

# Vorlesung „Embedded Software-Engineering im Bereich Automotive“

Technische Universität Dresden, Fakultät Informatik,  
Professur Softwaretechnologie

Sommersemester 2010

Dr. rer. nat. Bernhard Hohlfeld

[bernhard.hohlfeld@daad-alumni.de](mailto:bernhard.hohlfeld@daad-alumni.de)

1. Motivation und Überblick
2. Grundlagen Fahrzeugentwicklung, KFZ-Elektronik und Software
3. Übersicht Automotive Elektrik/Elektronik-Entwicklung (E/E)
- 4. Kernprozess zur Entwicklung von elektronischen Systemen und Software**
5. Unterstützungsprozesse für die Embedded Software Entwicklung
6. Beispiele aus der Praxis
7. Wichtige Normen/Standards/Empfehlungen für die Embedded Software Entwicklung

## 4. Kernprozess zur Entwicklung von elektronischen Systemen und Software



1. Grundbegriffe

2. Entwicklungsobjekt: Kombiinstrument

3. Analyse und Spezifikation der Benutzeranforderungen

4. Analyse und Spezifikation der technischen Anforderungen

5. Analyse und Spezifikation der Software-Anforderungen

6. Spezifikation der Software-Komponenten

**7. Design und Implementierung der Software-Komponenten**

8. Test der Software-Komponenten

9. Integration der Software-Komponenten

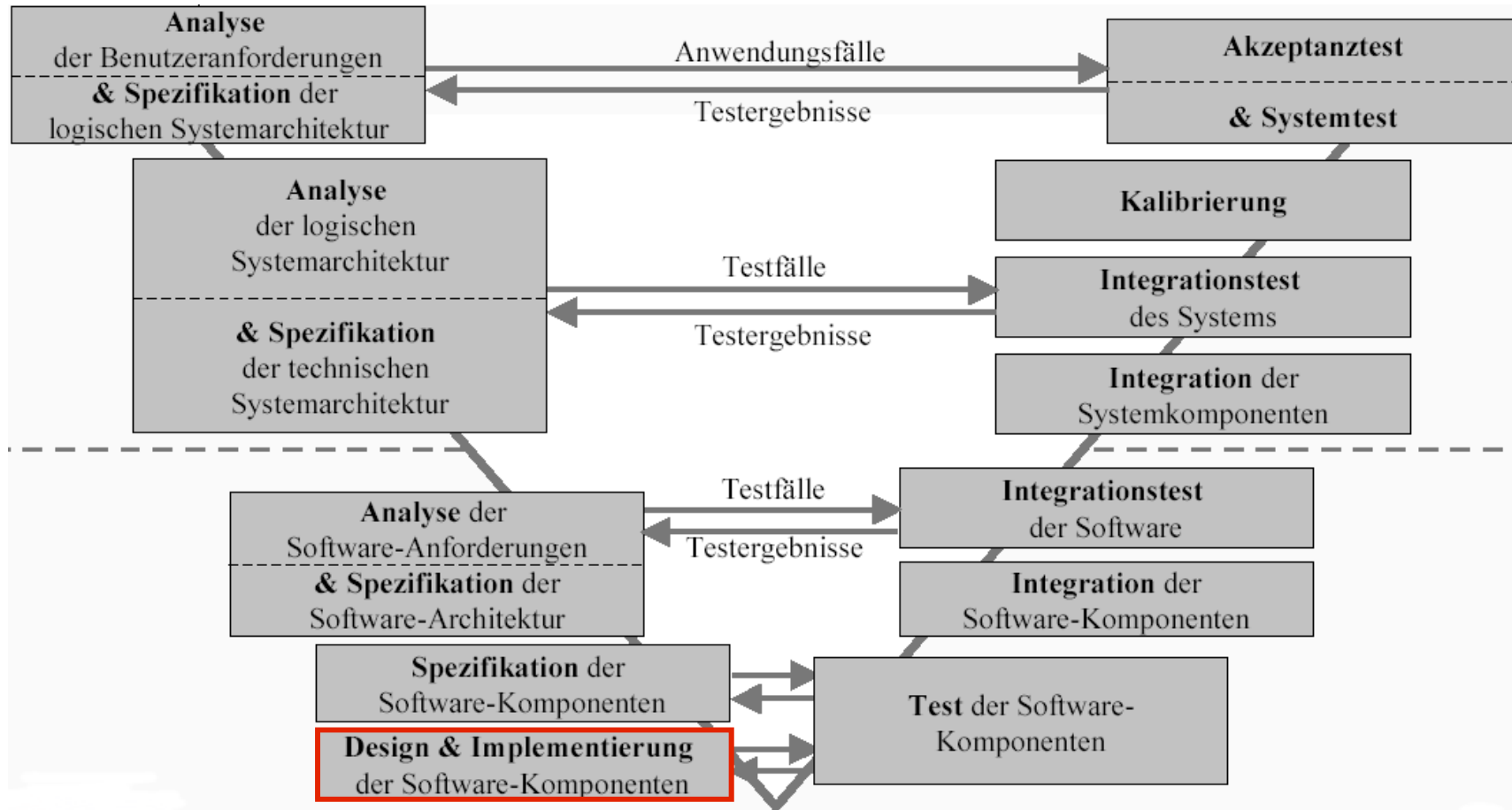
10. Integrationstest der Software-Komponenten

