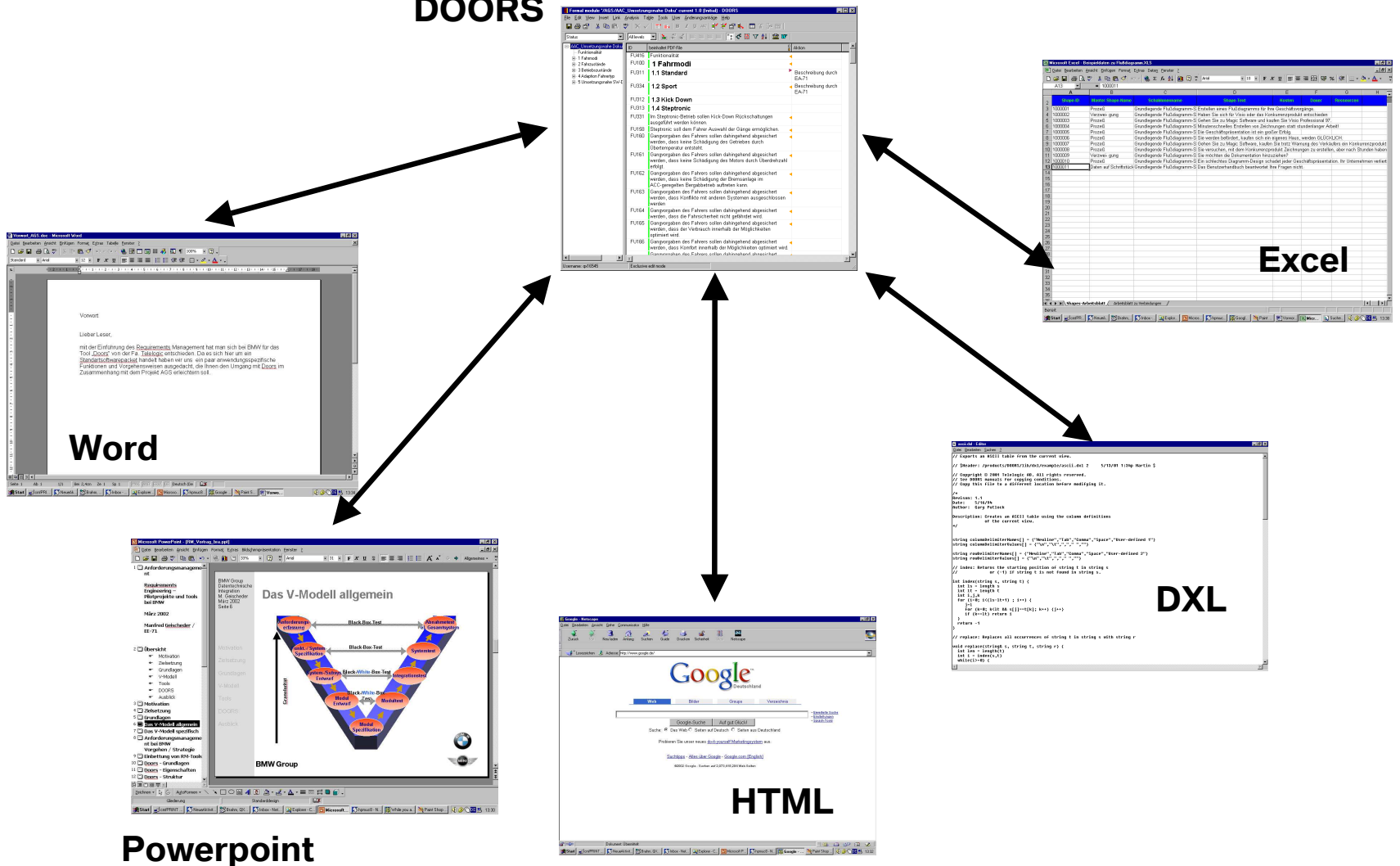
Formales Modul



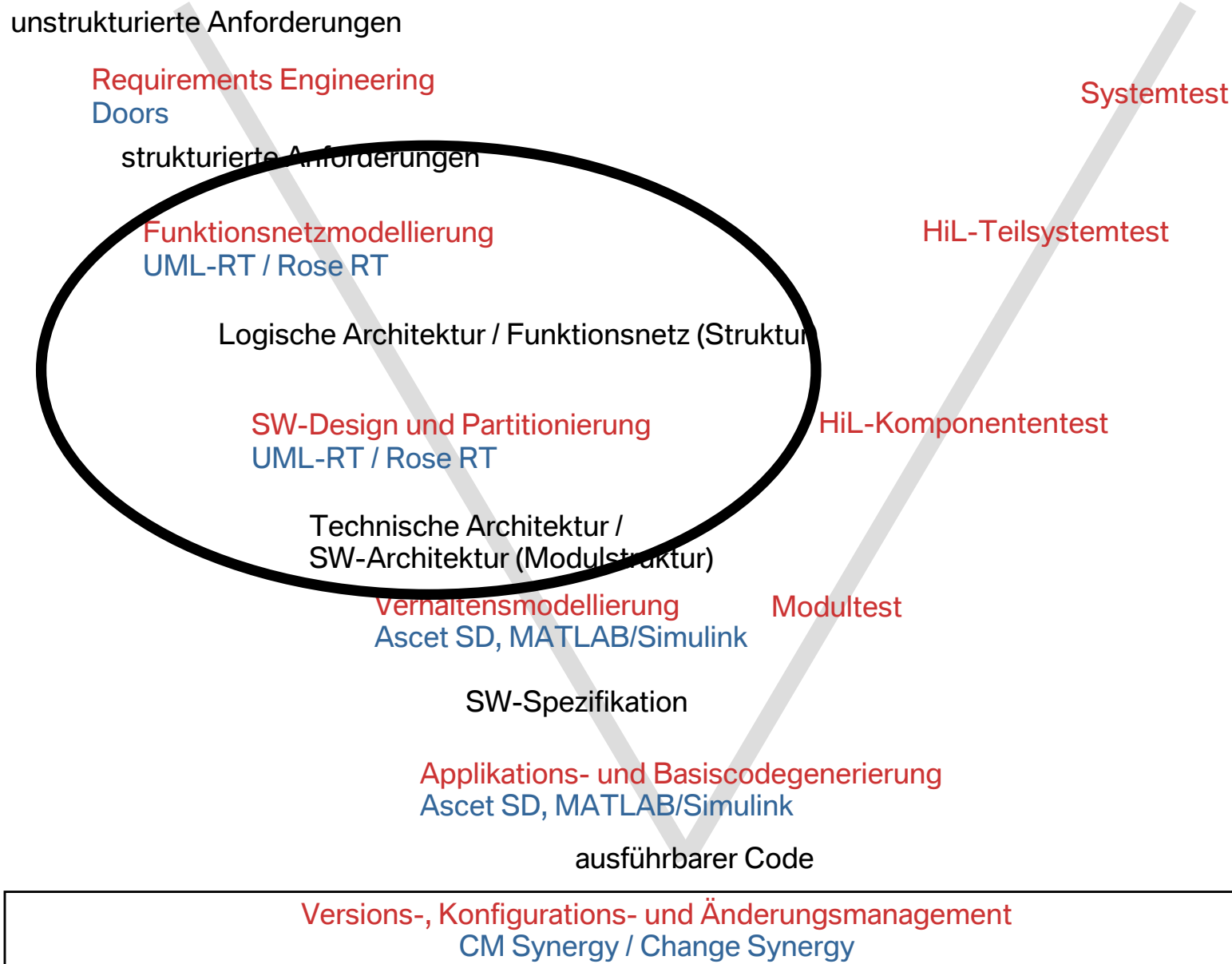
Formales Modul

Requirements Engineering mit DOORS. Schnittstellen zur Außenwelt.

DOORS



Präsentation einzelner Entwicklungswerkzeuge. Funktionsnetzmodellierung mit Rose RT.



Funktionsnetzmodellierung.

Motivation.

Herausforderungen bei der Entwicklung eingebetteter Systeme in der Automobiltechnik:

- Schnell wachsende Funktionsanzahl
- Steigende Komplexität der Funktionen
- Stark steigende Zahl von Interaktionen zwischen Funktionen
- Verlagerung von HW- zur SW-Realisierung
- Sicherheitskritische Anwendungen
- Wiederverwendbarkeit

=> **Hohe Komplexität in der Systementwicklung!**

Funktionsnetzmodellierung. Konsequenzen.

- Komplexitätsbeherrschung:
Zur Beherrschung der Komplexität ist eine übersichtliche graphische Systemsicht auf die Funktionsgesamtheit erforderlich.
- Konkrete Aufgabenstellung:
Entwicklung einer Methodik zur Erstellung einer Systemsicht „Funktionsnetz“:

Funktionsnetzmodellierung

- Intuitive Verständlichkeit
- Abstraktion (Verzicht auf unnötige Details)
- Hierarchische Struktur
- Präzise Notation

Funktionsnetzmodellierung.

Anforderungen an Methodik und Tools.

- Graphische Darstellung von Funktionen, SW-Komponenten, HW-Komponenten und Partitionierungsinformation
- Präzise Definition von Schnittstellen
- Hierarchische Darstellung von Funktionen
- Klare Trennung zwischen logischer u. technischer Architektur
- Darstellbarkeit von Typ- und Instanzinformation
(mit sauberer Trennung)

Funktionsnetzmodellierung.

