



Dr. Peter DENCKER
ETAS GmbH

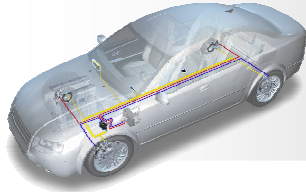
Modellbasierte Entwicklung effizienter SW

in der Automobilindustrie

Software-Workshop

24.01.2008, Technologiepark Karlsruhe

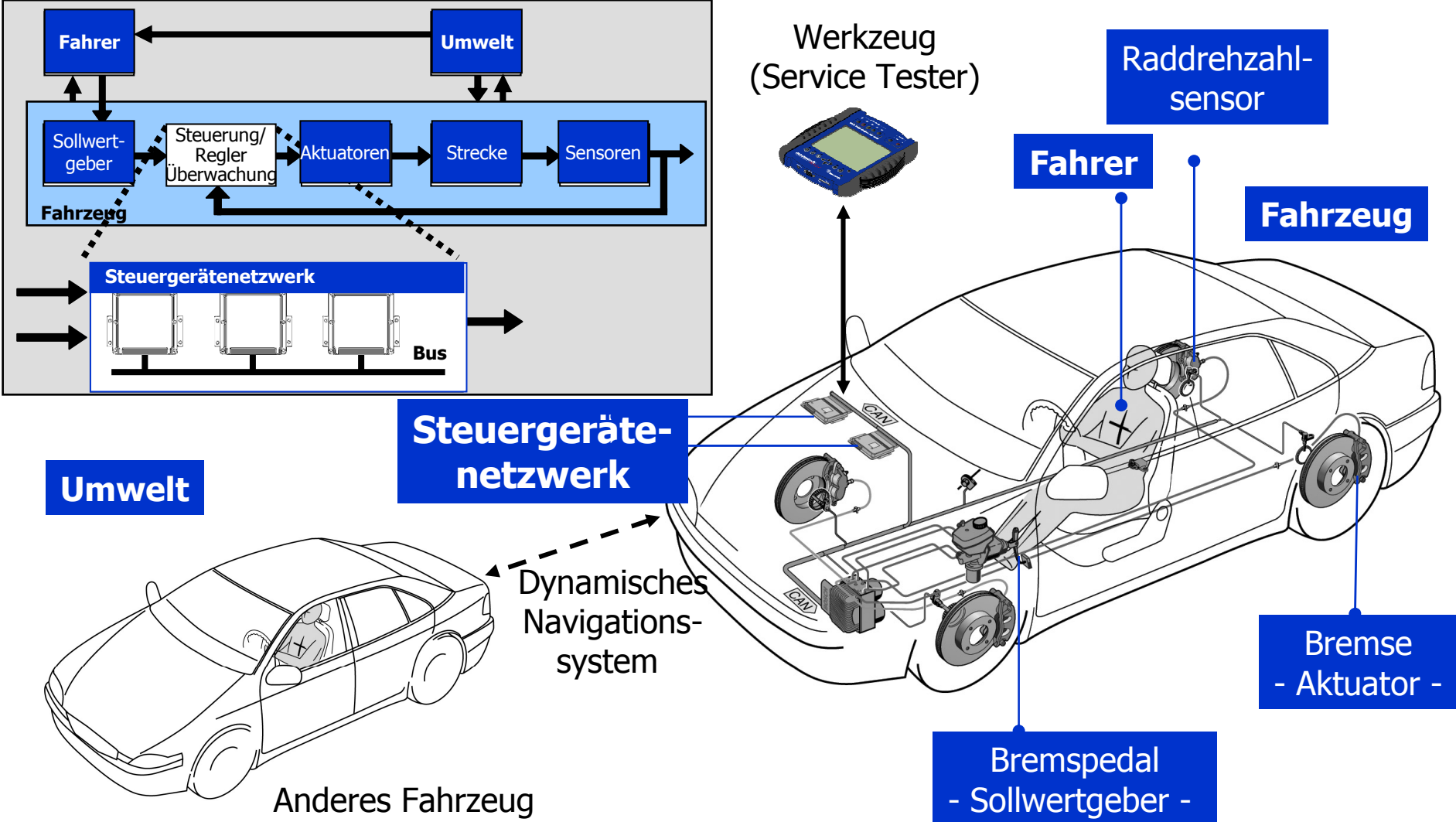
Agenda



- **Ist Automotive Software speziell?**
- **Konsequenzen aus den speziellen Anforderungen**
- **Effizienz im Entwicklungsprozess**
- **Automotive Software Modellierungswerkzeuge**
- **Warum automatische Codeerzeugung?**
- **ASCET Code Generation**
- **Code Effizienz**
- **AUTOSAR**
- **Zusammenfassung**

Ist Automotive Software speziell?