11. Integration der System-Komponenten

12. Integrationstest des Systems

13. Kalibrierung

14. Akzeptanz- und Systemtest

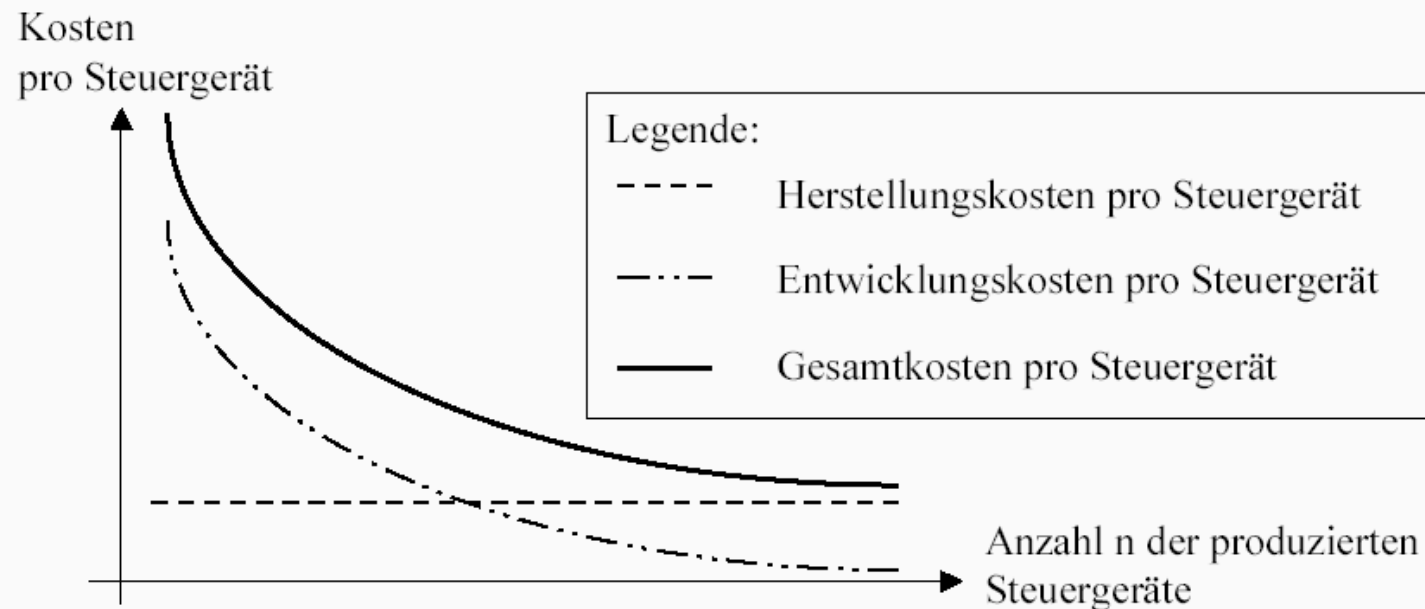


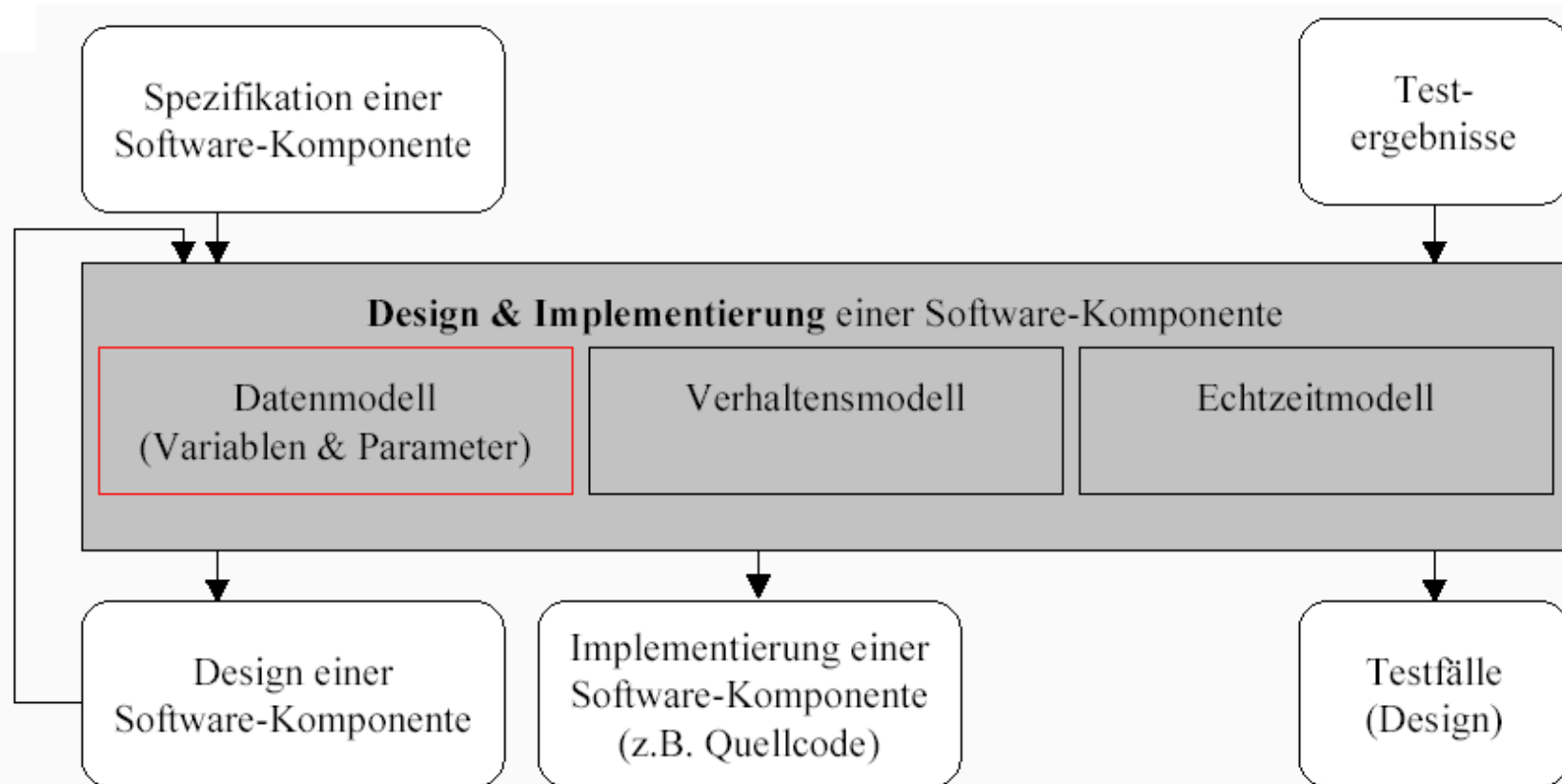
## Berücksichtigung der geforderten nichtfunktionalen Produkteigenschaften

