



**PORSCHE**

**26. Treffen der GI-Arbeitsgruppe „Test, Analyse und Verifikation von Software“  
Stuttgart, den 06.12.2007**

**Systematischer Test modellbasiert entwickelter  
Steuergeräte**

Dipl.-Ing. Matthias Wiese\*, Prof. Dr. Hans-Christian Reuss\*\*, Dr. Rüdiger Dorn\*, Dr. Rolf Zöller\*

\* Dr. Ing. h.c. F. Porsche AG

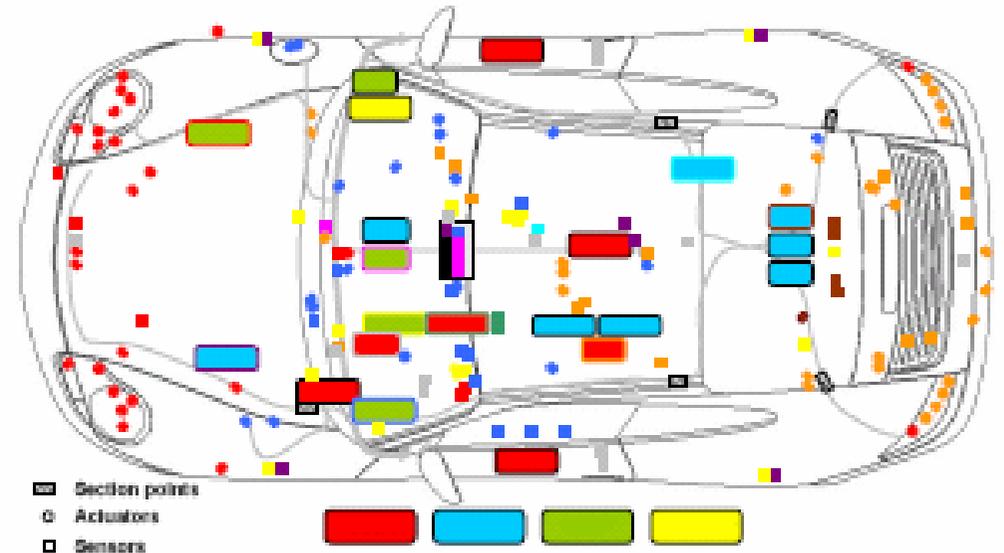
\*\* Forschungsinstitut für Kraftfahrwesen und Fahrzeugmotoren Stuttgart

## Gliederung

- **Modellbasierte Steuergeräteentwicklung beim OEM**
- Absicherung der modellbasierten Entwicklung
- Agile Ansätze in der modellbasierten Steuergeräteentwicklung
- Ein Top-Down-Ansatz zur systematischen Testspezifikation
- Zusammenfassung

## Hohe Komplexität führt zu Fehlern

- Schäden an der Fahrzeugelektrik/elektronik waren laut ADAC-Pannenstatistik häufigste Pannenursachen im Jahr 2006.
- Heutzutage enthalten Kraftfahrzeuge der Premiumklasse bis zu 70 Steuergeräte.
- Die hohe Komplexität führt zu neue Fehlerbildern (systematische Fehler lösen verschleißbedingte Fehler ab).
- Es ist sinnvoll die hohe Komplexität zu reduzieren.

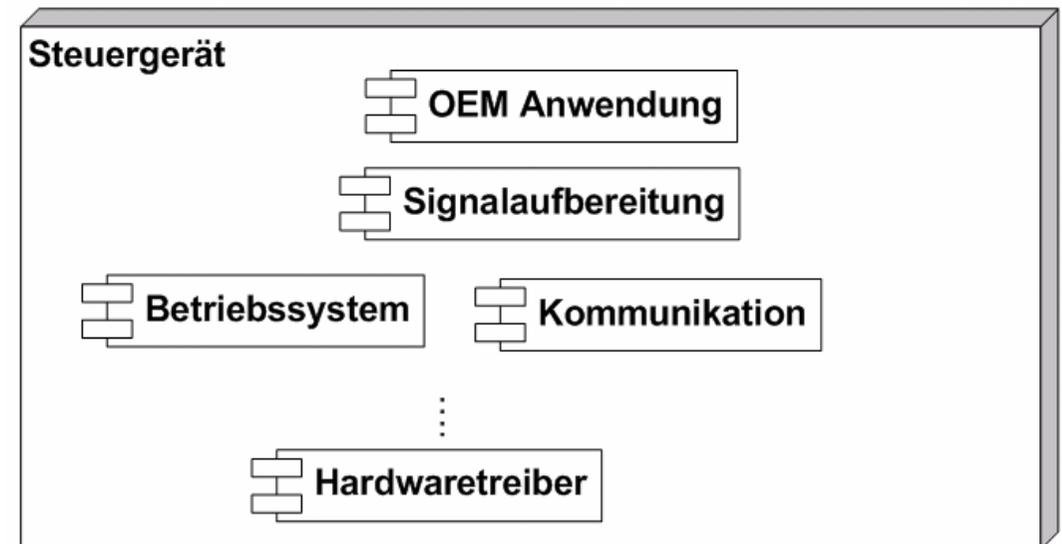


### Vernetzte Fahrzeugelektrik/Elektronik eines Porsche Sportwagens

## Verschiedene Softwarekomponenten im Steuergerät

- Die Steuergerätesoftware besteht aus mehreren Komponenten, die von verschiedenen Kooperationspartnern stammen können (Divide et impera).
- Der OEM kann sein Know-how durch Einbringen eigener Softwareanteile schützen.
- Die Softwarekomponenten können vom OEM in verschiedenen Projekten wiederverwendet werden.
- AUTOSAR unterstützt das Einbringen von eigenen Softwarekomponenten durch den OEM (Plug-and-Play über definierte Schnittstellen)

### Steuergerät mit verschiedenen Softwarekomponenten



## Modellbasierte Entwicklung von Softwarekomponenten

Vorteile für den OEM durch modellbasierte Entwicklung von Applikationssoftwarekomponenten:

- Ausführbare Modelle ermöglichen eine frühe Validierung der Funktion (Frontloading). Dies führt zu Kostenreduktion, da Fehler früh im Entwicklungsprozess beseitigt werden können.
- Durch den Top-Down-Ansatz bei der modellbasierten Entwicklung wird die steigende Komplexität der Elektronik im Kraftfahrzeug beherrschbar (hardwareunabhängige Entwicklung).
- Der OEM kann Funktionen in einer domänenspezifischen Sprache selbst entwickeln und somit Alleinstellungsmerkmale eigenständig realisieren.
- Das Modell gibt einen Einblick (White-Box), was für Coverage-Messungen hilfreich ist.
- Das Funktionsmodell ermöglicht eine Spezifikation ohne großen Interpretationsspielraum (im Gegensatz zu textuellen Spezifikationen).