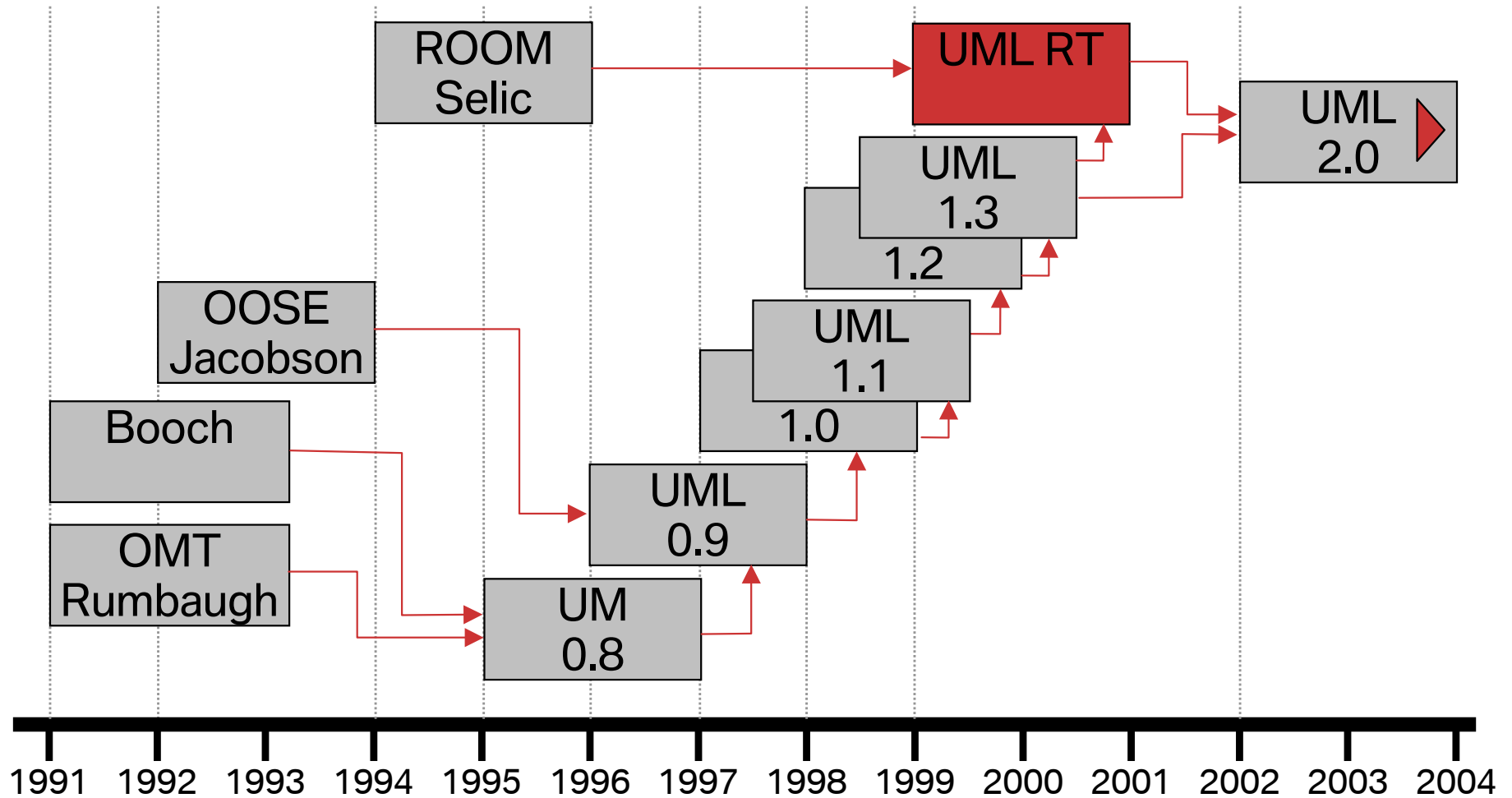
Wahl der Modellierungsnotation (1).

UML

- Umfangreiche Menge objekt-orientierter graphischer Notationen für Analyse und Design von SW-Systemen
- Standardisiert durch OMG
- Quasi-Industrie-Standard
- Umfassende Tool-Unterstützung
- Kein spezieller Anwendungsbereich

Funktionsnetzmodellierung.

Die Historie der UML.



Funktionsnetzmodellierung.

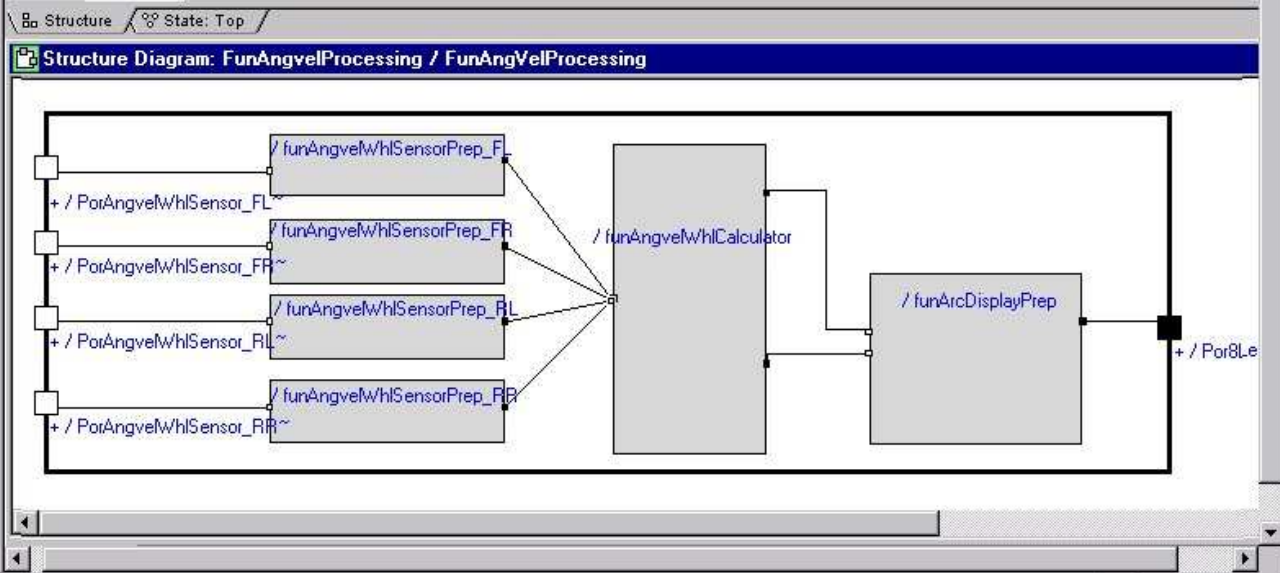
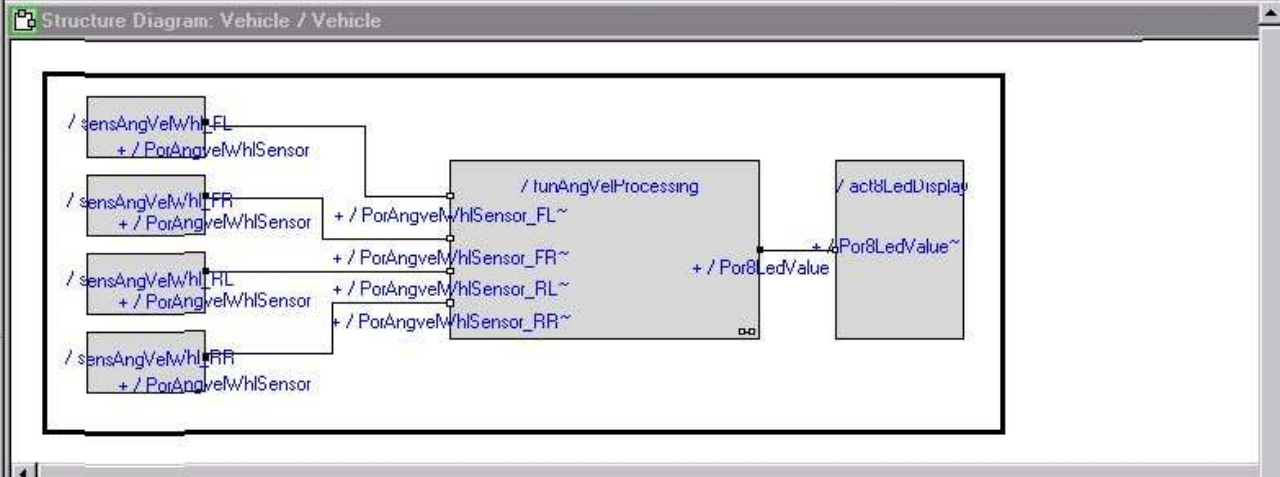
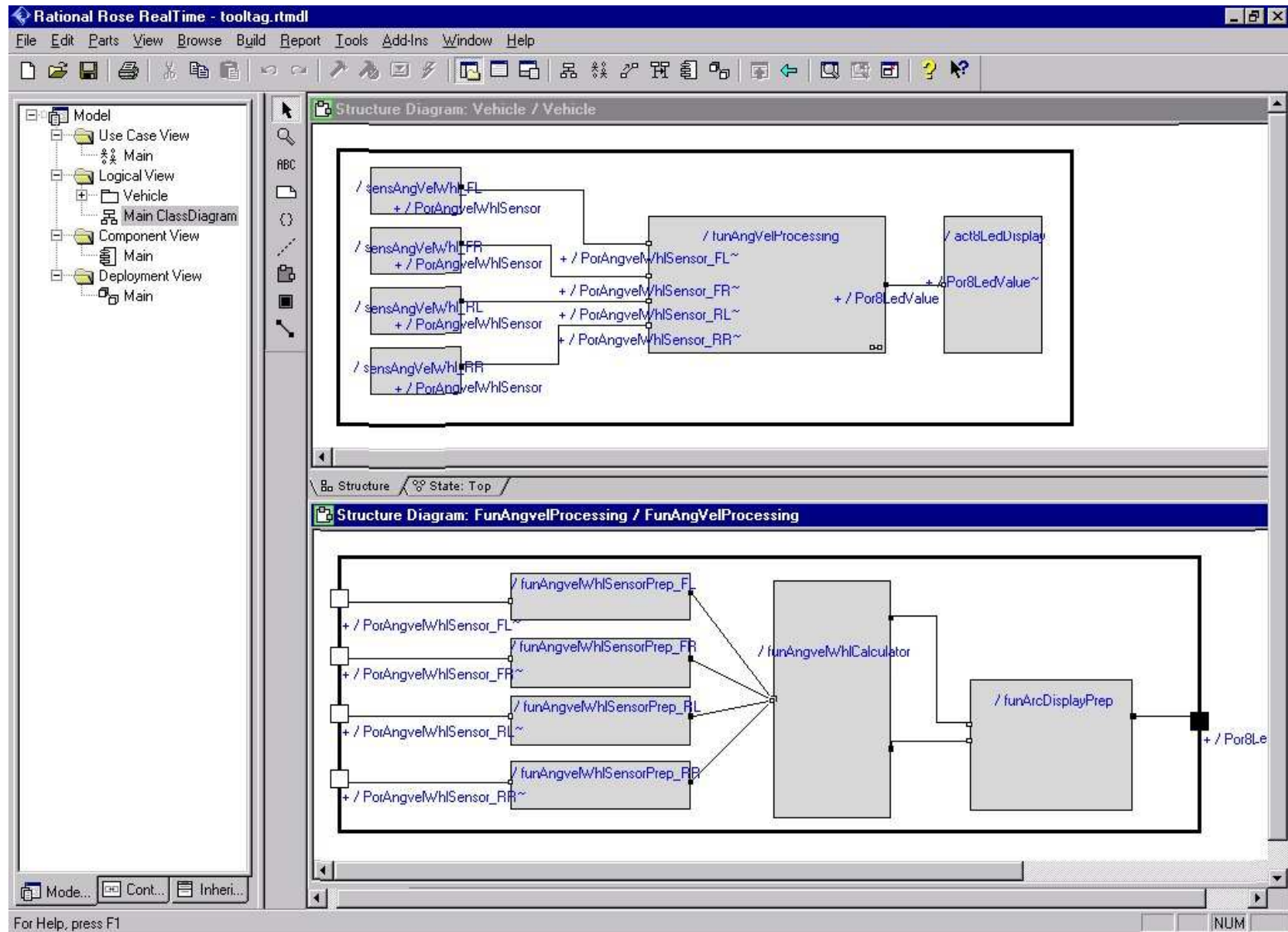
Wahl der Modellierungsnotation (2).

UML-RT

- Spezialisierung der UML für eingebettete Systeme
- Übersichtliche Darstellung von Struktur und Verhalten
(**hier**: Beschränkung auf Struktur!)
- Unterstützt komponentenorientierte Entwicklung
- Möglichkeit zur Hierarchiebildung (Capsules enthalten Capsules)
- Unterscheidung zwischen „provided“ und „required“ Schnittstellen
- Werkzeugunterstützung durch Rational Rose RT

⇒ UML-RT für Funktionsnetzmodellierung gewählt!