- ◆ Unterscheidung zwischen Programm- und Datenstand
- ◆ Beschränkung der Hardware-Ressourcen

Beispiel: Kostenschranken bei Steuergeräten

Gesamtkosten pro Steuergerät  $\approx$  (Entwicklungskosten+ Herstellungskosten) / n

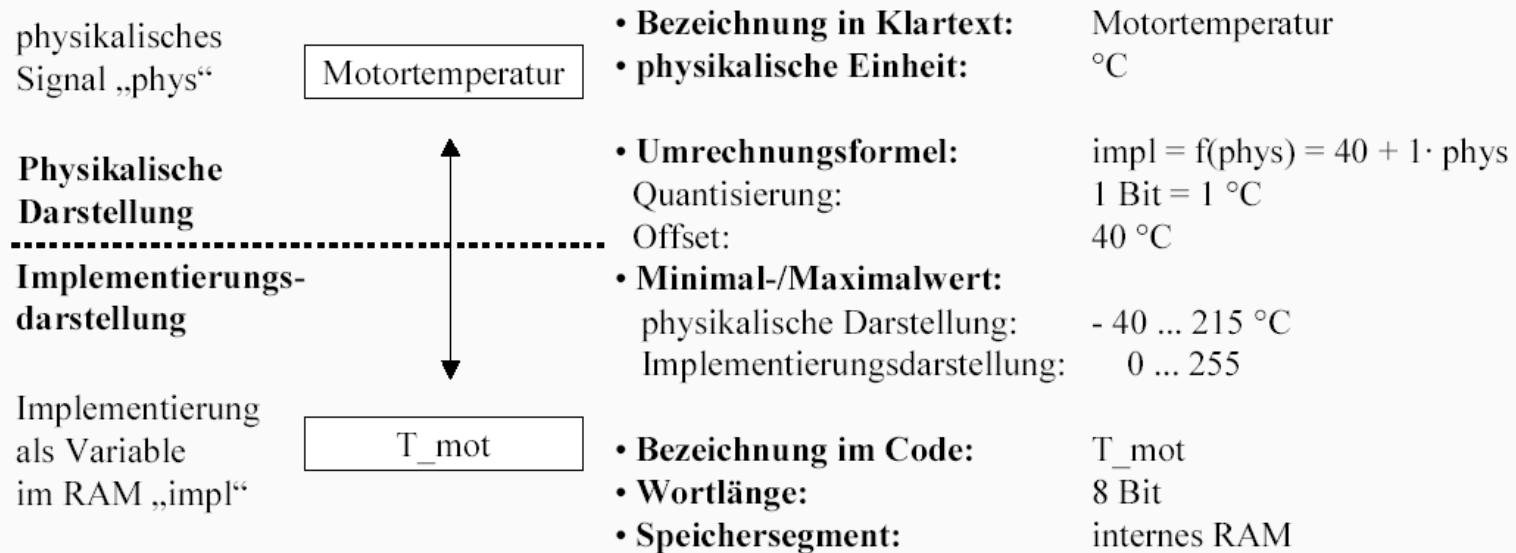


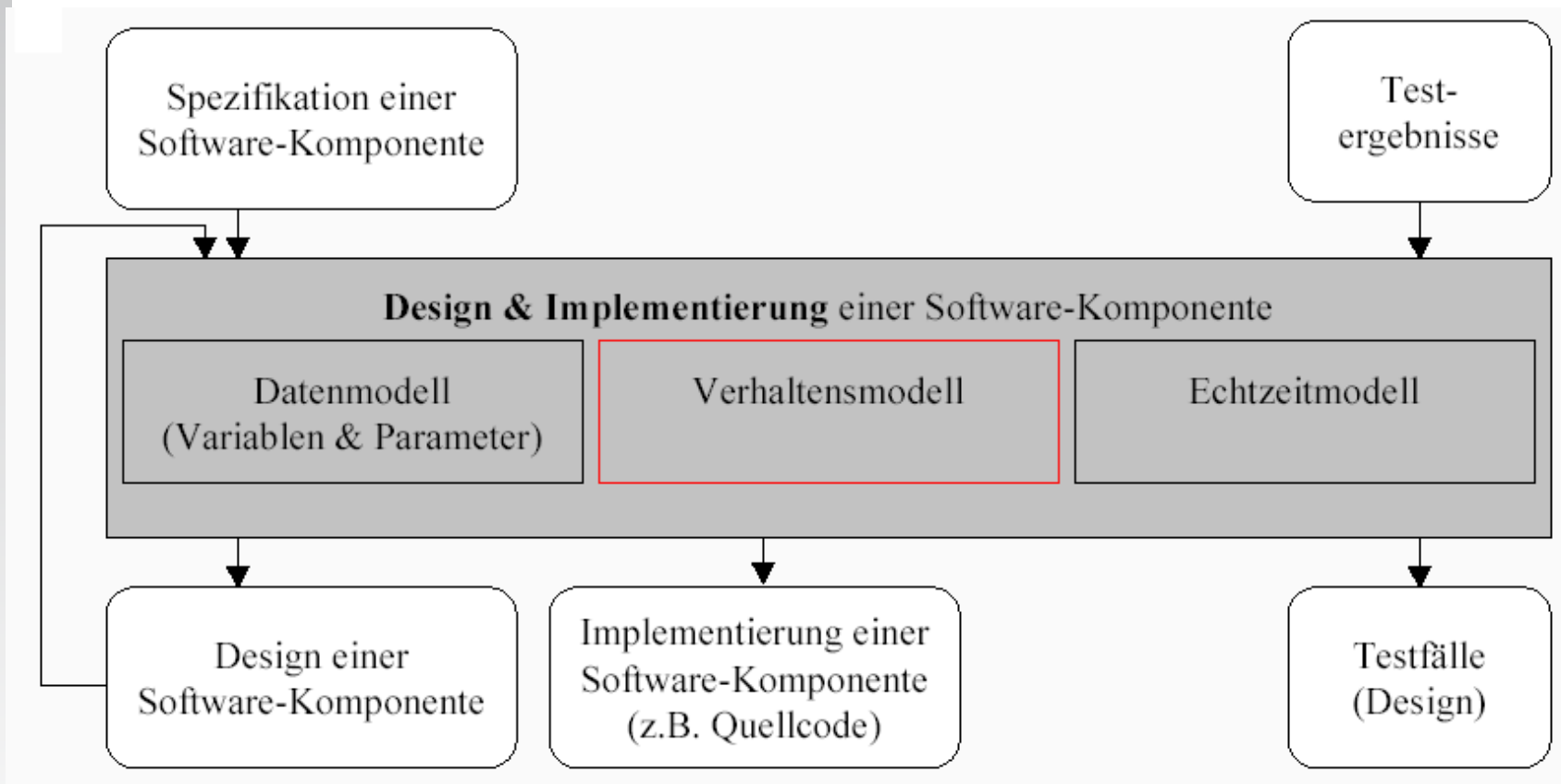


## Design und Implementierung des Datenmodells

- ◆ Unterscheidung zwischen Variablen und durch das Programm nicht veränderbaren Parametern