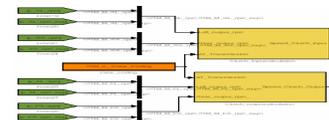
### Assembler

```
push ds es ax
mov ax, 0A000h
mov es, ax
```

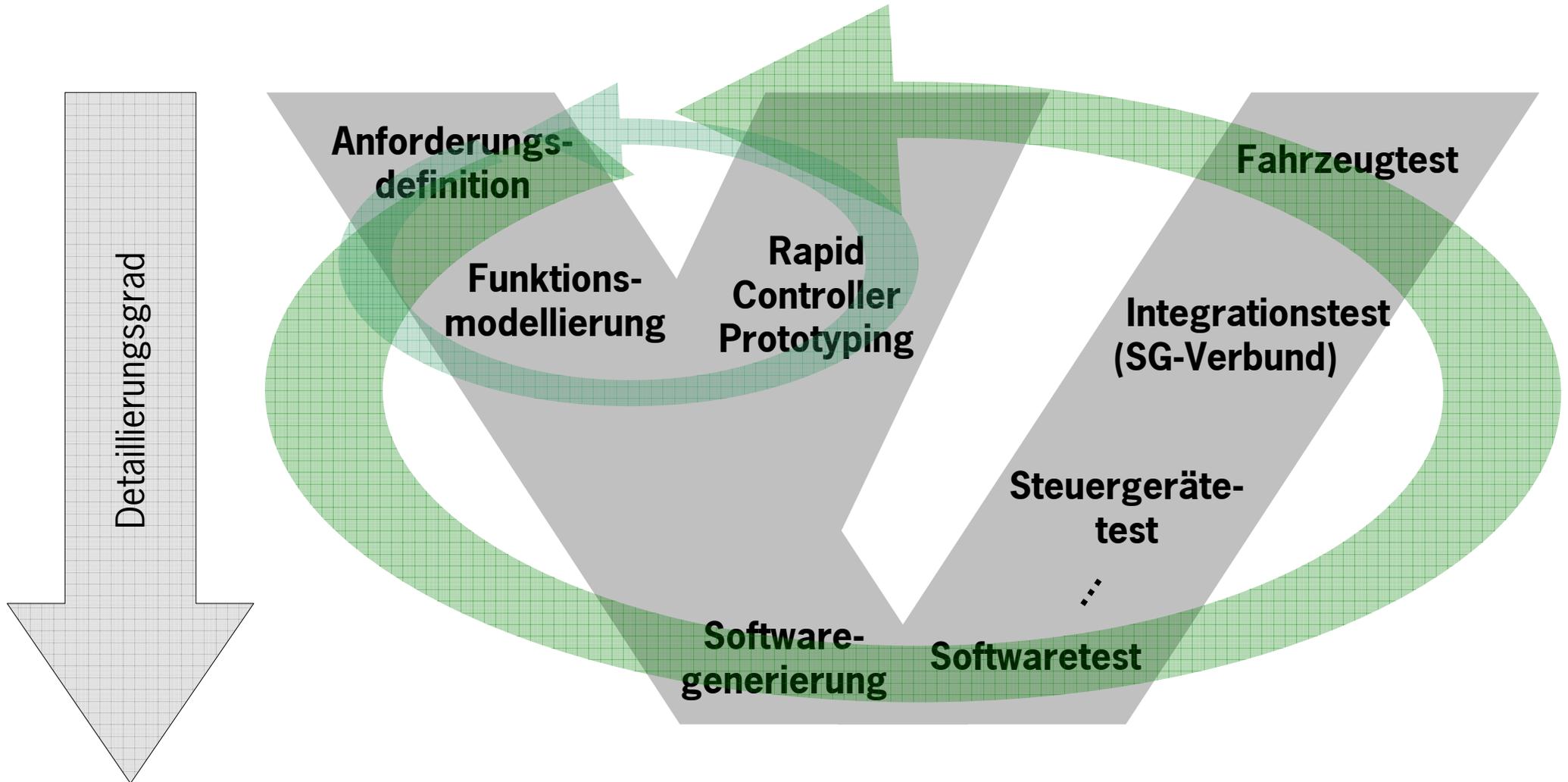
### C-Code

```
void receive(void)
{ unsigned char test[32];
  rf12_rxdata(test, 32);
}
```

### Modell



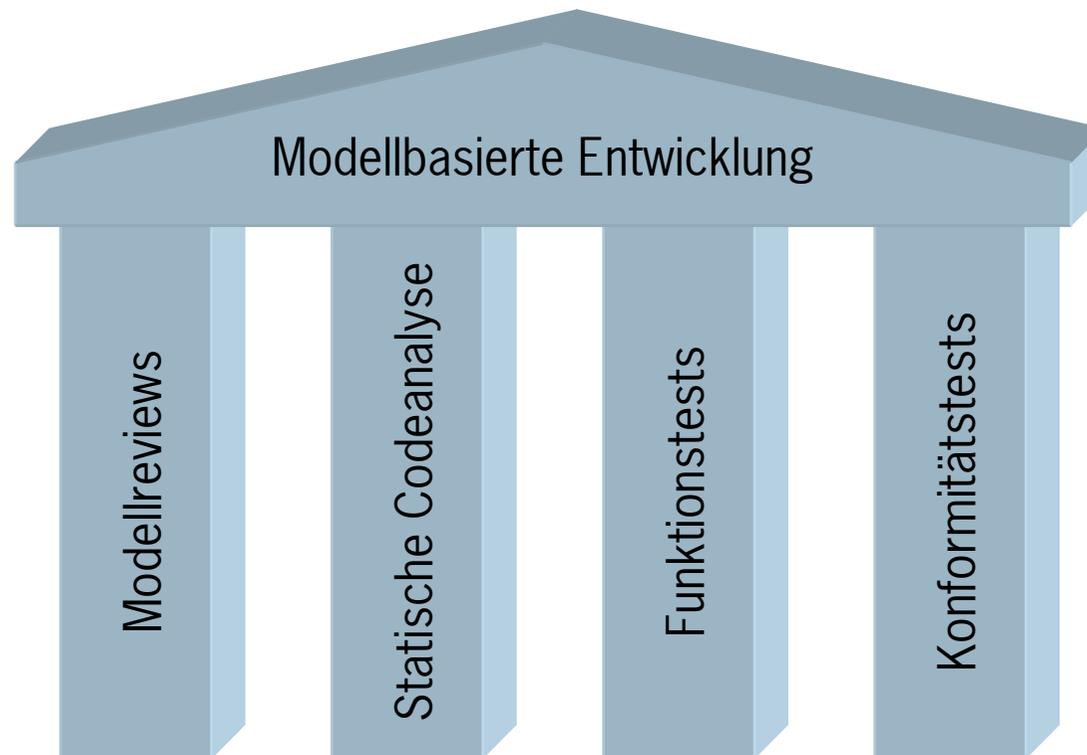
# Prinzip der modellbasierten Softwareentwicklung beim OEM



## Gliederung

- Modellbasierte Steuergeräteentwicklung beim OEM
- **Absicherung der modellbasierten Entwicklung**
- Agile Ansätze in der modellbasierten Steuergeräteentwicklung
- Ein Top-Down-Ansatz zur systematischen Testspezifikation
- Zusammenfassung

## Absicherung der modellbasierten Softwareentwicklung



Die modellbasierte Software-Entwicklung stützt sich auf **statische und dynamische Testverfahren**.