Beispiel: Fahrwerksregelungssystem



Automotive Software ist speziell, weil

- Der Preis von Steuergeräten in Verbindung mit **großen Stückzahlen** ergibt eine **extreme Kostensensibilität**
 - Beispiel VW Golf: sinkt der Preis des Motorsteuergeräts um 20€, so **spart VW 400 Mio €!**

Modell	Golf 1	Golf 2	Golf 3	Golf 4
Stückzahl	6.780.050	6.301.000	4.805.900	4.300.000

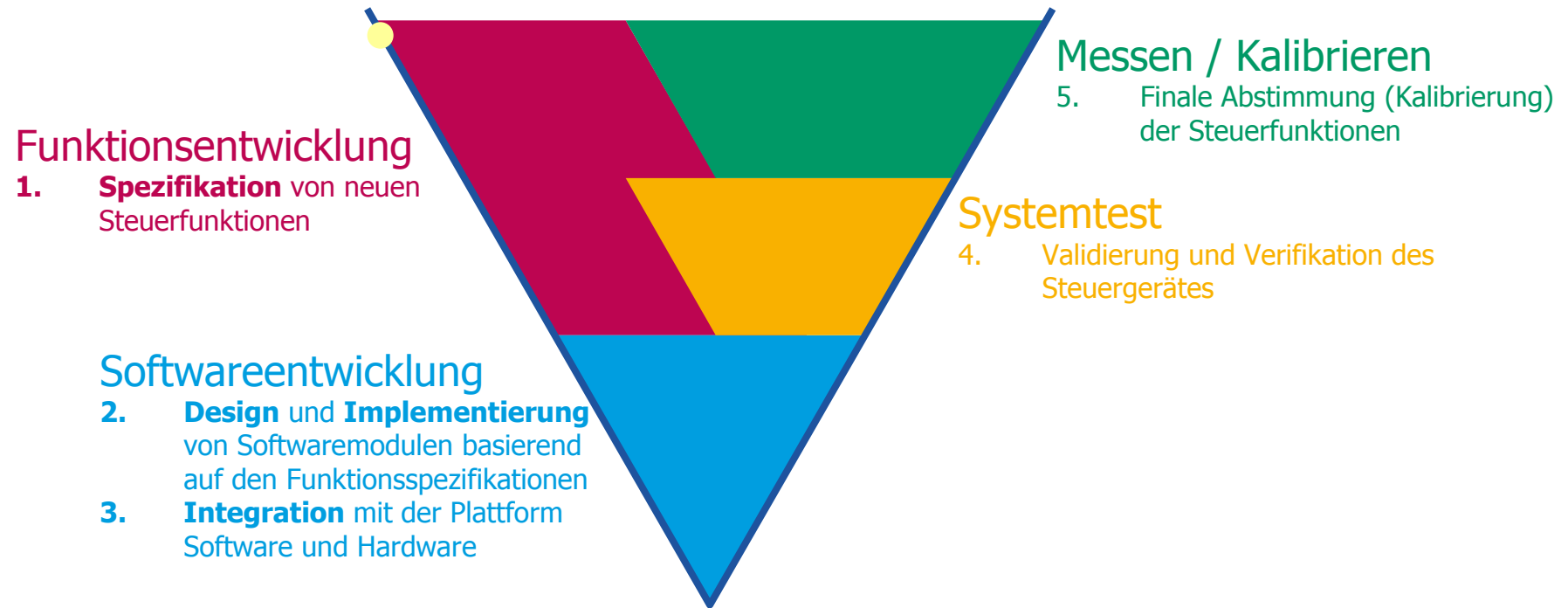
- Mangelnde **Zuverlässigkeit** der Software ist ein **Kostenrisiko**
 - 9. Dezember, 2007:
Ford ruft in dem USA 1,17 Millionen Fahrzeuge wegen einer Fehlfunktion des Motorsensors zurück.
 - Geht man von 100\$ aus pro Fahrzeug, wird das sehr teuer!