- ◆ Design-Entscheidungen: prozessorinterne Darstellung und Speichersegment für die Ablage (RAM bzw. ROM)

### Abbildung der physikalischen Spezifikation auf die Implementierung



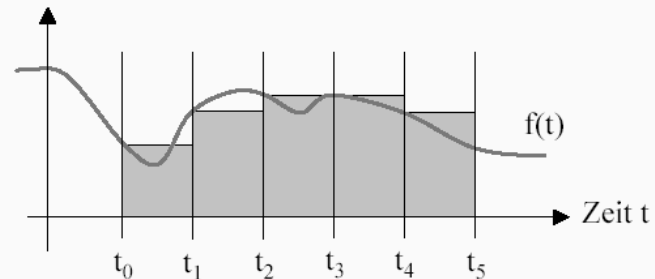




## Genauigkeit

- ◆ Fehler in den Eingabedaten
- ◆ Rundungsfehler
- ◆ Approximationsfehler

## Integrationsverfahren nach Euler



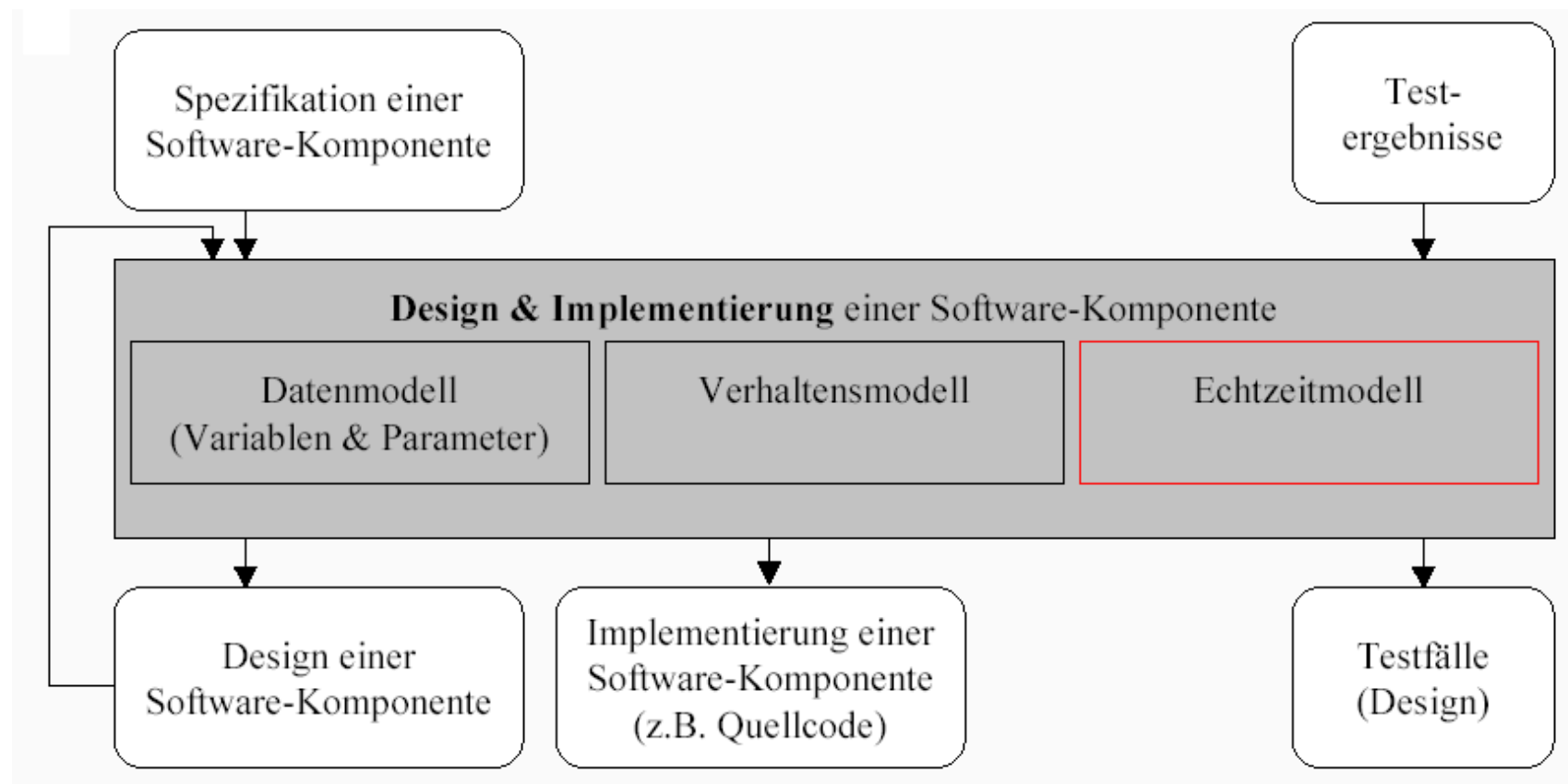
$$F(t_n) = \int_{t_0}^{t_n} f(t) dt \quad \text{durch}$$