## Qualitätssicherung durch statische Tests

### Modell-Reviews:

- Die Modelle werden bezüglich Funktionalität und Codegenerierbarkeit überprüft.
- Prüfung auf Einhaltung der konstruktiven Qualitätsmaßnahmen (Dokumentations- und Modellierungsrichtlinien)
- Ziel: Qualitätssicherung der Funktionsmodelle



### Statische Codeanalyse:

- Der generierte Quellcode wird mittels Analysewerkzeugen auf mögliche Laufzeitfehler untersucht (Overflow, Underflow, Zero-Division usw.).
- Ziel: Abfangen von Fehlern, die bei der Codeerzeugung auftreten. Dies sind vorwiegend Skalierungsfehler.

## Funktionstests am Modell mittels Rapid-Prototyping

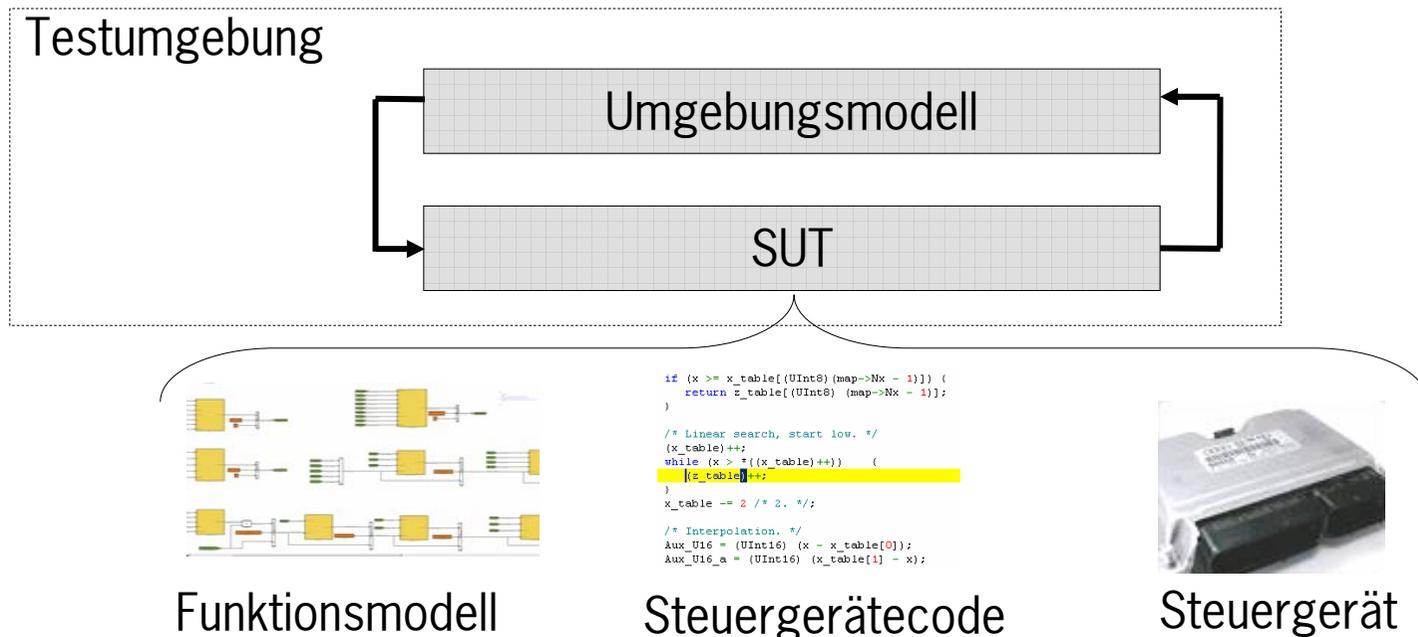
- Je früher Fehler beseitigt werden desto weniger Kosten entstehen (vgl. Boehm).
- Beim Rapid-Prototyping im Fahrzeug werden Softwarekomponenten „freigeschnitten“ und mithilfe von Bypassing-Techniken auf einem Universalsteuergerät ausgeführt.
- Dadurch können Funktionen direkt von den Modellieren erfahren werden.
- Im Fahrzeug werden textuelle Prüfanweisungen abgearbeitet.

Rapid-Prototyping:



## Funktionstests am Modell

- Eine In-the-Loop-Simulationsumgebung ermöglicht das Nachstellen von Extremsituationen.
- Die White-Box-Tests am Modell bzw. der Softwarekomponente ermöglichen Coverage-Messungen.
- Die Funktionsentwickler testen ihre Funktionsmodule vor der Integration in das Gesamtmodell.