Automotive Software ist speziell, weil

- Die **Variantenvielfalt** fordert **spezielle Software Architekturen** und eigenen Entwicklungsmethoden
 - Beispiel Daimler: im Jahr 2006 wurde nur eine „Dublette“ der S-Klasse in Sindelfingen gebaut, d.h. es wurden 10.000 **verschiedene** S-Klasse PKW in 2006 gebaut.
 - Gleichzeitig **große Zahl** verschiedener PKW **Modelle** von vielen Herstellern
- Die **hohe Modellfrequenz** erfordert **schnelle Entwicklungszyklen**
 - Beispiel VW: 5 neue Golf Modellreihen in 30 Jahren, d.h. alle 6 Jahre ein kompletter Redesign, alle 2-3 Jahre eine Modellpflege.
- **90%** aller **neuer Fahrzeugfunktionalität** kommt aus der **Elektronik**
- **60%** der **Elektronikentwicklung** sind **SW Entwicklungskosten**

Automotive Software ist speziell

Konsequenzen für den Entwicklungsprozess



Durchgängige Toolunterstützung des obigen V-Modells
von der Spezifikation bis zum Kalibrieren

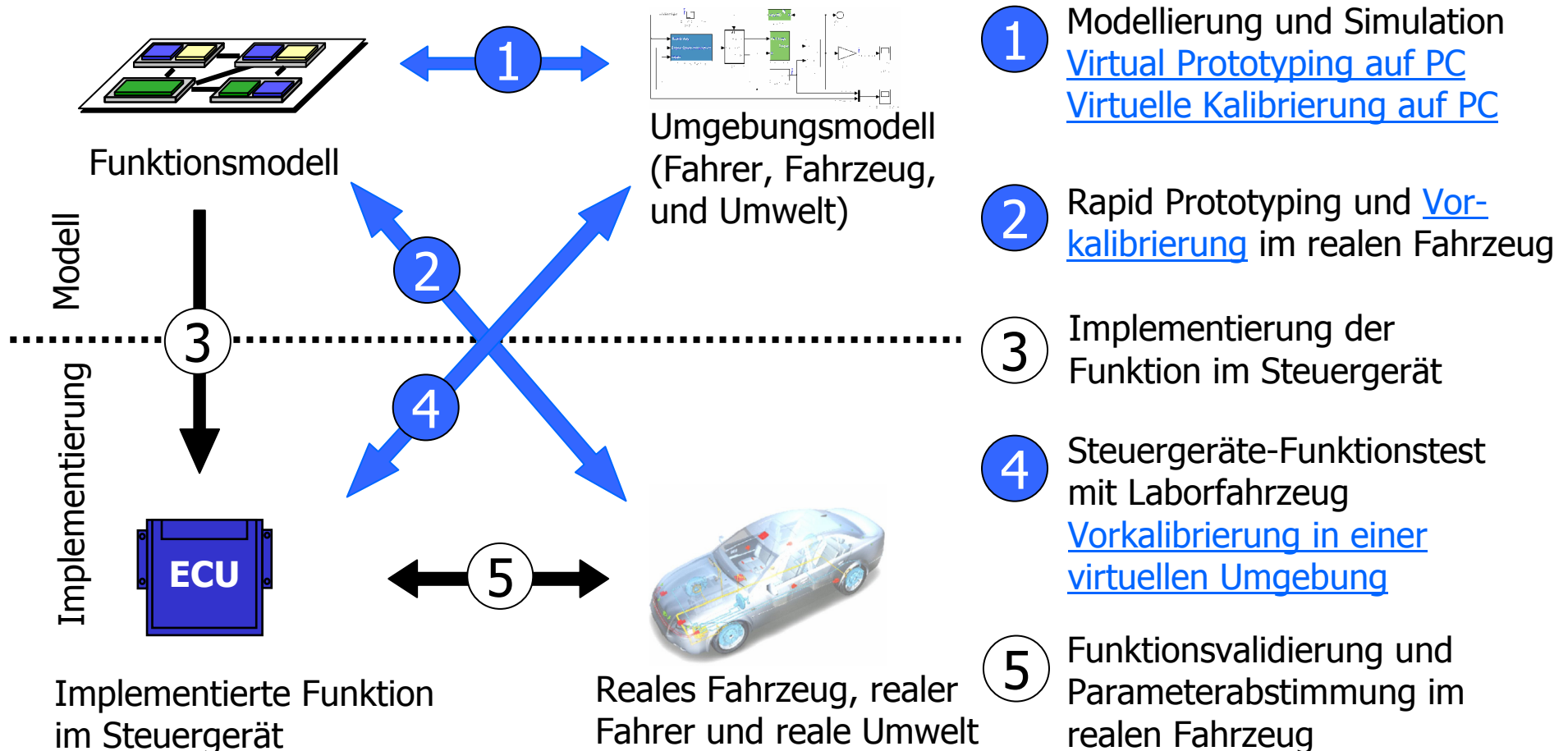
Automotive Software ist speziell

Konsequenzen für die Software/Hardware Architektur