$$F^*(t_n) = \sum_{i=0}^{n-1} (t_{i+1} - t_i) * f(t_i) \quad \text{approximiert}$$

$$dT_i = (t_{i+1} - t_i) \quad (\text{Schrittweite})$$

Inkrementelle Berechnung:

$$F^*(t_{i+1}) = F^*(t_i) + dT_i * f(t_i)$$



## **Design und Implementierung des Echtzeitmodells**

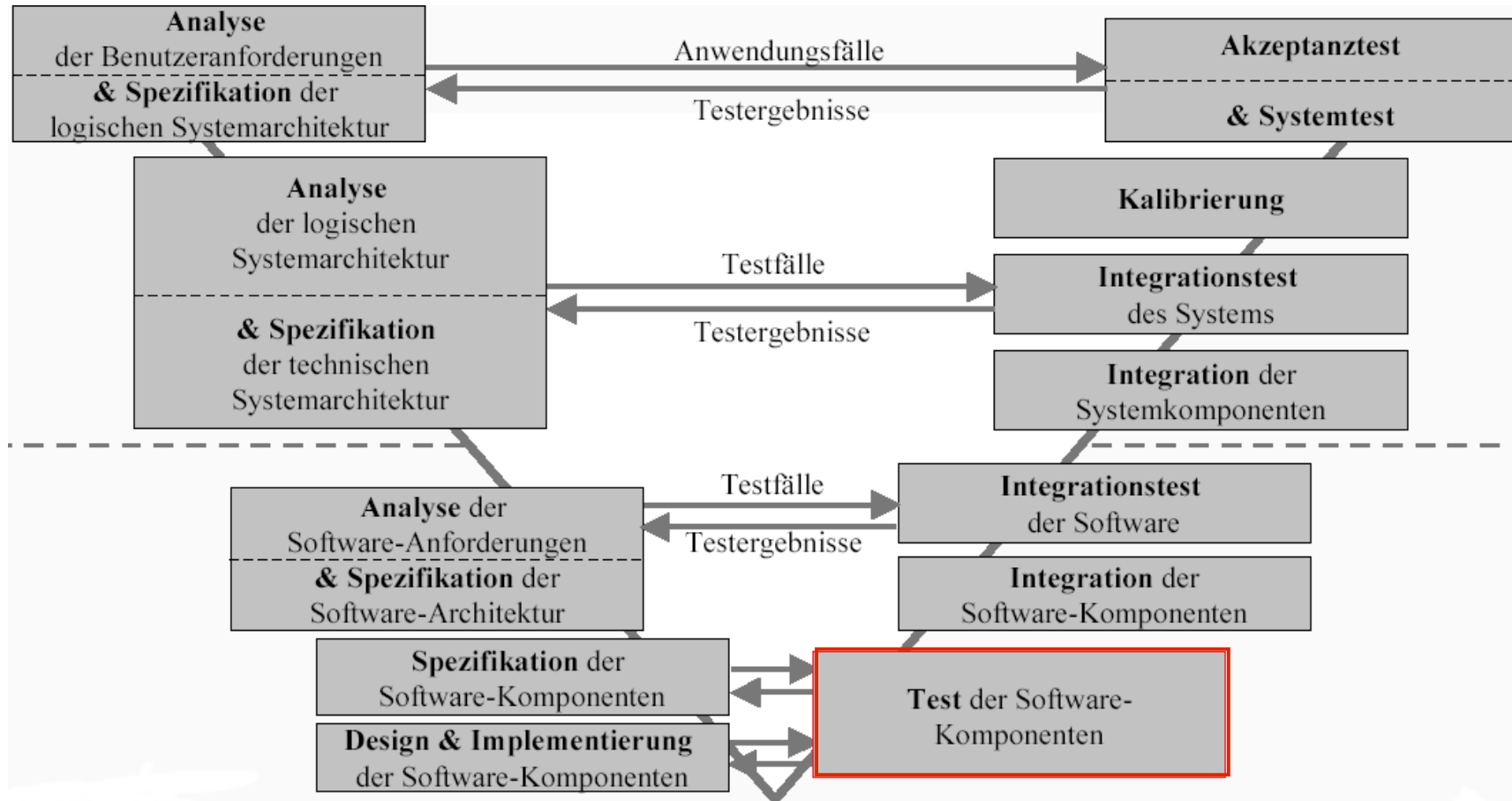
- ◆ Hardware- und Software-Interrupt-System des Mikrocontrollers berücksichtigen
- ◆ Festlegung der Konfiguration des Echtzeitbetriebssystems