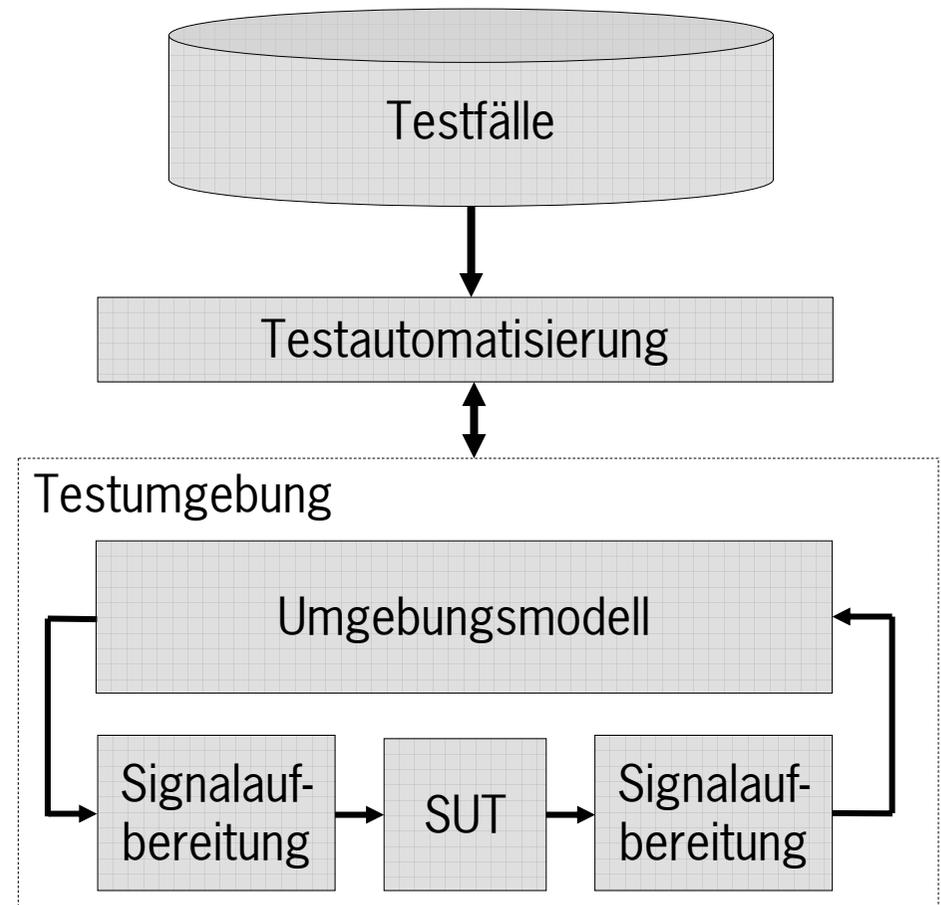


# Automatisierte Funktionstests mit X-in-the-Loop-Simulation

- Eine durchgängige Testumgebung ermöglicht die Wiederverwendung von automatisierten Tests auf verschiedenen Teststufen (MiL, SiL, HiL).

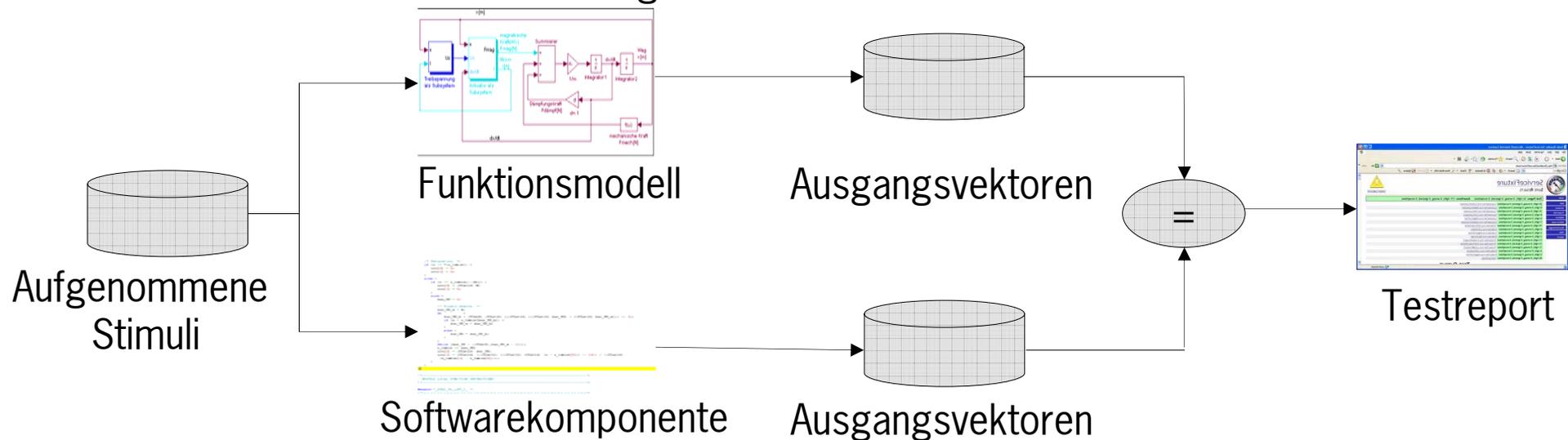
Vorraussetzungen:

- Die Signalaufbereitung muss an das SUT angepasst werden.
- Die Testumgebung und Testautomatisierung sind zu synchronisieren (Echtzeit).



## Konformitätstests

- Mit Back-to-Back-Tests soll die Verhaltenskonformität zwischen dem Modell und der generierten Software überprüft werden.
- Dazu kommen Testvektoren zum Einsatz (aus Fahrversuchen und synthetisch erzeugt)
- Tests werden hierbei im Open-Loop-Verfahren durchgeführt.
- Coverage-Messungen liefern Rückmeldung bezüglich nicht abgedeckter Funktionen, Modellteile bzw. Code. Daraus werden Maßnahmen abgeleitet.



## Gliederung

- Modellbasierte Steuergeräteentwicklung beim OEM
- Absicherung der modellbasierten Entwicklung
- **Agile Ansätze in der modellbasierten Steuergeräteentwicklung**
- Ein Top-Down-Ansatz zur systematischen Testspezifikation
- Zusammenfassung

## **Best Practices agiler Methoden beim Test modellbasiert entwickelter Steuergeräte**

### **Der dargestellte Entwicklungsprozess enthält bereits Elemente agiler Methoden:**

- Die kurzen Iterationszyklen in Form von kurzen Verbund-Releases und Rapid-Controller-Prototyping liefern schnelles Feedback.
- Die Priorität liegt auf funktionsfähigen Artefakten (Modell, Steuergerätesoftware, Steuergerät).
- Die textuellen Prüfanweisungen ähneln stark den User Stories aus dem eXtreme Programming.

### **Herausforderungen für die Tester:**

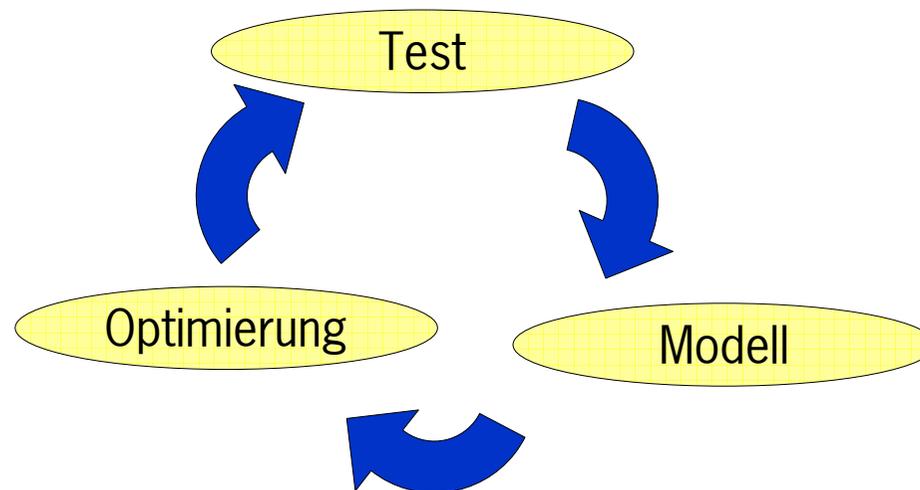
- Da das Funktionsmodell häufig als ausführbares Lastenheft angesehen wird, erfolgt die textuelle Beschreibung der funktionalen Anforderungen u.U. weniger detailliert.
- Wogegen werden die Modelle dann getestet (Testreferenz/Testorakel)?