- Software Architektur
 - Aufteilung in Plattformsoftware und Applikationssoftware (Varianten)
 - Parameter, Kennlinien und -felder statt Funktionen (Varianten)
 - Integer-Arithmetik statt Floatingpoint-Arithmetik (Stückkosten)
- Standardisierte Steuergeräte mit Plattform Software (Zuverlässigkeit, Stückkosten)
- Prototyping so früh wie möglich (Zuverlässigkeit, Zykluszeit)
 - Bypass Technologie ist die Folge
- Meistens wird eine neue Funktion aus bestehenden Funktionen abgeleitet und ist nicht komplett neu (Modellfrequenz).
- Sehr hohe Codeeffizienz (Stückkosten, Speicher+Geschwindigkeit)

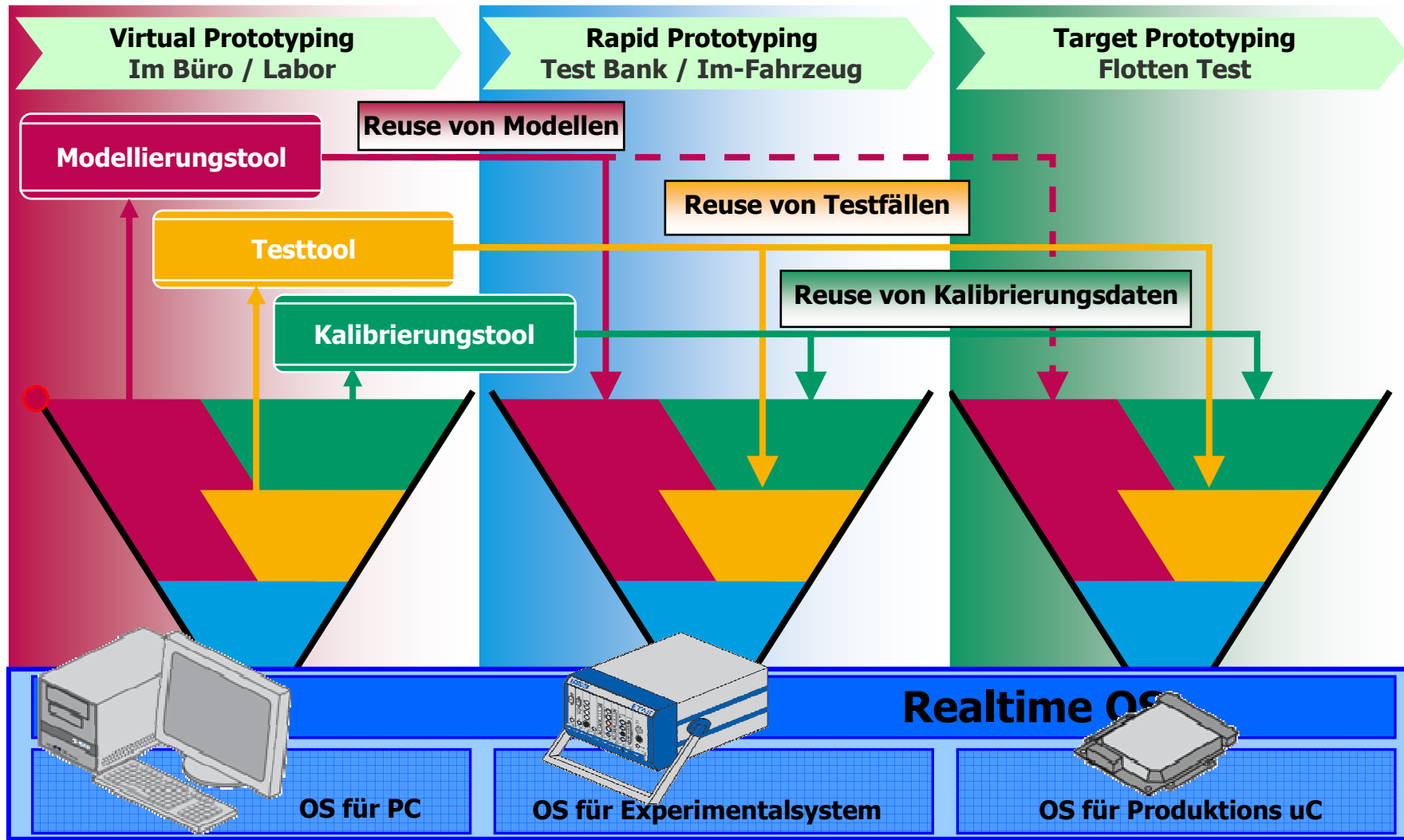
Effizienz im Entwicklungsprozess

Integrierte und virtuelle Entwicklung sichert Qualität



Effizienz im Entwicklungsprozess

Wiederverwendung vom virtuellen bis zum realen Prototyp



Automotive Software Modellierungswerkzeuge

Überblick – Verfügbare Modellierungselemente

Steuergeräte Softwareentwicklungsumgebung

Blockdiagramm
(Daten-, Kontrollfluss, OO-Modellierung, Hierarchien)