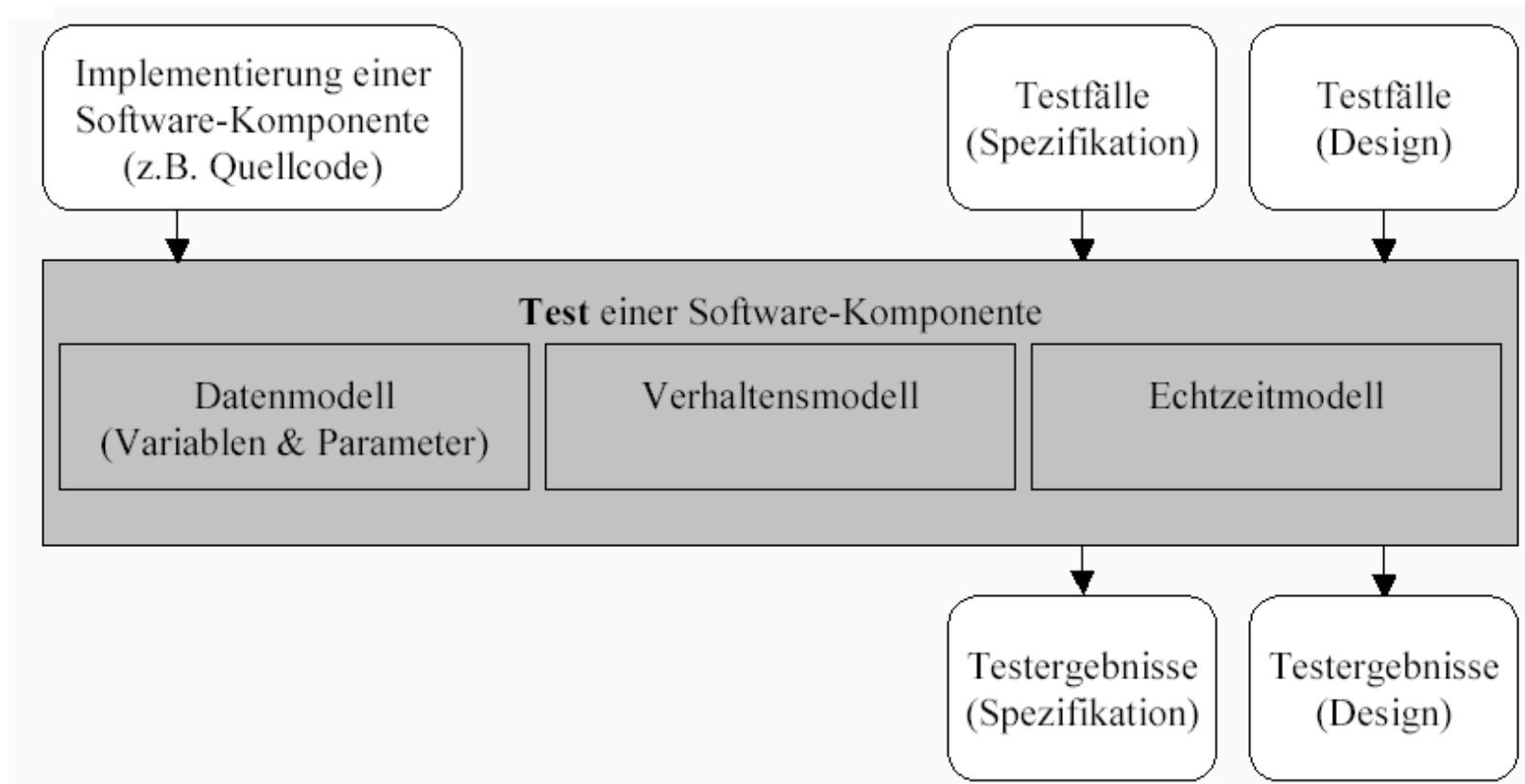
## 4. Kernprozess zur Entwicklung von elektronischen Systemen und Software



1. Grundbegriffe
2. Entwicklungsobjekt: Kombiinstrument
3. Analyse und Spezifikation der Benutzeranforderungen
4. Analyse und Spezifikation der technischen Anforderungen
5. Analyse und Spezifikation der Software-Anforderungen
6. Spezifikation der Software-Komponenten
7. Design und Implementierung der Software-Komponenten
- 8. Test der Software-Komponenten**
9. Integration der Software-Komponenten
10. Integrationstest der Software-Komponenten
11. Integration der System-Komponenten
12. Integrationstest des Systems
13. Kalibrierung
14. Akzeptanz- und Systemtest

# Test der Software-Komponenten





## Übersicht über Maßnahmen zur Qualitätssicherung von Software

### Verifikation

#### Statische Techniken

##### Review

Walkthrough, Fagan-Inspektion,  
Code-Inspektion, Peer-Review, ...

##### Analyse

Statische Analyse, Formale Prüfung,  
Kontroll- und Datenfluss, ...

#### Dynamischer Test Komponenten-/Integrationstest

##### Black-Box-Test

Funktionale Leistungsfähigkeit,  
Stress, Grenzwert, Fehlererwartung, ...

##### White-Box-Test

Struktur, Pfad, Zweig, Bedingung,  
Abdeckung, ...

### Validation

#### Animation

Formale Spezifikation

Modellierung

Simulation

Rapid Prototyping, ...

#### Systemtest/Akzeptanztest

Funktionale Leistungsfähigkeit

Stresstests, Grenzwerttests,

Fehlererwartungstests, Ursache-

Wirkungs-Graph, Äquivalenz-

klassentests, ...