### **Vorschlag:**

- Best Practices agiler Methoden sollen bei der modellbasierten Steuergeräteentwicklung bewusst zum Einsatz kommen.

## Einsatz von testgetriebener Entwicklung

- Anwendung von Prinzipien der testgetriebenen Entwicklung auf den Prozess der modellbasierten Steuergeräteentwicklung
- Automatisierte Regressionstest sichern jedes neue Release ab
- Aus dem akkumulierten Expertenwissen der Entwickler wird eine Testreferenz extrahiert.
- Die Testfälle treten an die Stelle einer ausführbaren Spezifikation.



Testgetriebener Entwicklungsansatz für modellbasierte Steuergeräteentwicklung

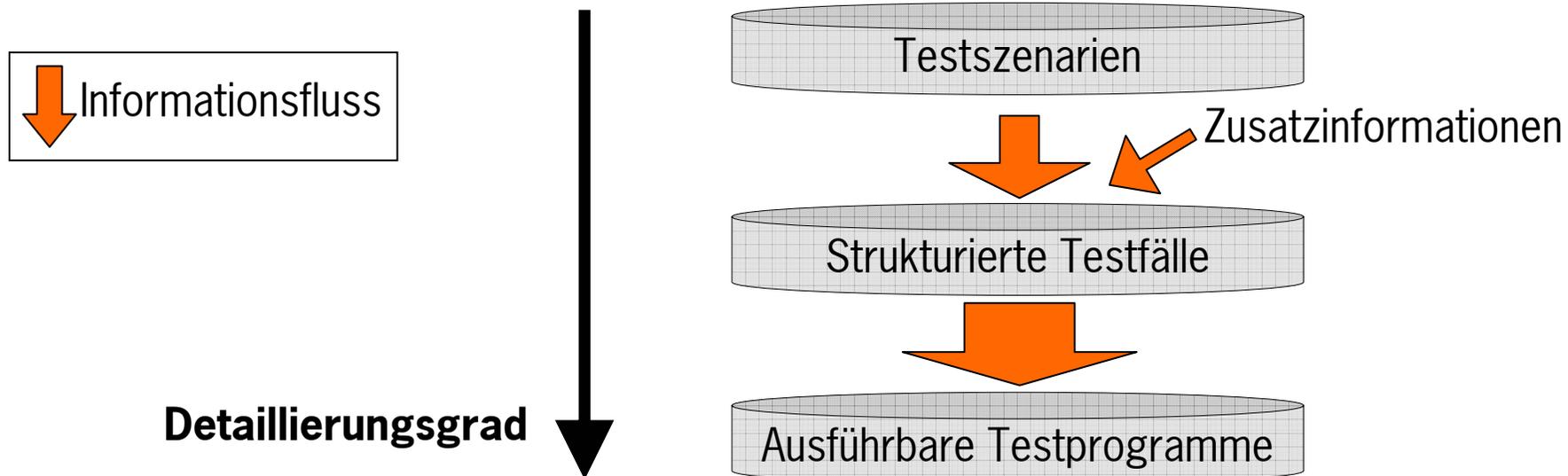
## Gliederung

- Modellbasierte Steuergeräteentwicklung beim OEM
- Absicherung der modellbasierten Entwicklung
- Agile Ansätze in der modellbasierten Steuergeräteentwicklung
- **Ein Top-Down-Ansatz zur systematischen Testspezifikation**
- Zusammenfassung

# Systematische Testspezifikation

## Top-Down-Vorgehen zur Testspezifikation:

1. Testszenarien werden von den Funktionsentwicklern textuell beschrieben (Prüfanweisungen).
2. Diese Testszenarien werden durch die Testingenieure zu Testfällen verfeinert (evtl. mit Entwicklern).
3. Aus den Testfällen werden ausführbare Testprogramme generiert (analog zur modellbasierten Entwicklung)



## Von User Stories zu Testfällen

### Vorbedingungen:

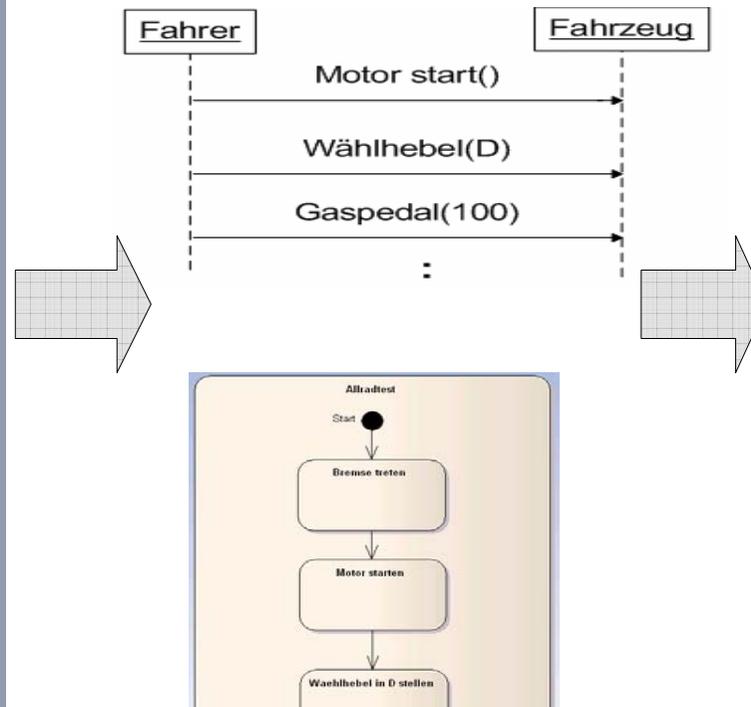
Das Fahrzeug steht mit laufendem Motor auf einer Ebene.  
Die Hinterachse befindet sich auf Untergrund mit niedrigen Reibwerten.

### Prüfablauf:

Der Fahrer gibt soviel Gas, dass es zum Durchdrehen der Hinterräder kommt.