Standard Basis-Block Bibliothek

Zustandsautomaten

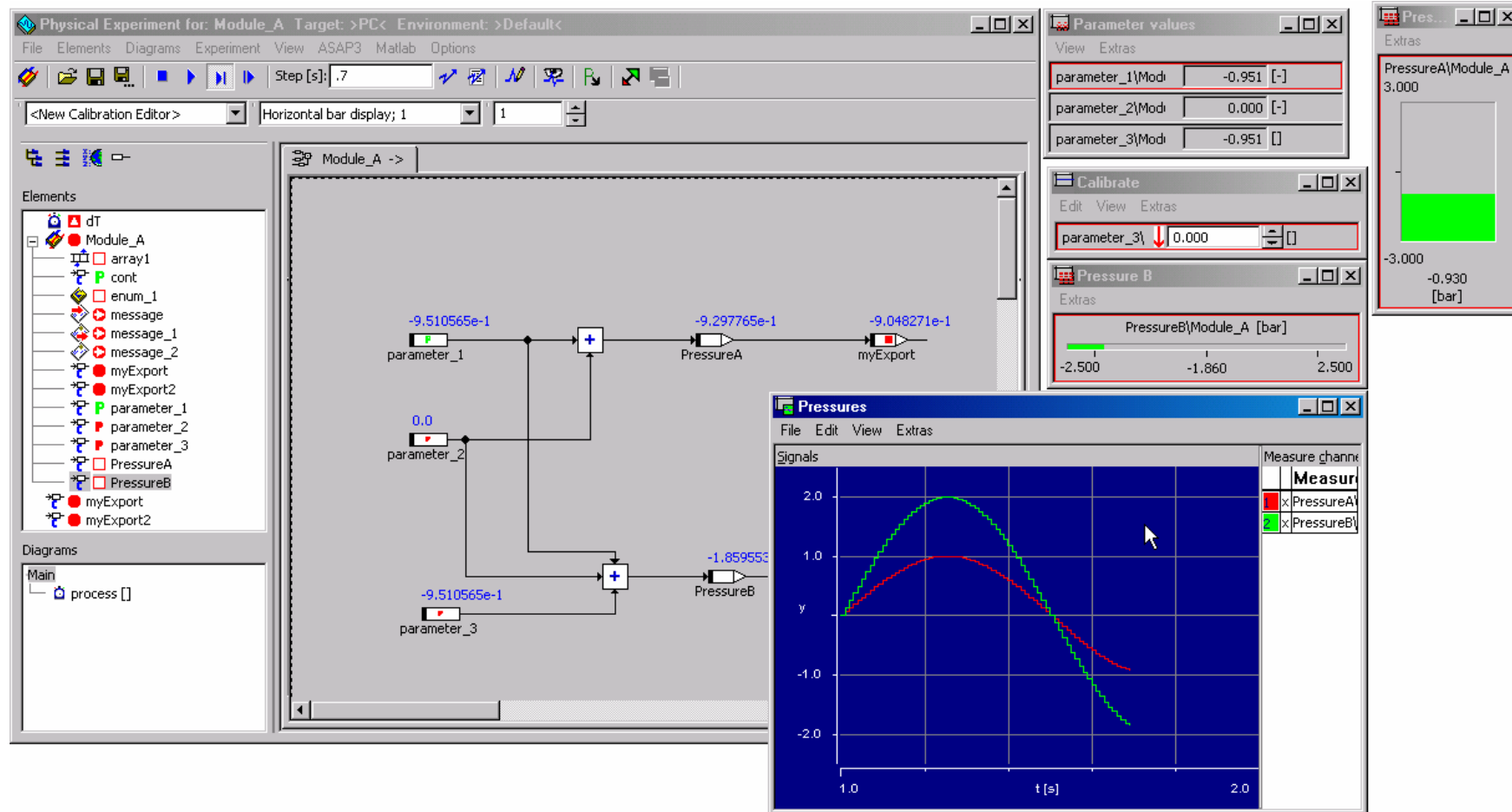
ESDL Beschreibung
(Java Syntax)