Funktionsnetzmodellierung. Ein Funktionsnetz-Beispiel.



Funktionsnetzmodellierung.

Logische und technische Architektur.

Logische Architektur

- Modellierung mit Hilfe von Funktionsnetzen
- keine lösungsorientierte Modellierung: insbesondere keine Entscheidung für Realisierung in SW oder in HW

Technische Architektur

- Modellierung der Funktionen aus logischer Architektur in SW oder in HW (d.h. lösungsorientiert)

Funktionsnetzmodellierung.

UML-RT-Sprachkonstrukte für die logische Architektur.

- Capsules
 - modellieren Funktionen
 - besitzen präzise definierte Schnittstellen
 - Kommunikation mit der Umgebung erfolgt
 - signalbasiert
 - nur über Schnittstellen und Kanäle
 - hierarchische Verfeinerung durch ein System miteinander kommunizierender Capsules
- Ports und Protokolle (von Capsules)
 - modellieren Funktionsschnittstellen
- Konnektoren (zwischen Ports)
 - modellieren Kanäle zwischen Funktionsschnittstellen

Funktionsnetzmodellierung.

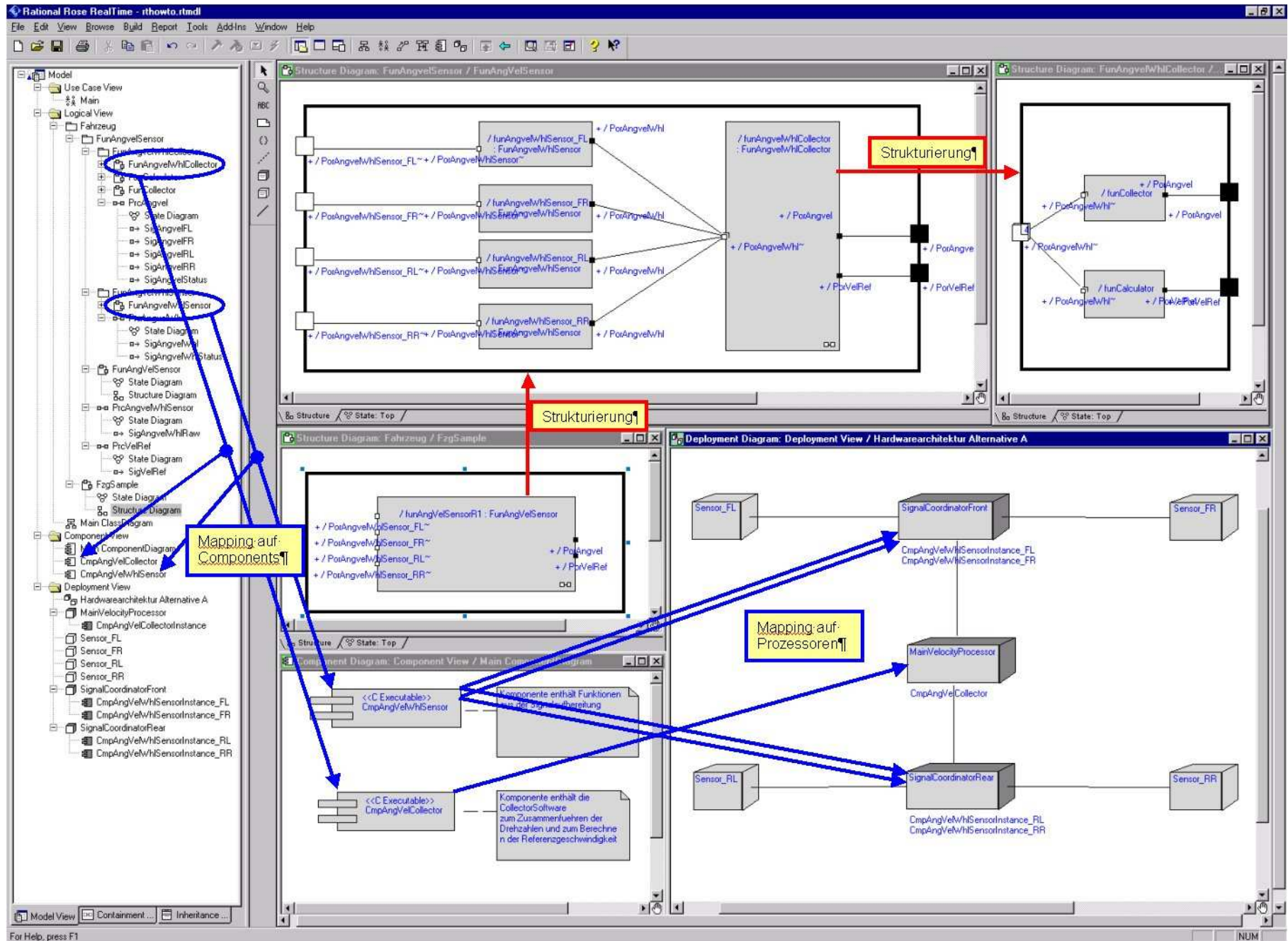
UML-RT-Sprachkonstrukte für die technische Architektur.

- Components
 - modellieren SW-Komponenten

- Nodes
 - modellieren HW-Komponenten
 - Steuergeräte
 - Sensoren
 - Aktuatoren
 - Gateways

Funktionsnetzmodellierung.

Beispiel: Logische und technische Architektur.

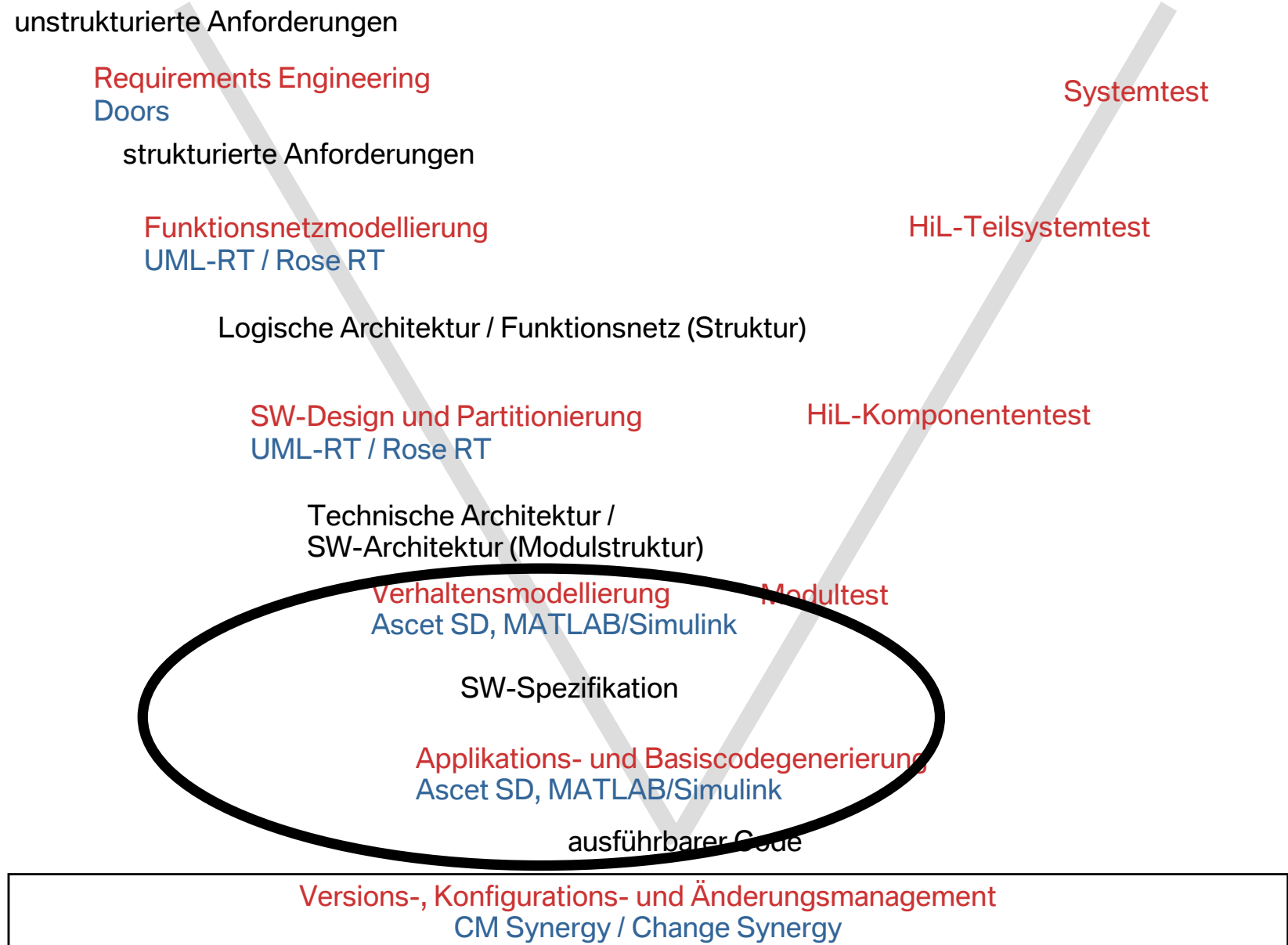


Funktionsnetzmodellierung.

Zusammenfassung.

- Funktionsnetzmodellierung:
Einsatz der UML-RT zur übersichtlichen graphischen
Strukturbeschreibung komplexer Funktionsnetze
- Enge Einbettung in den Entwicklungsprozess:
 - Anforderungserstellung
 - Verhaltensmodellierung und Codegenerierung
 - Versions- und Konfigurationsmanagement
- Umfassende Werkzeugunterstützung

Präsentation einzelner Entwicklungswerkzeuge. Verhaltensspezifikation mit MATLAB/Simulink.



Toolbasierte Entwicklung von Steuerungen und Regelungssystemen.

Motivation.

Beschreibung dynamischer Systeme

- Regelsysteme im Fahrzeug
- zugehörige Streckenmodelle
mit Hilfe mathematischer Ausdrücke



Wichtige Anwendungsdomänen im Automotive-Bereich:
Fahrwerk und Antrieb.

Verwendung signalflussbasierter Blockdiagramme zur Erstellung modularer grafischen Abbildes solcher mathematischer Modelle.

- Hilfsmittel bei Analyse und Stabilisierung dieser Systeme
- Beitrag zur Komplexitätsbeherrschung
- Verkürzung der Entwicklungszeiten und Kostensenkung.