### Sollreaktion:

Es wird Moment auf die Vorderräder gegeben, so dass das Fahrzeug beschleunigt.



```
# function definition
def slipTest(signal_info_local):
    SG_INIT(signal_info_local['signalFile'])
    :
```

Prüfanweisung

Testfallmodellierung

Testskript

## Gliederung

- Modellbasierte Steuergeräteentwicklung beim OEM
- Absicherung der modellbasierten Entwicklung
- Agile Ansätze in der modellbasierten Steuergeräteentwicklung
- Ein Top-Down-Ansatz zur systematischen Testspezifikation
- **Zusammenfassung**

## Kernaussagen

- Die modellbasierte Entwicklung ermöglicht es dem Funktionsentwickler beim OEM Softwarekomponenten für Steuergeräte in einer **domänenspezifischen Sprache** zu beschreiben und somit funktionale Alleinstellungsmerkmale eigenständig zu realisieren.
- Die Teststrategie zur Absicherung der modellbasierten Entwicklung kombiniert **statische und dynamische Testverfahren**.
- **Best Practices agiler Methoden** lassen sich auf den modellbasierten Entwicklungsprozess von Steuergeräten abbilden.
- **Tests** können an die Stelle einer **ausführbaren Spezifikation** treten.
- Ein **systematischer Top-Down-Ansatz zur Testspezifikation** – analog zur modellbasierten Entwicklung – erscheint zielführend.

# PORSCHE



**Vielen Dank für ihre  
Aufmerksamkeit**

## Backup

# Prinzipien agiler Softwareentwicklung

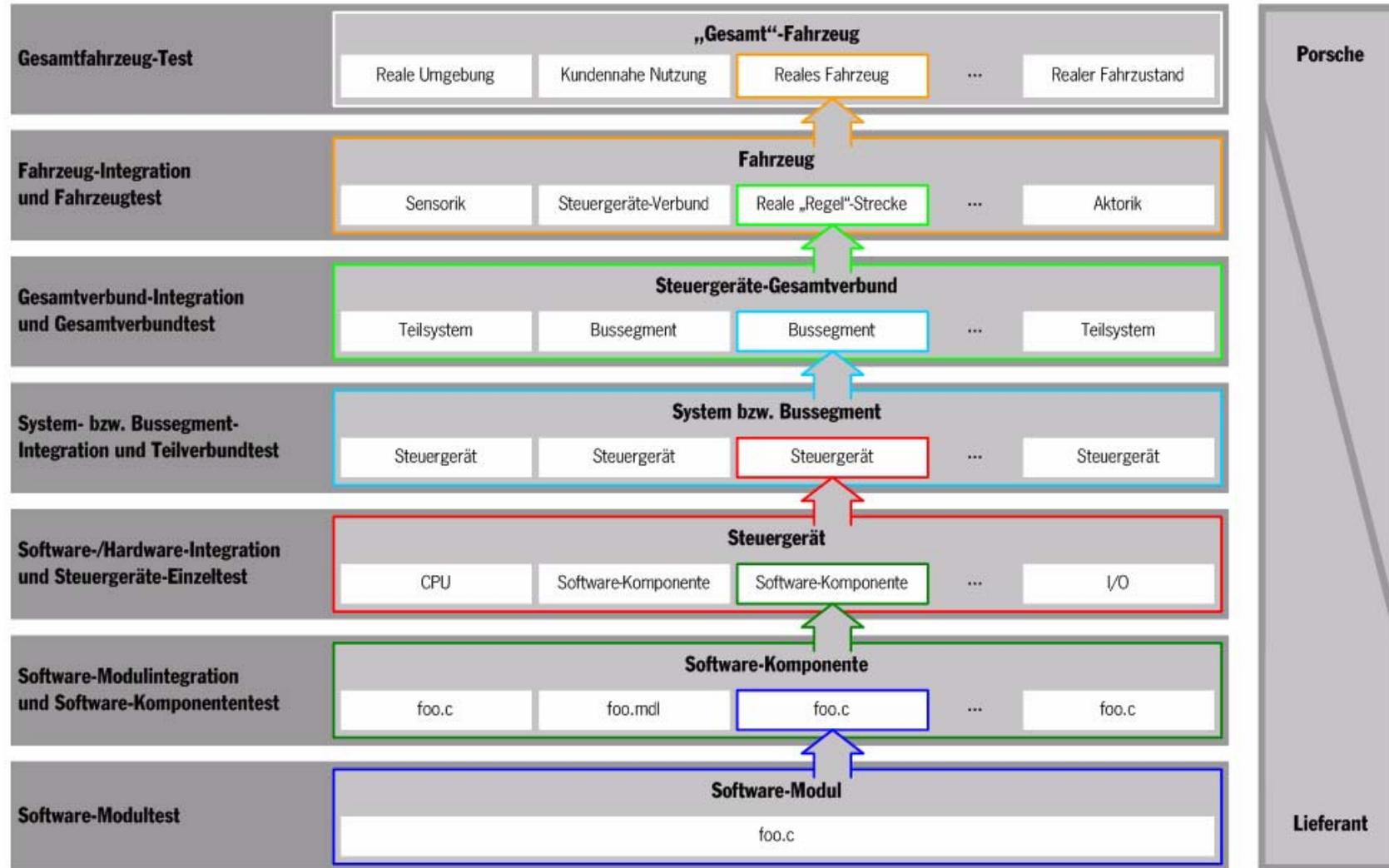
## Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