C Code Beschreibung

The screenshot displays a complex software development environment. On the left, a 'Blockdiagramm' window shows a control loop with blocks like 'difference::cont', 'PID1::PID1', and 'conversion_1::1D[cont]'. In the center, a 'Database Browser' window lists a 'Standard Basis-Block Bibliothek' with categories like 'Control' (DT1, P, PI, PID, PIDLimited, PILimited, PT1, PT2) and 'Transferfunction'. To the right, a 'Zustandsautomaten' window shows a state machine diagram with nodes like 'temperature' and 'speed'. Below the database browser, an 'ESDL Beschreibung' window shows Java-like code for array processing. On the far right, a 'C Code Beschreibung' window shows C code for reading sensor data from an ETK.

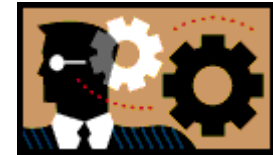
Automotive Software Modellierungswerkzeuge

Experimentalumgebung - PC und Experimentaltarget



Wozu Automatische Codeerzeugung?

Programmier Fehler im Hand Code



Programmierfehler wie z.B.

```
if (v_wheel=v_max) { /* should be (v_wheel== v_max) */  
    do_something(v_wheel);  
}
```

```
y = a/(x-z); /* (x-z) could turn 0 unexpectedly */
```

```
int32 dosomething(a uint32,b uint16) {  
    uint16 locVar;  
    ... /*locVar should be initialised */  
    return locVar;  
}
```

...

Wozu Automatische Codeerzeugung? Modelltransformation zur Integer Arithmetik



- Ein einfaches Beispiel:
$$p_E = \frac{(v_A + v_B) * k_C}{k_D}$$
 - v_A, v_B sind Geschwindigkeiten [**m/s**]
 - p_E ist der Bremsdruck [**bar**]
 - k_C Formfaktor [**Ns/m**]
 - k_D Konversionsparameter von Kraft zu Druck [**N/bar**]
- Zusätzliche physikalische Information
 - v_A, v_B mit Wertebereich [0..100] **m/s** und Quantisierung $q = 1/100$ **m/s**
 - $p_E \rightarrow$ [0..300] **bar**, $q = 1/4$ **bar**
 - $k_C \rightarrow$ [-20..20] **Ns/m**, $q = 0,3$ **Ns/m**
 - $k_D \rightarrow$ [10..20] **N/bar**, $q = 0,1$ **N/bar**

ASCET Codeerzeugung

Limitierter quantisierter Integer Code



BDE for: Module_BDE_Example [Main] Project: Project_Example [ANSI-C/Object]

Component Diagram Element Edit View Sequence Calls

```

3  /* public process [] */
9
0  void MODULE_BDE_EXAMPLE_IMPL_process (void)
1  {
2      /* temp. variables */
3      sint32 _t1sint32;
4
5      {
6          _t1sint32
7          = (sint32)(A + B) * C * (sint32)2 / (sint32)25;
8          _t1sint32 = ((D == (sint16)0) ? _t1sint32 : _t1sint32 / D);
9          /* process: sequence call #1 */
0          /* assignment to E: min=0, max=1200, hex=4phys+0, limit=(maxBitLength: true, assign: true), zero incl.=true */
1          E = ((_t1sint32 >= (sint32)0) ? ((_t1sint32 <= (sint32)1200) ? (uint16)_t1sint32 : (uint16)1200) : (uint16)0);
2
3      }
4  }
5  }

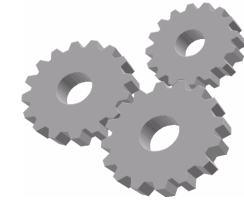
```

Diagrams

- Main
 - process []

Name	Type	Impl. Type	Impl. Min	Impl. Max	Q	Formula	Limit to me bit length	Limit Assignmer	Zero not incl.	Min
D	cont	uint8	10	20	0	F_01	Auto		No	1.0
E	mesg[cont]	uint16	0	1200	0	F_025	Auto	Yes	No	0.0

ASCET Codeerzeugung Optimierung



- **Benutze 2er Potenz Näherungswerte für Literale**
- schnellere Berechnung, auf Kosten der Genauigkeit -