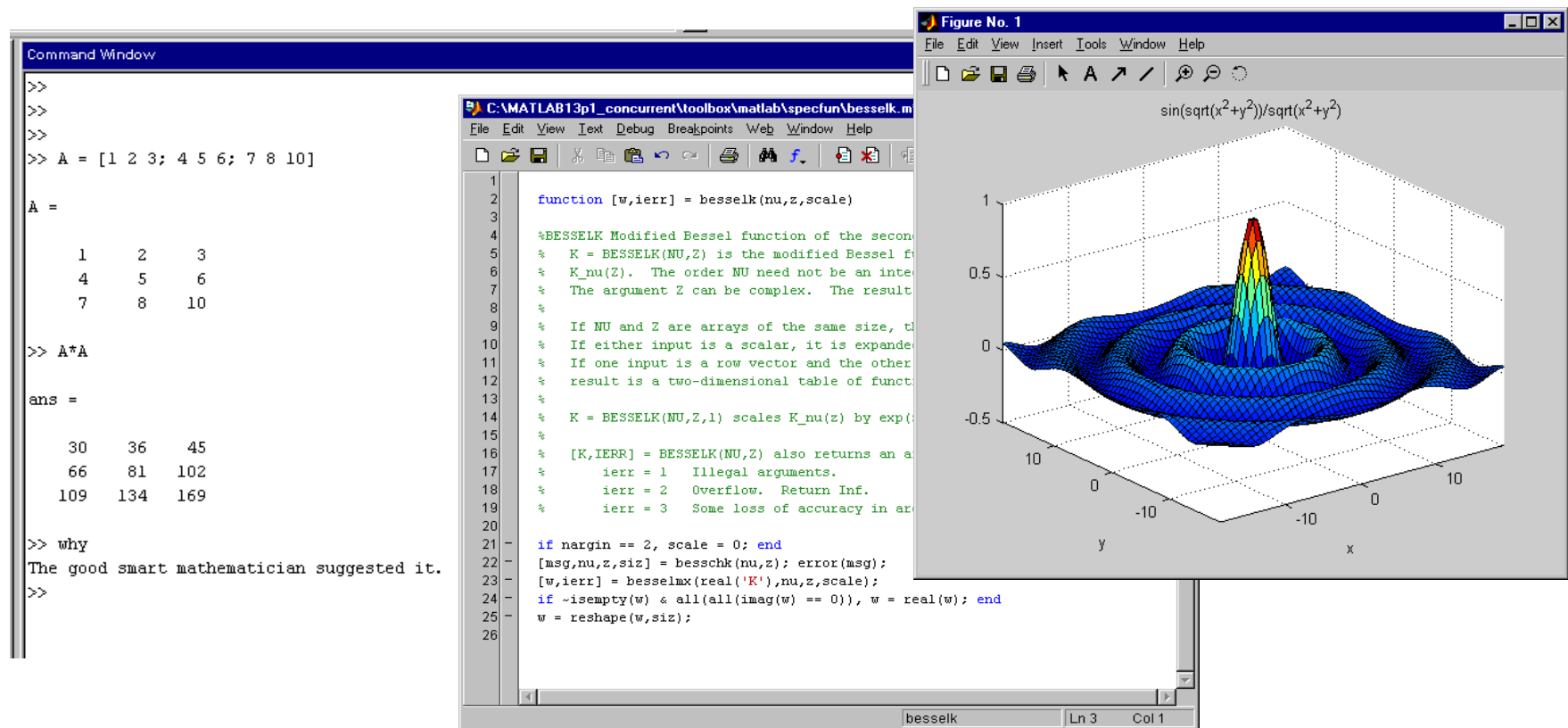
MATLAB/Simulink bietet dazu folgende Konzepte:

Toolüberblick.

MATLAB.

MATLAB:

- Software-Paket für numerische Berechnungen
- Erstellung von Grafiken zur Visualisierung von Daten
- interaktive Sprache und Programmierumgebung

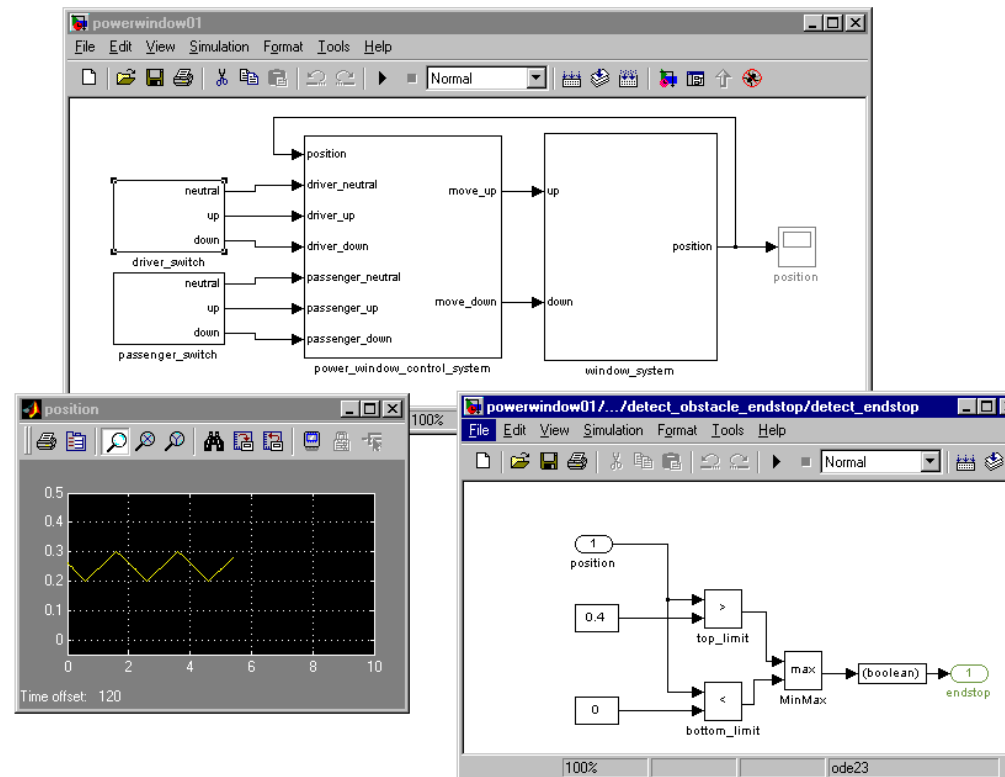


In die MATLAB Entwicklungsumgebung sind verschiedene SW-Pakete modular integrierbar:

Toolüberblick. Simulink.

Simulink:

- eine auf Blockdiagrammen basierende grafische Oberfläche zum Entwurf, zur Simulation und Analyse dynamischer Systeme.



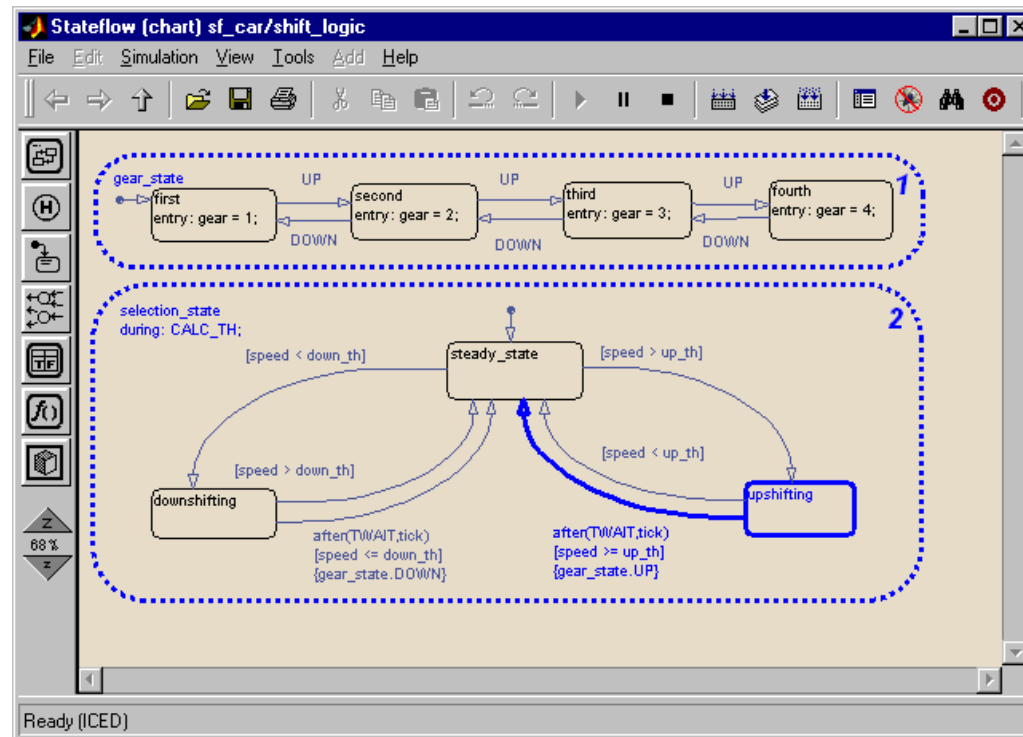
Darstellbar:

- lineare oder nichtlineare zeitabhängige Vorgänge, die sich mit Differential-, Differenzgleichungen und Algebraischen Ausdrücken beschreiben lassen.

Toolüberblick. Stateflow.

Stateflow :

grafische Simulations-Umgebung basierend auf Zustandsautomaten für den Entwurf ereignisgesteuerter Systeme.



Typische Anwendungsbereiche:

- Steuerlogiken in Regelungssystemen
- Modellierung von ereignisgesteuertem Verhalten
- Synchronisation von Modell-Subsystemen.

Toolüberblick.

Toolboxes, Blocksets und Codegeneratoren.

Weitere **Toolboxes und Blocksets** für unterschiedlichste Anwendungsgebiete, wie z. B. die Entwicklung von Regelsystemen

Ergänzend:

C-Codegenerator: **Real-Time Workshop**

→ Test von Entwürfen in Echtzeit-Umgebungen

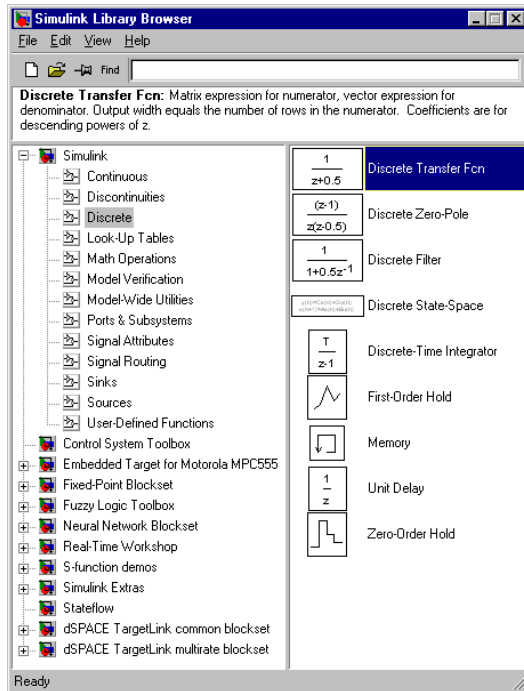
→ Durchführung notwendiger Änderungen vor Produktion der Entwicklung.

Darüber hinaus: Integrationsmöglichkeit von Drittanbieterprodukten – insbesondere Codegeneratoren:

– **TargetLink** (Firma dSPACE, enge Integration)

– **SCADE** (Firma Esterel, „Input/Output“-Integration)

Modellierung und Simulation. MATLAB/Simulink/Stateflow.