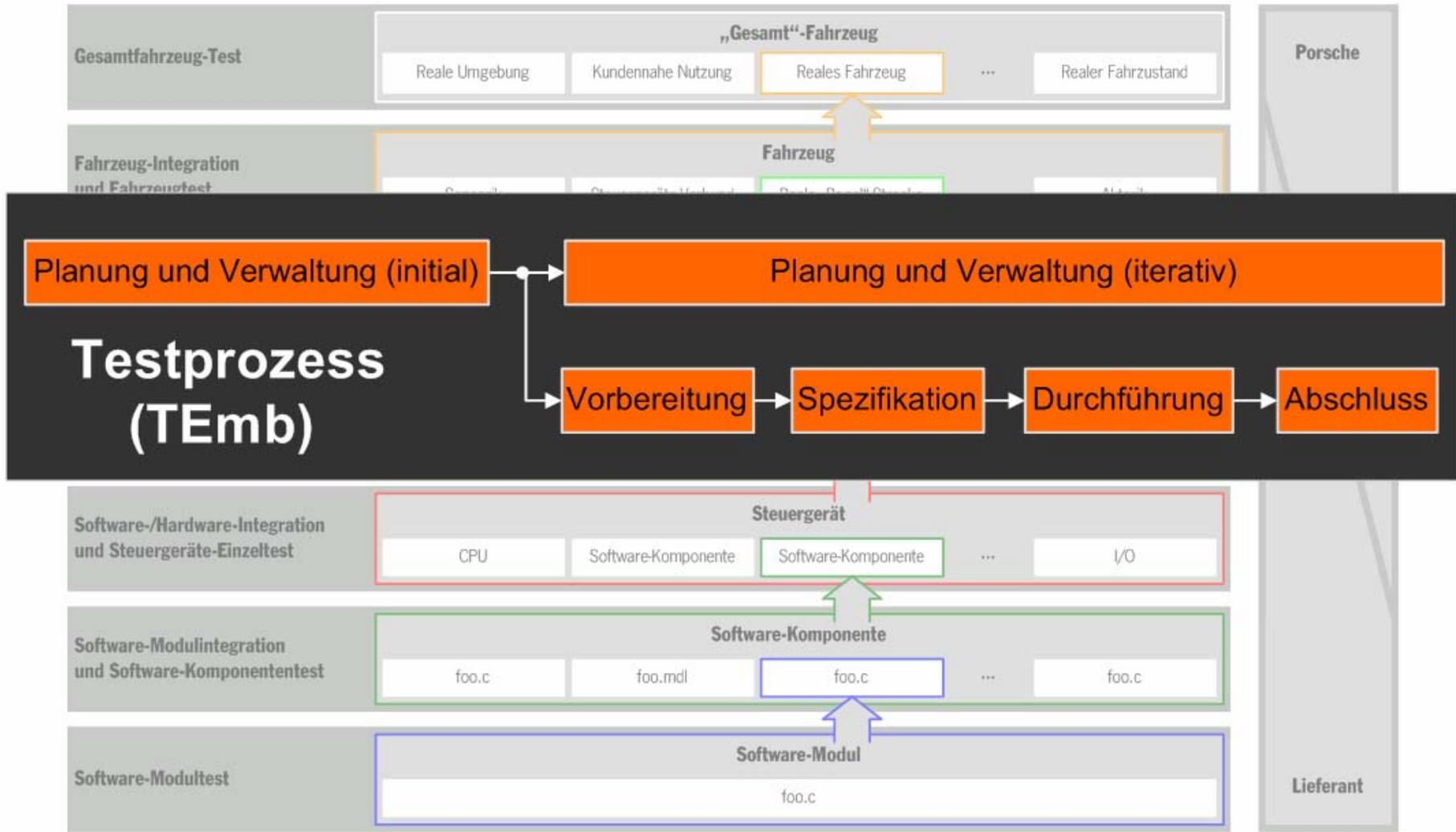
**Individuals and interactions** over processes and tools  
**Working software** over comprehensive documentation  
**Customer collaboration** over contract negotiation  
**Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

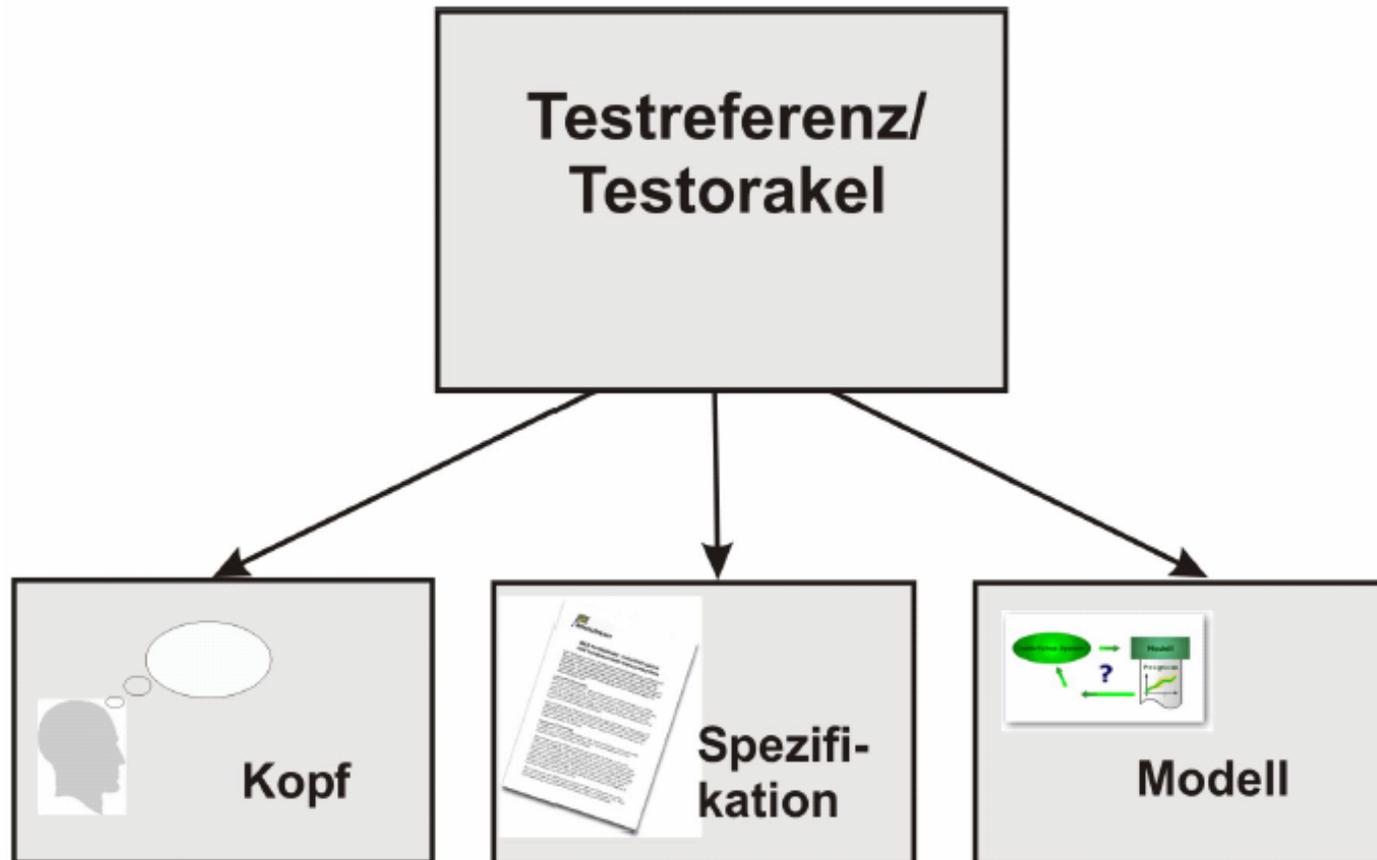
## Teststrategie und Testprozess nach TPI/TEmb