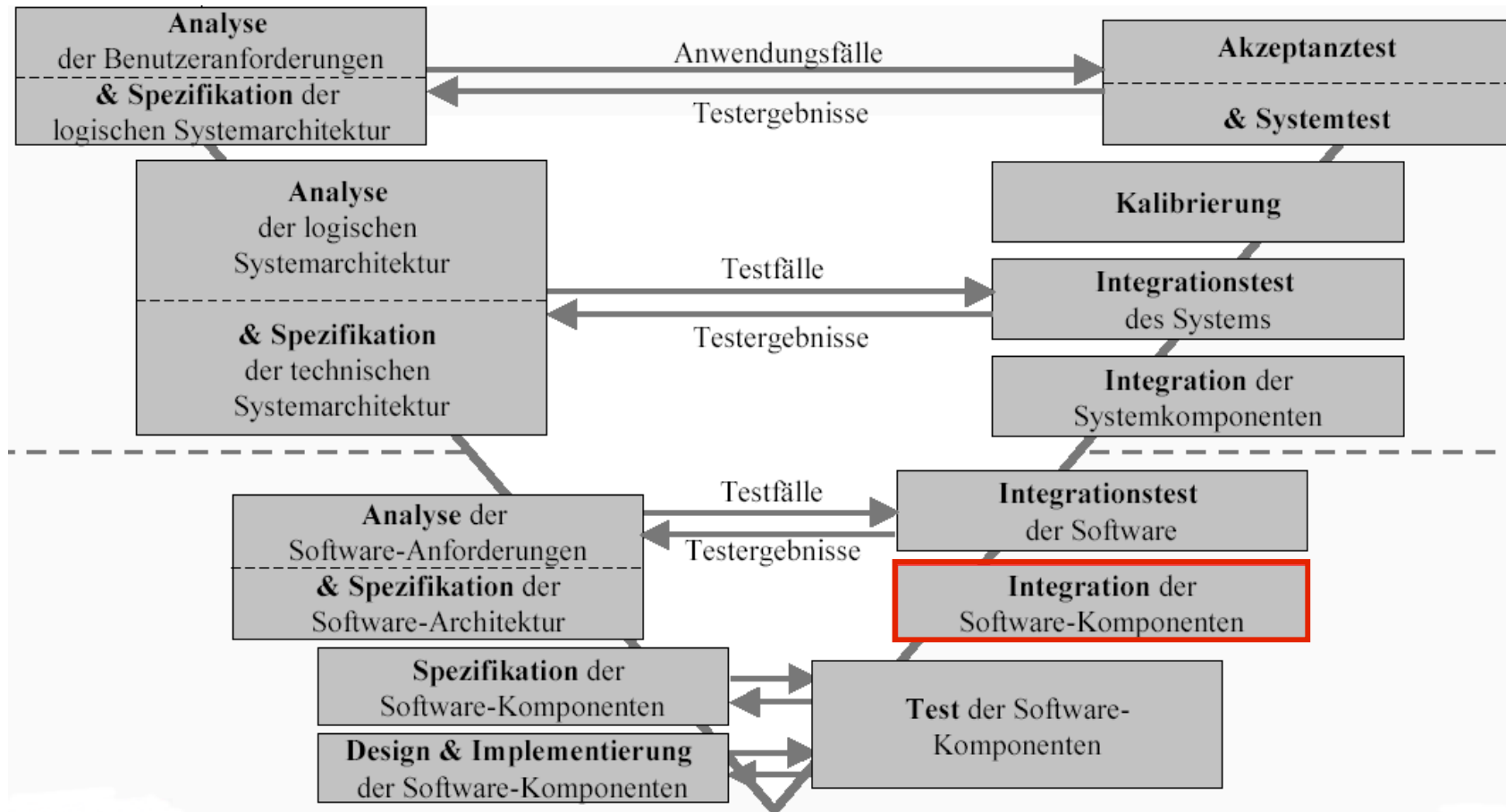
## 4. Kernprozess zur Entwicklung von elektronischen Systemen und Software



1. Grundbegriffe
2. Entwicklungsobjekt: Kombiinstrument
3. Analyse und Spezifikation der Benutzeranforderungen
4. Analyse und Spezifikation der technischen Anforderungen
5. Analyse und Spezifikation der Software-Anforderungen
6. Spezifikation der Software-Komponenten
7. Design und Implementierung der Software-Komponenten
8. Test der Software-Komponenten
- 9. Integration der Software-Komponenten**
10. Integrationstest der Software-Komponenten
11. Integration der System-Komponenten
12. Integrationstest des Systems
13. Kalibrierung
14. Akzeptanz- und Systemtest



# Integration der Software-Komponenten



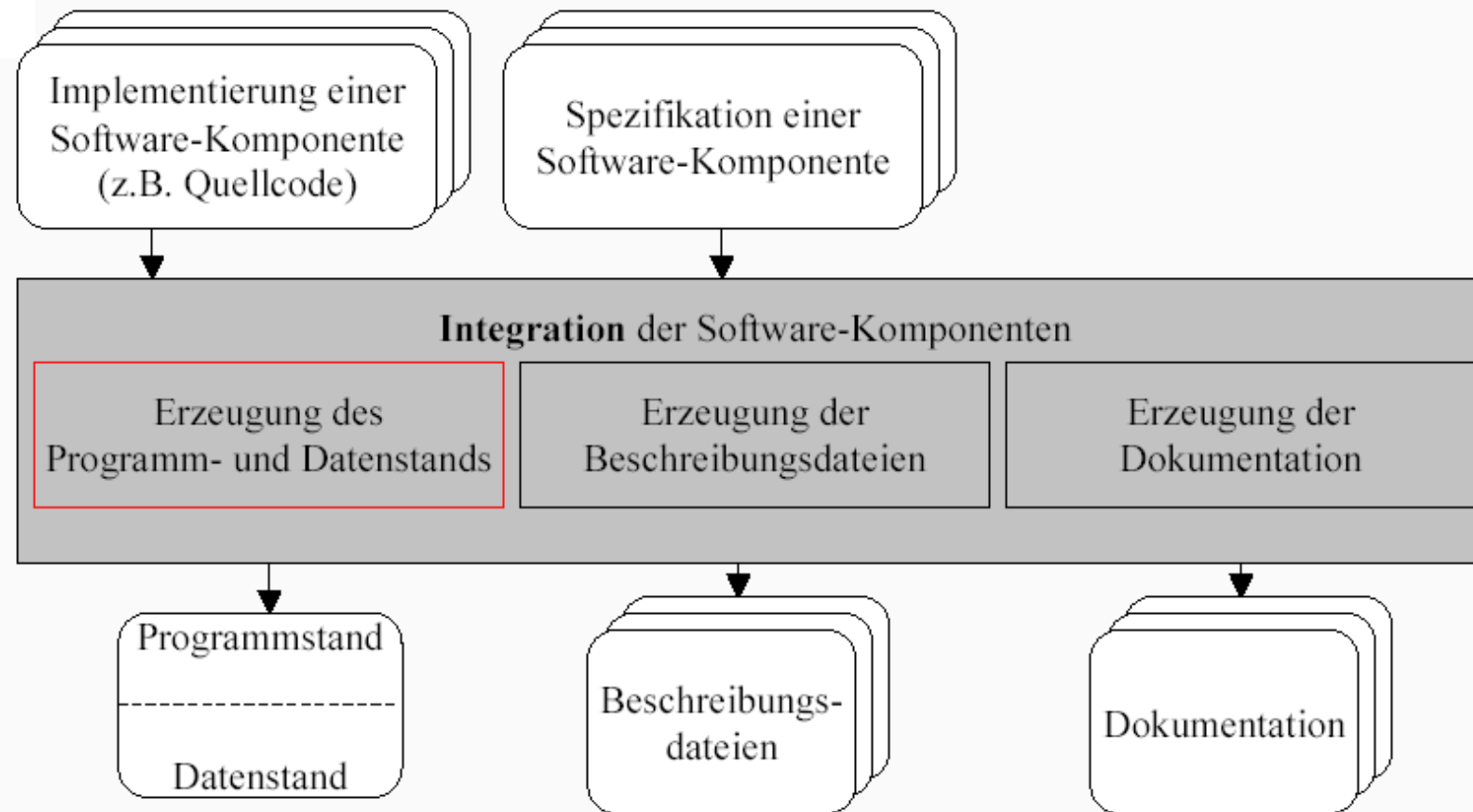
## Software-Stand für ein Seriensteuergerät