Simulink Bibliothek:

ca. 200 Blöcke zur Modellbildung:

- Eingangssignale
- dynamische Elemente
- algebraische und nichtlineare Funktionen
- Datenanzeige-Blöcke

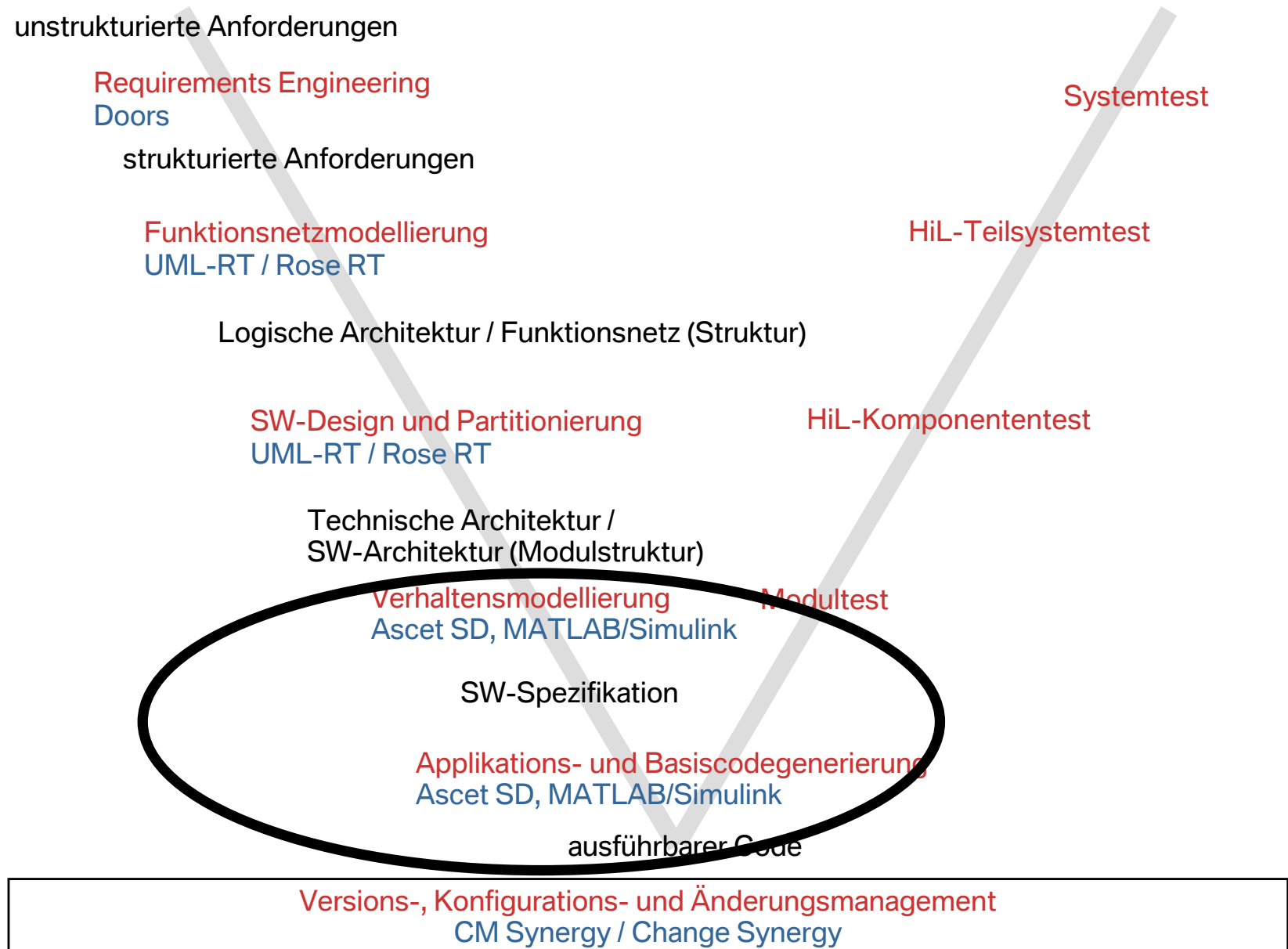
Verwendung von Triggern, Enables und Disables in Simulink-Blöcken zum Einbau konditional ausführbarer Subsysteme in die Modelle

Möglichkeit von Offline-**Simulationen** ("Perfect World Assumption") von

Regler- und Streckenmodell ("Closed Loop Simulation") zur

- Entwurfsbewertung
- Korrektur von Fehlentwicklungen
- Erstellung der Grundapplikation der Regler vor Prototypaufbau

Präsentation einzelner Entwicklungswerkzeuge. Verhaltensspezifikation mit ASCET-SD.



Verhaltensspezifikation mit ASCET-SD. Einsatzgebiet und Aufgabenbereiche.

Einsatzgebiet

Entwicklung datenfluss- und/oder zustandsbasierter Steuer- und Regelungsalgorithmen für eingebettete Systeme.

Aufgabenbereiche

- Modellierung
- Simulation
- Rapid Prototyping
- Codegenerierung

Hersteller

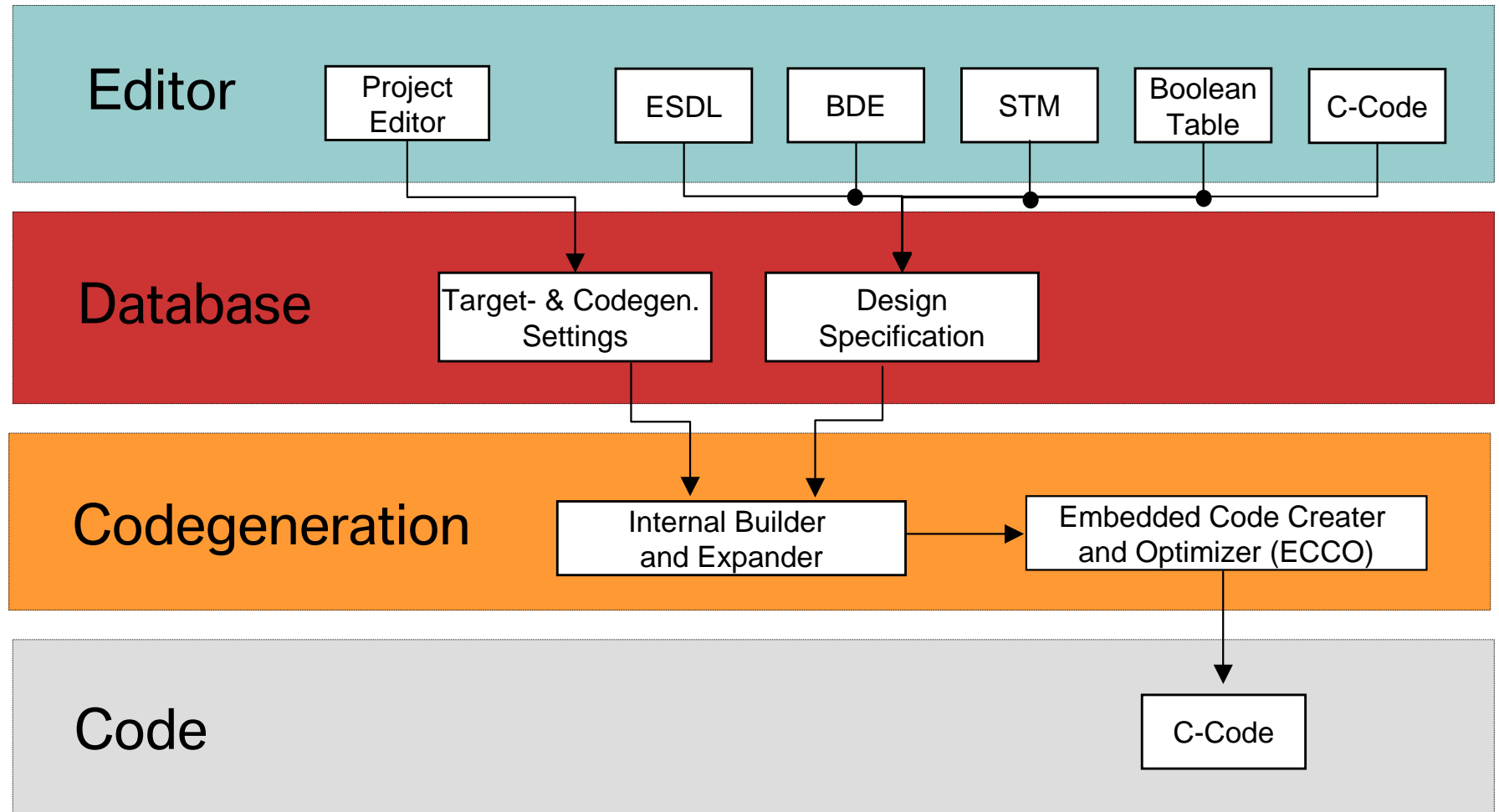
ETAS Entwicklungs- und Applikationswerkzeuge für elektronische Systeme GmbH

-Sitz in Stuttgart

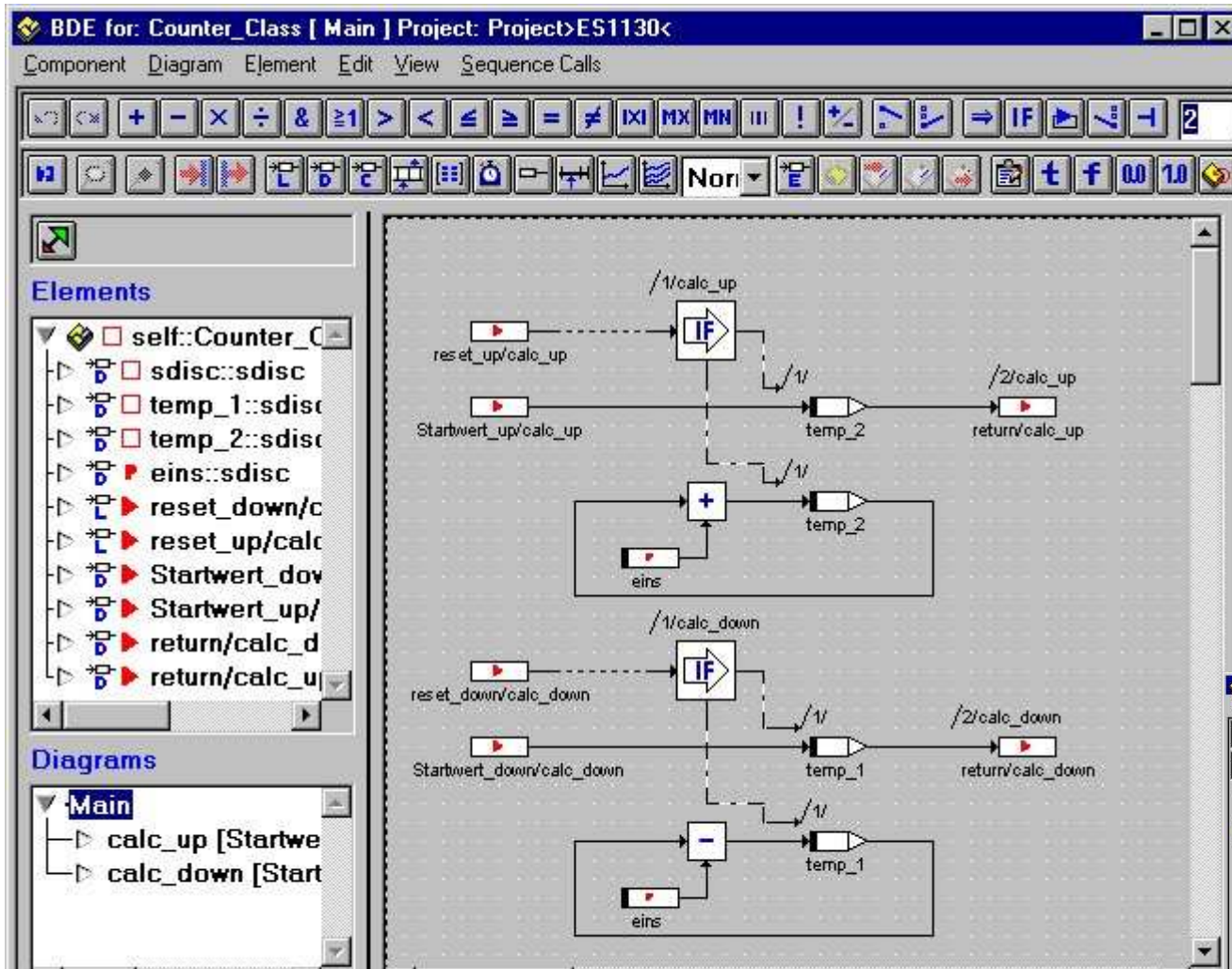
-Gründung 1994 als Tochtergesellschaft der Robert Bosch GmbH