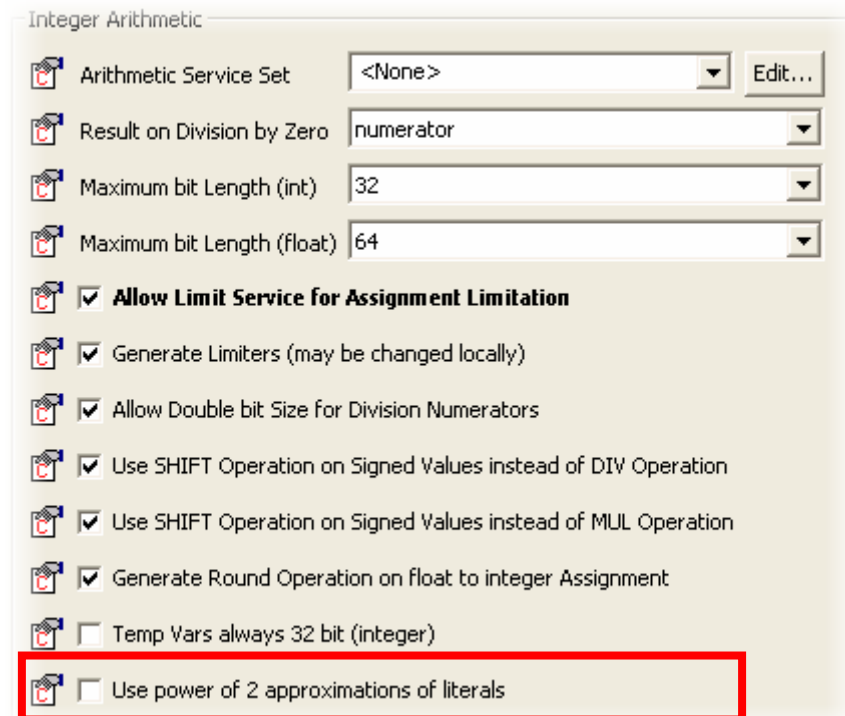
Z.B.:

Näherungswert für Literal **0.866** ist

$$28377/2^{**}17 \sim 0,86599731$$

anstelle von

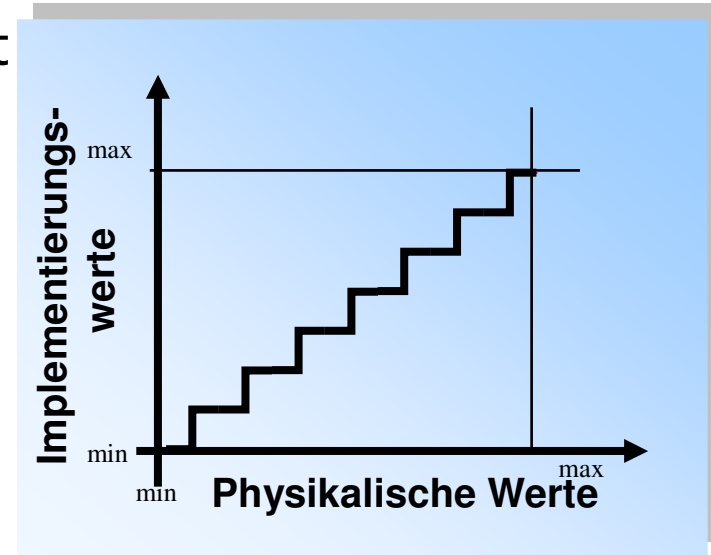
$$433/500 = 0.866$$



ASCET Codeerzeugung Codegenerator Eigenschaften



- Automatische Transformation geschieht unter folgenden Randbedingungen
 - max. verfügbare Bitlänge des Zielprozessors
 - Implementierungstypen
 - Wertebereiche
 - Konversionsformeln

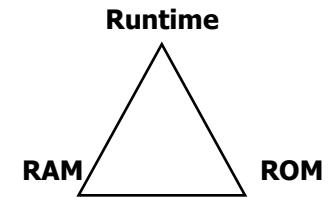


mit **automatischer**

- Behandlung von Überläufen
- Erzeugung von Quantisierungs- und Korrekturfaktoren (Konversionsfaktoren werden kombiniert)
- Erzeugung von limitiertem Code
- Vermeidung von Division durch "0" / "geschützte" Division durch "0"

Codeeffizienz

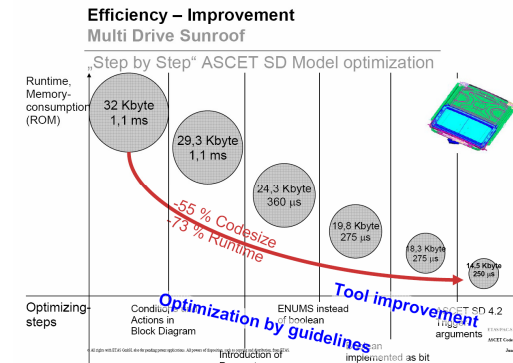
Kundenaussagen



- BMW AG

“Because of modelling guidelines the demand for runtime, ROM and flash is almost identical for automatically generated C-Code and C-Code which is coded by hand.”