## Teststrategie und Testprozess nach TPI/TEmb



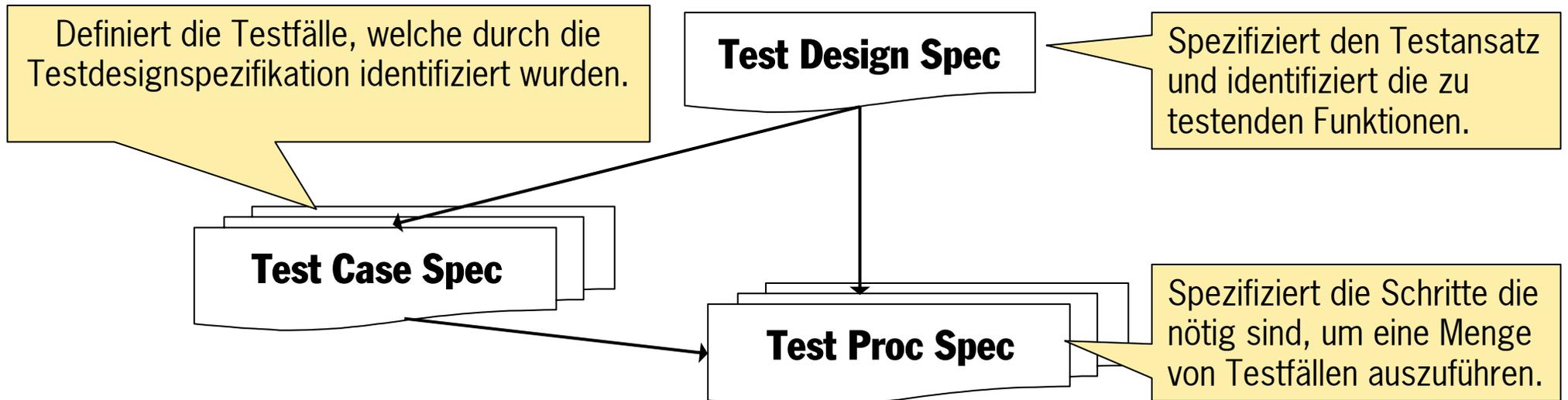
## Testreferenz



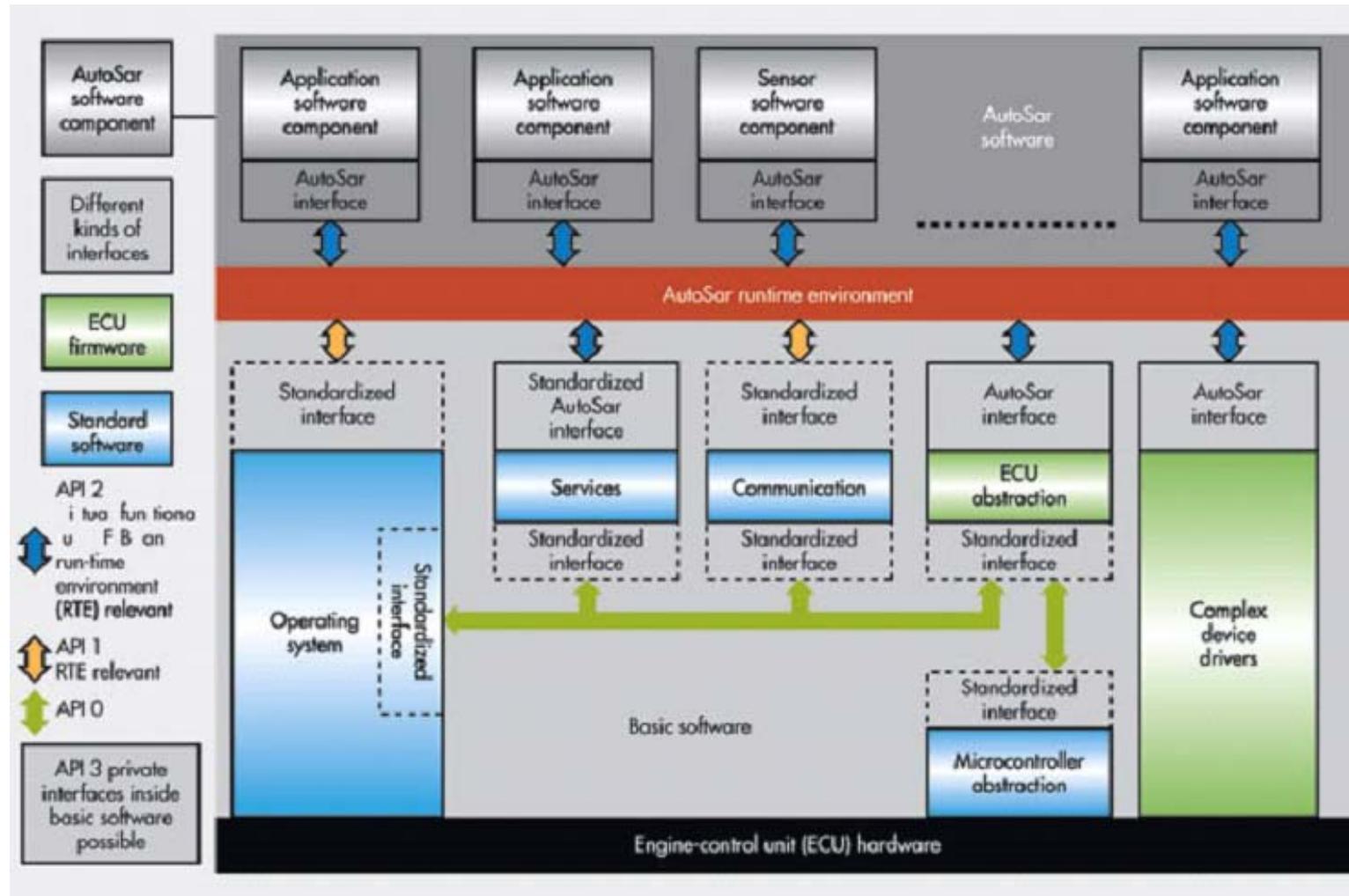
Quelle: M. Horstmann, Verflechtung von Test und Entwurf für eine verlässliche Entwicklung eingebetteter Systeme im Automobilbereich, 2005

# Systematische Testspezifikation nach IEEE 829

- Testfälle sollen nach einem Top-Down-Ansatz vom abstrakten zum konkreten kontinuierlich verfeinert werden



## Autosar ermöglicht dem OEM die Integration eigener SW-Cs



## Funktionstests mittels Rapid-Controller-Prototyping