Verhaltensspezifikation mit ASCET-SD.

Übersicht über Modellierung und Codegenerierung.

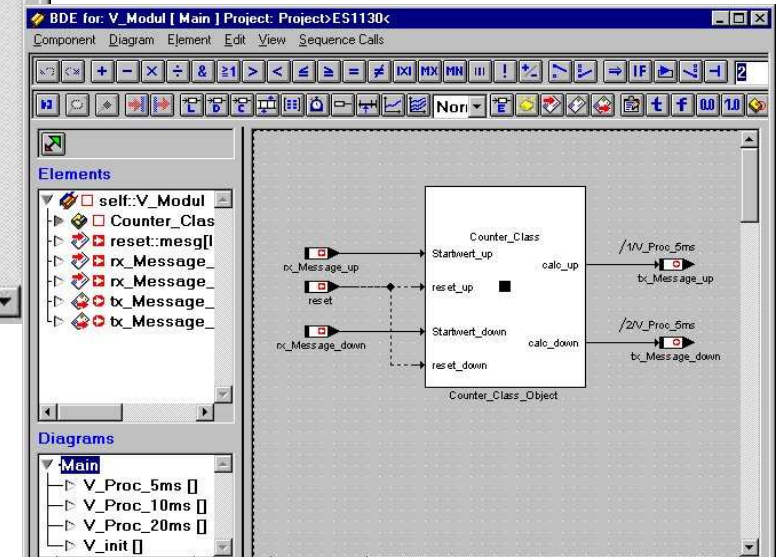


Verhaltensspezifikation mit ASCET-SD. Modelleditor.



Beschreibungs-
möglichkeiten einer
Funktion in ASCET-SD:

- Blockdiagramm (BDE)
- Zustandsautomat (STM)
- Boolesche Tabelle
- ESDL
- C-Code



Verhaltensspezifikation mit ASCET-SD.

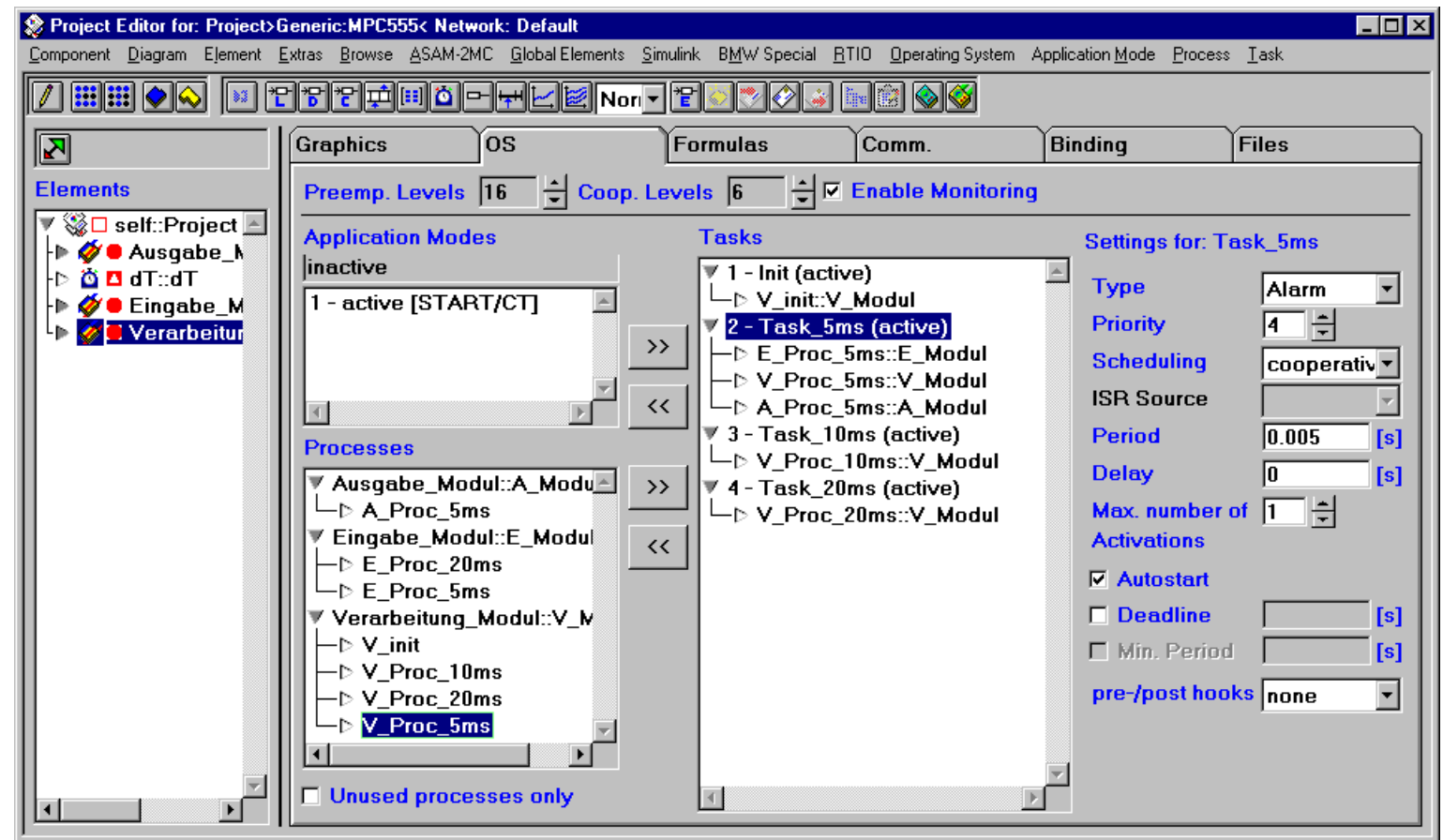
Projekt: OS-Konfiguration und Taskzuordnung

Projekt: ASCET-SD Element

- zur vollständigen Beschreibung eines eingebetteten Softwaresystems
- umfasst alle notwendigen Verhaltensspezifikationen (als Prozess definiert)
- stellt Schnittstelle zum Betriebssystem dar.



Generierter
Code

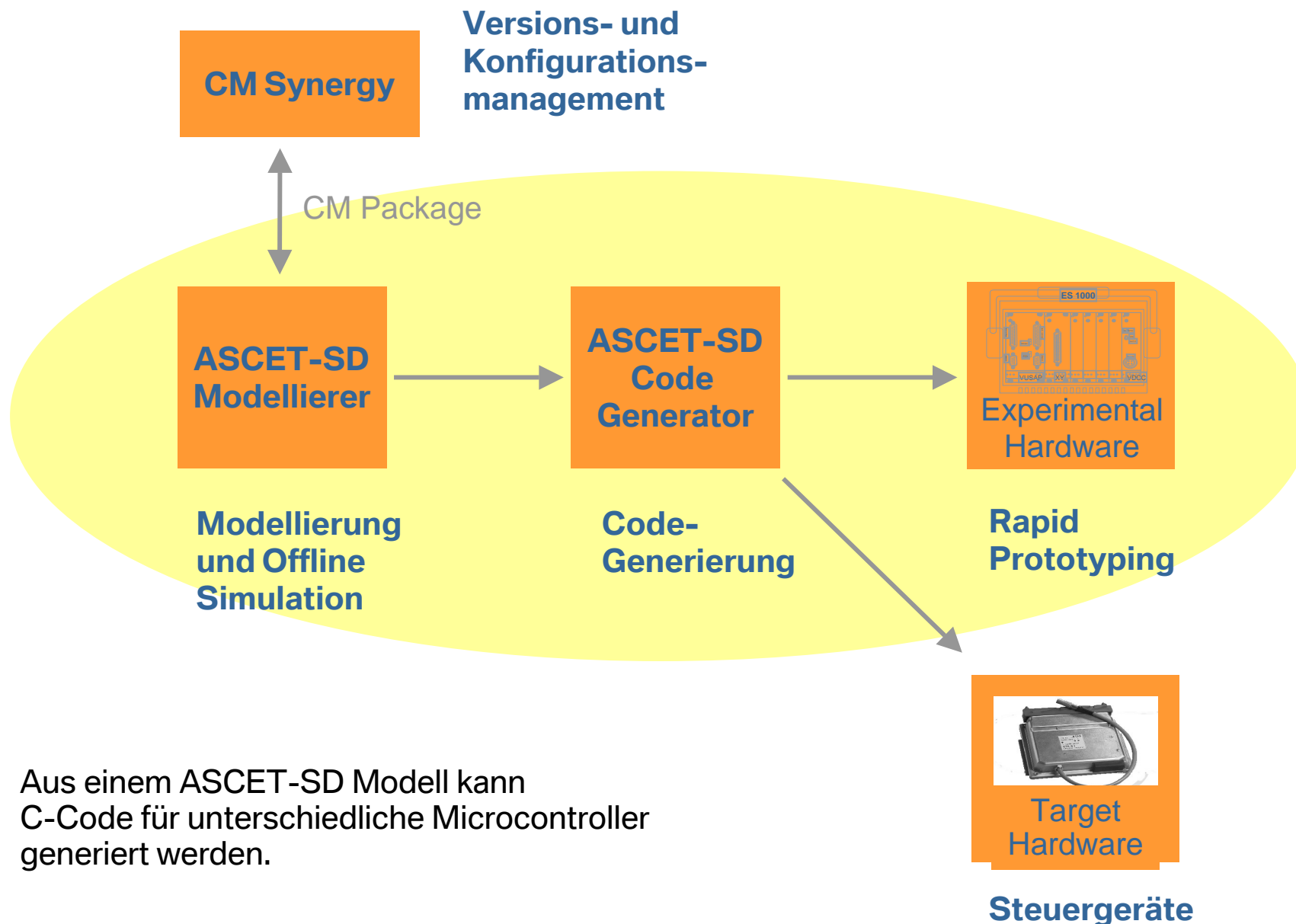


Verhalten wird in Prozessen
beschrieben.