Ref: BMW/Department Head Electronic Development, Mr. J. Hauser
– ETAS Competence Exchange Symposium 2004



- Robert Bosch GmbH – GS (Gasoline System)

„For complex models, the deployment of ASCET V5.1 may yield time savings in the area of up to 30 percent.“

Ref: BOSCH-GS/, Mr. Nicolaou – ETAS RealTimes

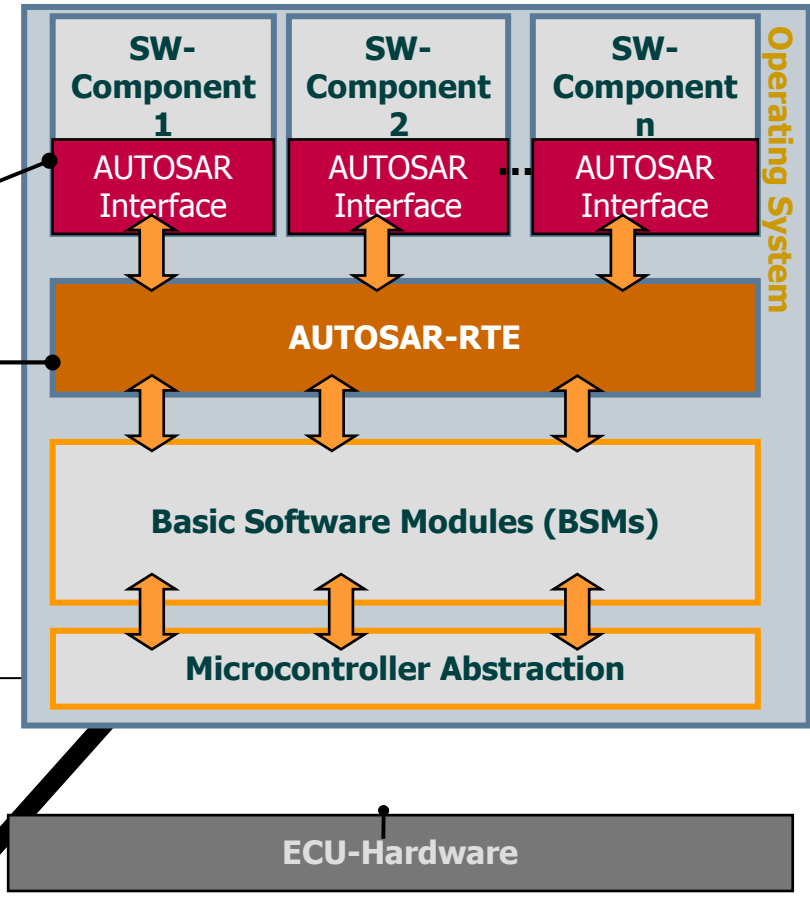
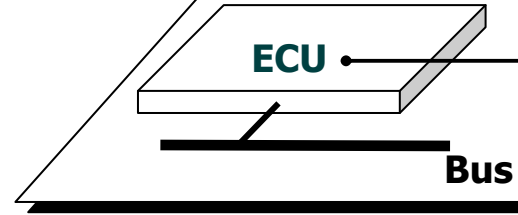


AUTOSAR : Ein neuer Standard

Steuergerätesoftwarearchitektur, in Zukunft standardisiert

Applikationssoftwarekomponenten sind *gekapselt* in ein *standardisiertes* AUTOSAR *interface*. Dieses Interface muß für die externe Kommunikation benutzt werden.

Die AUTOSAR-RTE (Runtime Environment) ist die *zentrale Kommunikationsplattform*. Alle Kommunikation zwischen Software Komponenten, sei es auf dem gleichen Steuergerät, oder auf anderen oder mit Sensoren und Aktuatoren (I/O) fließt durch die RTE.



Zusammenfassung – Modellbasierte Entwicklung ist effizient und liefert effiziente Software

Mehr als **60+ Millionen** Fahrzeuge, deren Steuergerätesoftware modellbasiert mit ASCET entwickelt wurde, sprechen für:

- Code **Qualität** die **höher** ist als Handcodierung
- Eine **sehr effiziente** Codeerzeugung durch ASCET
- Leichte Konfigurierbarkeit der Eigenschaften des automatisch erzeugten Codes

ETAS

LiveDevices
ETAS Group

Vetronix
ETAS Group



Danke für Ihre Aufmerksamkeit!