In einer Task befinden sich
ausschließlich Prozesse mit
gleichem Zeitzyklus.

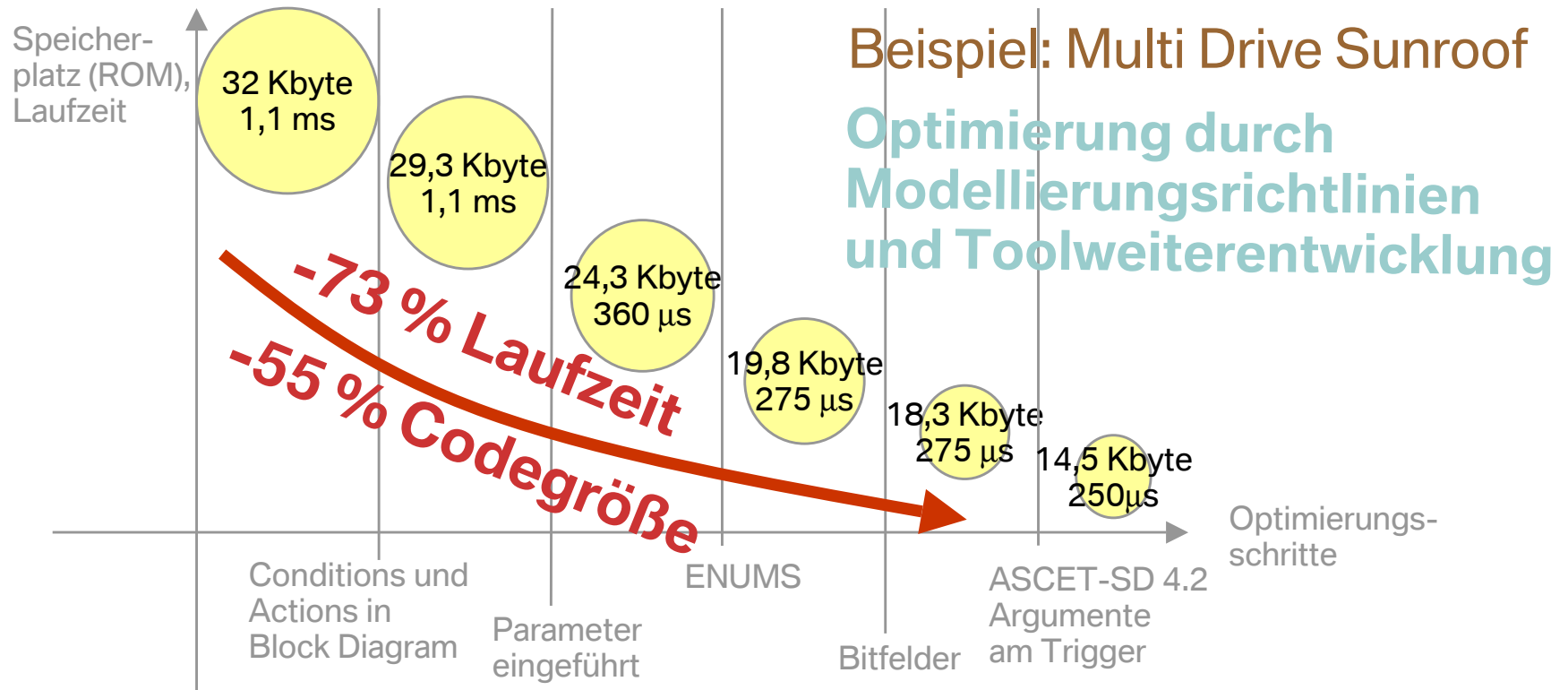
Eigenschaften einer
Task im Betriebssystem
werden hier konfiguriert.

Verhaltensspezifikation mit ASCET-SD. Anwendung in BMW Serienentwicklung.



Verhaltensspezifikation mit ASCET-SD. Erfahrungen.

Schrittweise Modelloptimierung in ASCET-SD

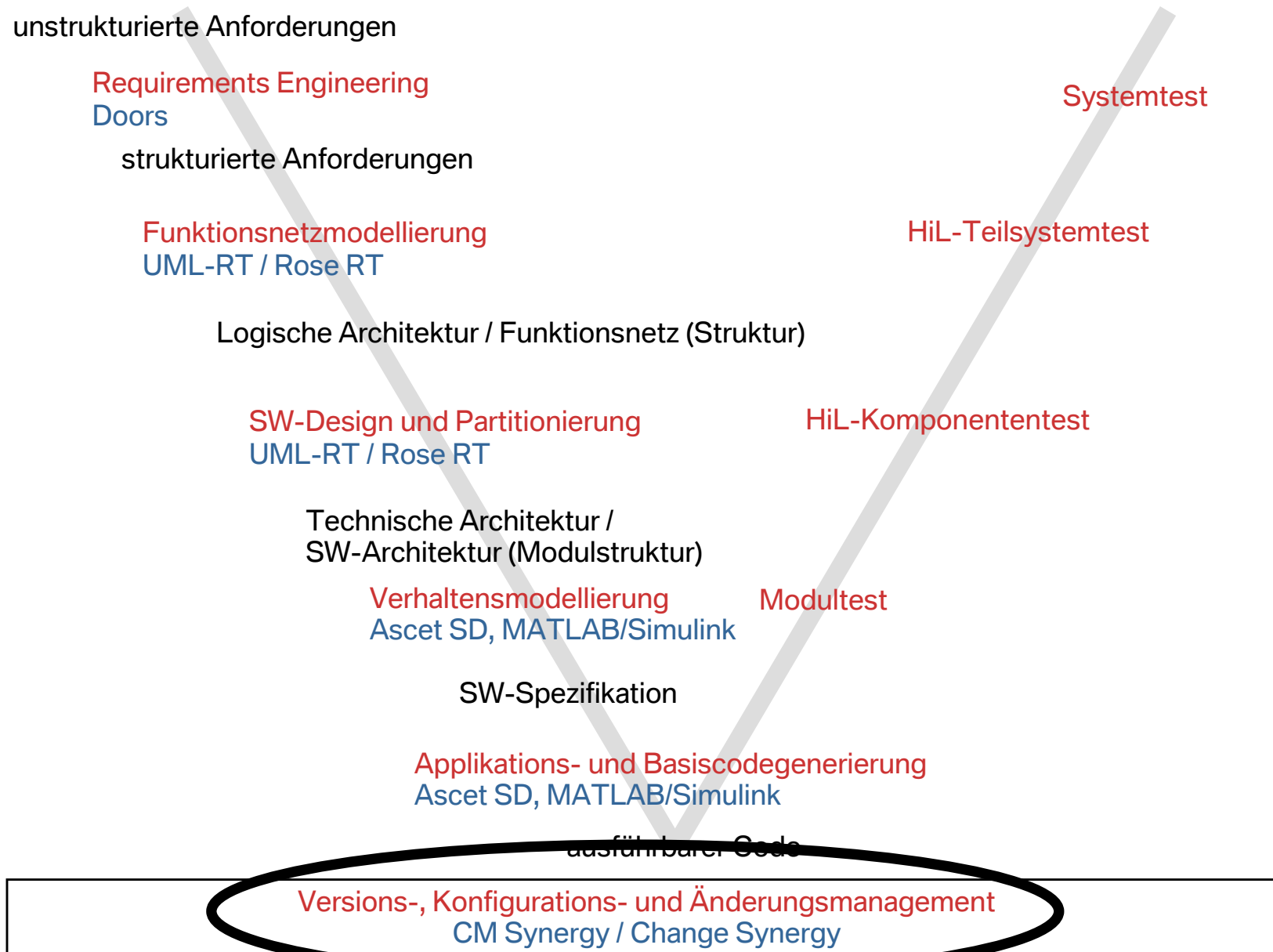


Gegenüberstellung von Autocode und Handcode

- Bei zertifizierter Toolkette ist bei Autocode **kein Codereview** notwendig!
- Bei Beachtung von Modellierungsrichtlinien sind Laufzeit, ROM- und Flashbedarf praktisch gleich, Autocode ist aber im Vergleich zur Handprogrammierung reproduzierbar.

Präsentation einzelner Entwicklungswerkzeuge.

Versions-, Konfigurations- und Änderungsmanagement.



Versions-, Konfigurations- und Änderungsmanagement. Problemstellung.

Ändere mal
eben...!

... Aber Meier hat
gesagt, ich muss das
ändern.

Warum hatten wir
das so gemacht ?

...welche war noch
die richtige Datei ???

- Wie organisiert man die SW-Entwicklung über die Zeit?
- Wie findet man die Quellen zu bereits ausgelieferten Ständen?
- Wie kann man die Änderungsursachen später nachvollziehen?

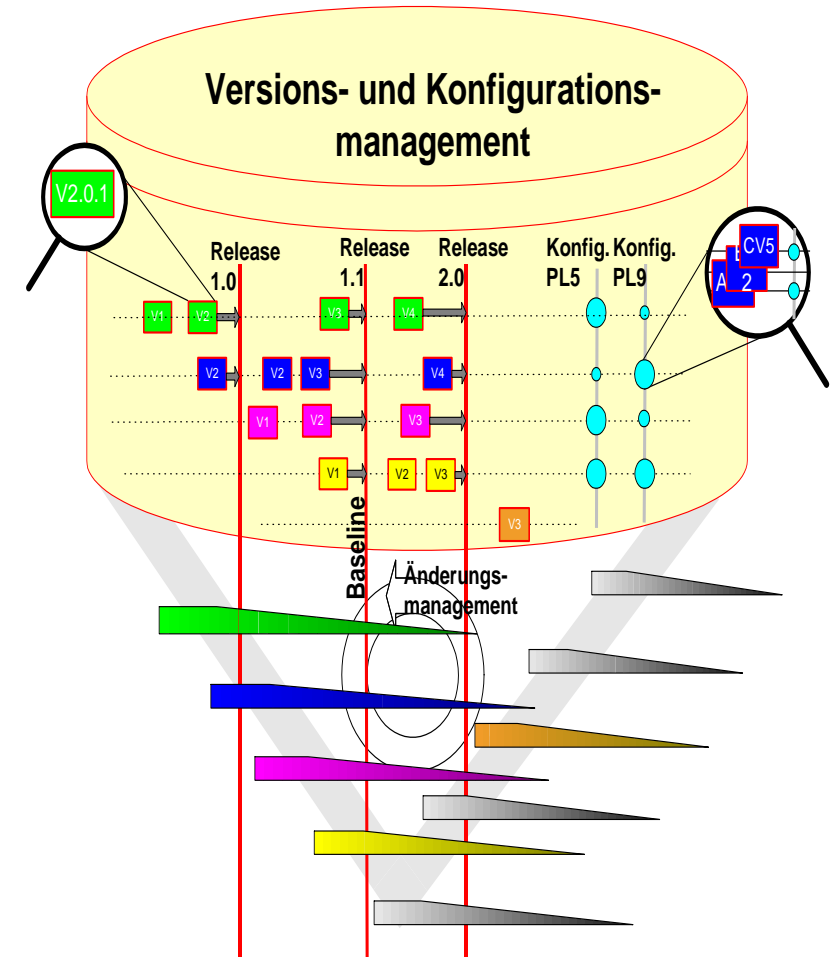
Versions-, Konfigurations- und Änderungsmanagement. Vorgehen zur Problemlösung.

- Versions- und Konfigurationsmanagement
 - archiviert Artefakte eindeutig, konsistent und rückverfolgbar
 - strukturiert Systeme und macht diese über den gesamten Lebenszyklus beherrschbar und nachvollziehbar

- Änderungsmanagement
 - Kontrollierte Einbringung von Änderungen, also keine „Änderungen auf Zuruf“
 - Änderungsbegehren an Artefakten werden von einem Gremium bewertet
 - Änderungen sind nachvollziehbar

Versions-, Konfigurations- und Änderungsmanagement. Methodik (1).

- Eindeutiges Identifizieren der Artefakte
- Kontrollierte Ablage der Artefakte über die Zeit (Versionierung)
- Erzeugung von Varianten und Zweigen (Versionierung)
- Unterstützung von Parallelarbeit
 - exklusive Bearbeitung eines Artefakts oder
 - kontrollierte Parallelbearbeitung mit anschließendem Merge
- Bilden von Konfigurationen:
Identifizieren zusammengehörender Artefakte



Versions-, Konfigurations- und Änderungsmanagement. Methodik (2).

Definition eines Änderungsmanagement-Prozesses:

- Änderungsantrag („Change Request“)
- formale Prüfung (Berechtigung und Vollständigkeit)
- inhaltliche Prüfung durch ein Gremium (Change Control Board)
- technische und wirtschaftliche Bewertung

Planung und Dokumentation:

- Erstellen eines Konfigurationsmanagement-Plans
(Rollen, technische Aspekte)
- Release-Plan: Schnittstelle zum Projektmanagement

Versions-, Konfigurations- und Änderungsmanagement. Definitionen.

- Version:
Eindeutig nummerierter Entwicklungsstand eines einzelnen Artefakts
- Konfiguration:
Konsistente Zusammenstellung mehrerer Entwicklungsergebnisse
- Release:
Auslieferungsrelevanter, die Funktionalität erweiternder, vollständiger und konsistenter Entwicklungsstand aller hierfür erforderlichen Entwicklungsergebnisse
- Baseline:
Stabiler, konsistenter vollständiger und dokumentierter Stand von Entwicklungsergebnissen, der Ausgangspunkt der Entwicklung für das nächste Release ist, oft das momentan aktuelle Release.

Versions- und Konfigmgmt. mit CM Synergy. Begriffsbestimmungen (1).

- Rollen
- Task (insbesondere bei CM Synergy!)
- Folder
- Projekt
- Work-Area
- Check-In / Check-In-Task / Check-Out
- Reconfigure

Versions- und Konfigmgmt. mit CM Synergy. Begriffsbestimmungen (2).

Rollen:

- Build-Manager
 - Verantwortlich für Release-Bildung
 - Testet die Systemintegrität
 - Legt Tasks (Arbeitspakete) fest
 - Stellt Entwicklern die neuesten Objekte zur Verfügung
- Developer
 - Arbeitet die einzelnen Tasks ab
 - Holt sich freigegebene Objekte (Dateien)

Versions- und Konfigmgmt. mit CM Synergy. Begriffsbestimmungen (3).

Task:

- Arbeitspaket für einen Entwickler
- Eine Task ist immer mit einem bestimmten Release verknüpft.
- Eine Task besitzt einen Status:
registered, assigned oder completed
- Eine Task kann eingecheckt werden. Damit werden alle mit der Task verknüpften Dateien von der Work-Area in ein Projekt eingecheckt.

Versions- und Konfigmgmt. mit CM Synergy. Begriffsbestimmungen (4).

Folder:

- Ein Folder dient zur Sammlung von Tasks.
- In einem Folder sind Dateien nur indirekt über eine Task vorhanden.
- Folder dienen zur Definition eines Workflows für Tasks.
Dieser Workflow definiert Reifegrade von Tasks:
 - Bearbeitung abgeschlossen
 - Integration getestet
 - System getestet

Versions- und Konfigmgmt. mit CM Synergy. Begriffsbestimmungen (5).

Projekt:

- Ein Projekt beinhaltet Objekte (Dateien) einer bestimmten Version.
- Je nach Konfiguration enthält ein CM-Synergy-Projekt selbst wieder Projekte.
- Die mit einem Projekt verknüpften Folder bestimmen den Inhalt des einzelnen Projekts.

Versions- und Konfigmgmt. mit CM Synergy. Begriffsbestimmungen (6).

Work-Area:

- Die Work-Area ist die physikalische Kopie eines Projektes.
- Die Work-Area befindet sich in der Regel auf der lokalen Festplatte.
- Hier sind die jeweiligen Dateien in der gleichen Hierarchie wie im Projekt enthalten.
- Alle Entwicklungstools greifen auf die Work-Area zu.

Versions- und Konfigmgmt. mit CM Synergy. Begriffsbestimmungen (7).

- **Check-In / Check-In-Task:**
 - Check-In kopiert eine Datei von der Work-Area in das Projekt.
 - Check-In-Task kopiert alle mit der Task verknüpften Dateien von der Work-Area in das Projekt.

- **Check-Out:**
 - Check-Out ändert den Status einer Datei auf „in Bearbeitung“.

- **Reconfigure:**
 - Reconfigure kopiert freigegebene und dem Entwickler zugewiesene Dateien gemäß dem einstellbaren Workflow vom Projekt in die Work-Area.

Änderungsmanagement mit Change Synergy. Überblick.

Werkzeug Change Synergy:

- webbasierte Client-Server Anwendung
 - Aufgabe: Verwaltung und Dokumentation von Änderungsbegehren (Change Requests) an Software
 - Statistiken, Reports über die Change Requests möglich
 - Rollen legen die Verantwortlichkeiten fest
 - Definierter Workflow
 - Verknüpfung mit CM-Synergy Tasks möglich
- Umfassend integriertes Änderungs- und Konfigurationsmanagement möglich.

Änderungsmanagement mit Change Synergy. Eingabemaske.

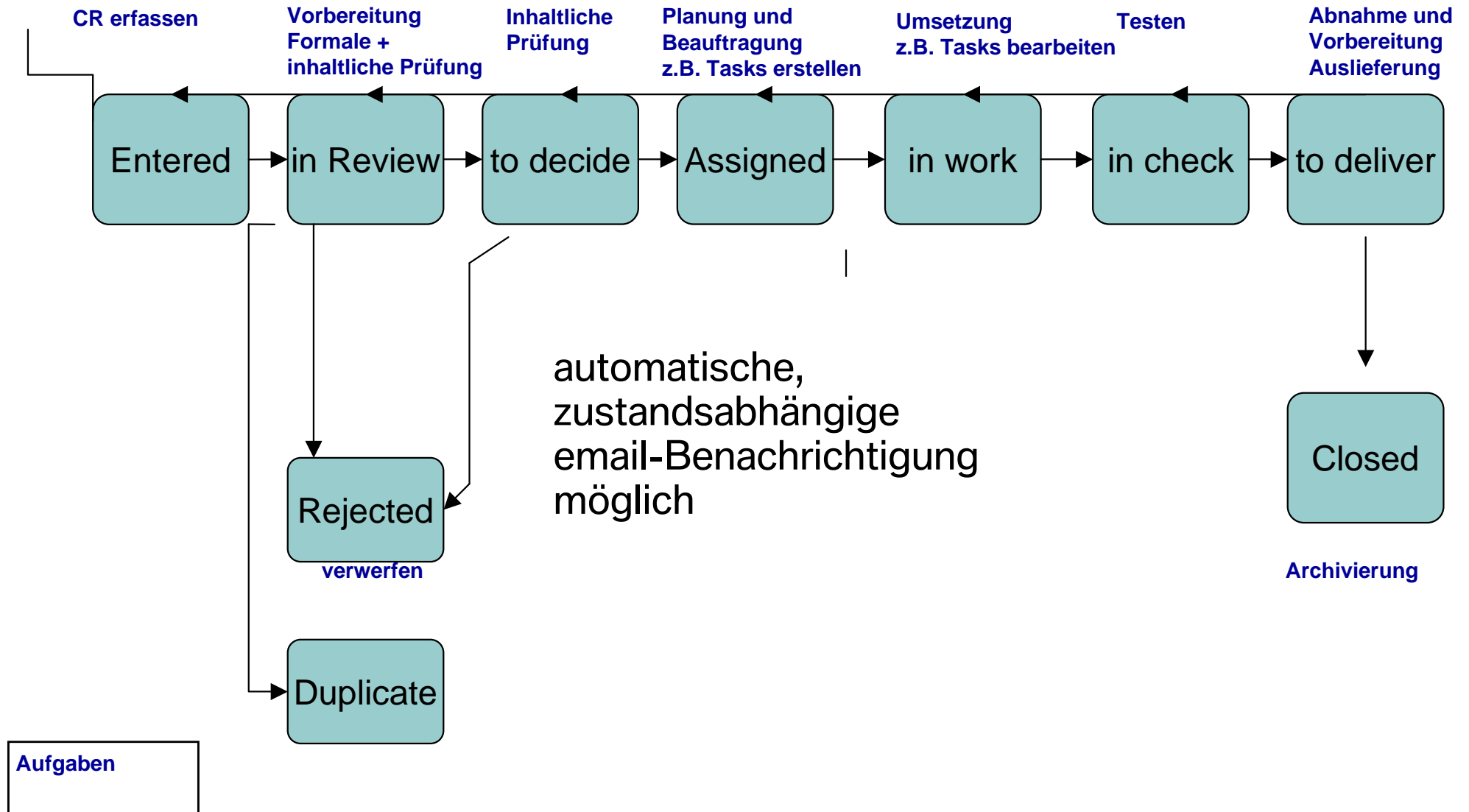
„Elektronisches Formular“

Statistiken und Abfragen

- Automatische Überprüfung der Eingabefelder möglich
- Verschiedene Eingabefelder möglich (Auswahlfelder, ...)

Pflichtfelder in diesem Zustand

Änderungsmanagement mit Change Synergy. Beispielworkflow.



Zusammenfassung.

- Anforderungen an Methoden und Entwicklungswerkzeuge
- Eine Methoden- und Werkzeugkette für einen Entwicklungsprozess
- Präsentation einzelner Entwicklungswerkzeuge:
 - Requirements Engineering
 - DOORS
 - Funktionsnetzmodellierung
 - Rose RT
 - Verhaltensspezifikation
 - MATLAB/Simulink
 - ASCET-SD
 - Versions-, Konfigurations- und Änderungsmanagement
 - CM Synergy
 - Change Synergy