- ◆ *Programm- und Datenstände* für alle Mikrocontroller des Steuergeräts
- ◆ *Dokumentation*
- ◆ *Beschreibungsdateien* für Produktions- und Servicewerkzeuge, wie z.B. Diagnose-, Software-Parametrisierungs- oder Flash-Programmierungswerkzeuge.

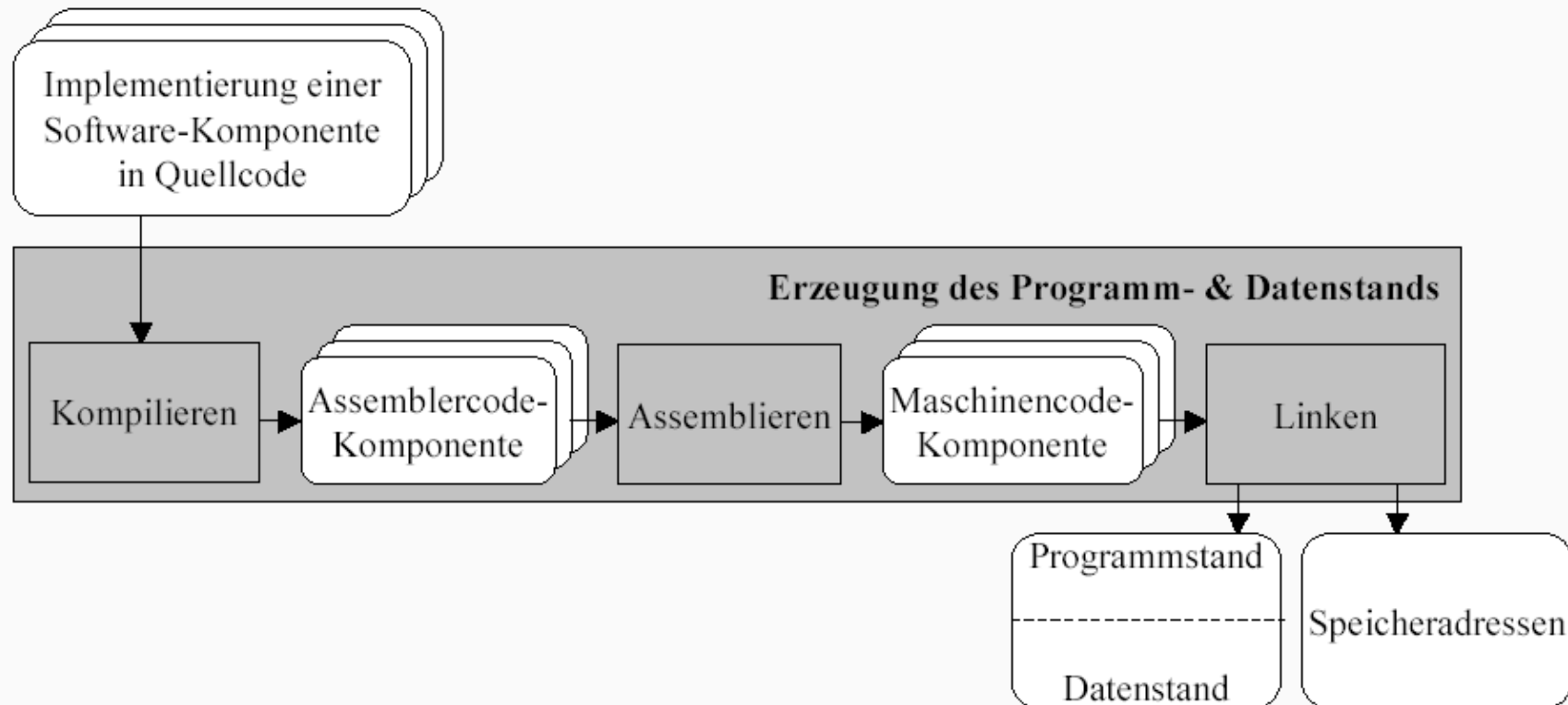
## Beschreibungsdateien für Entwicklungssteuergeräte

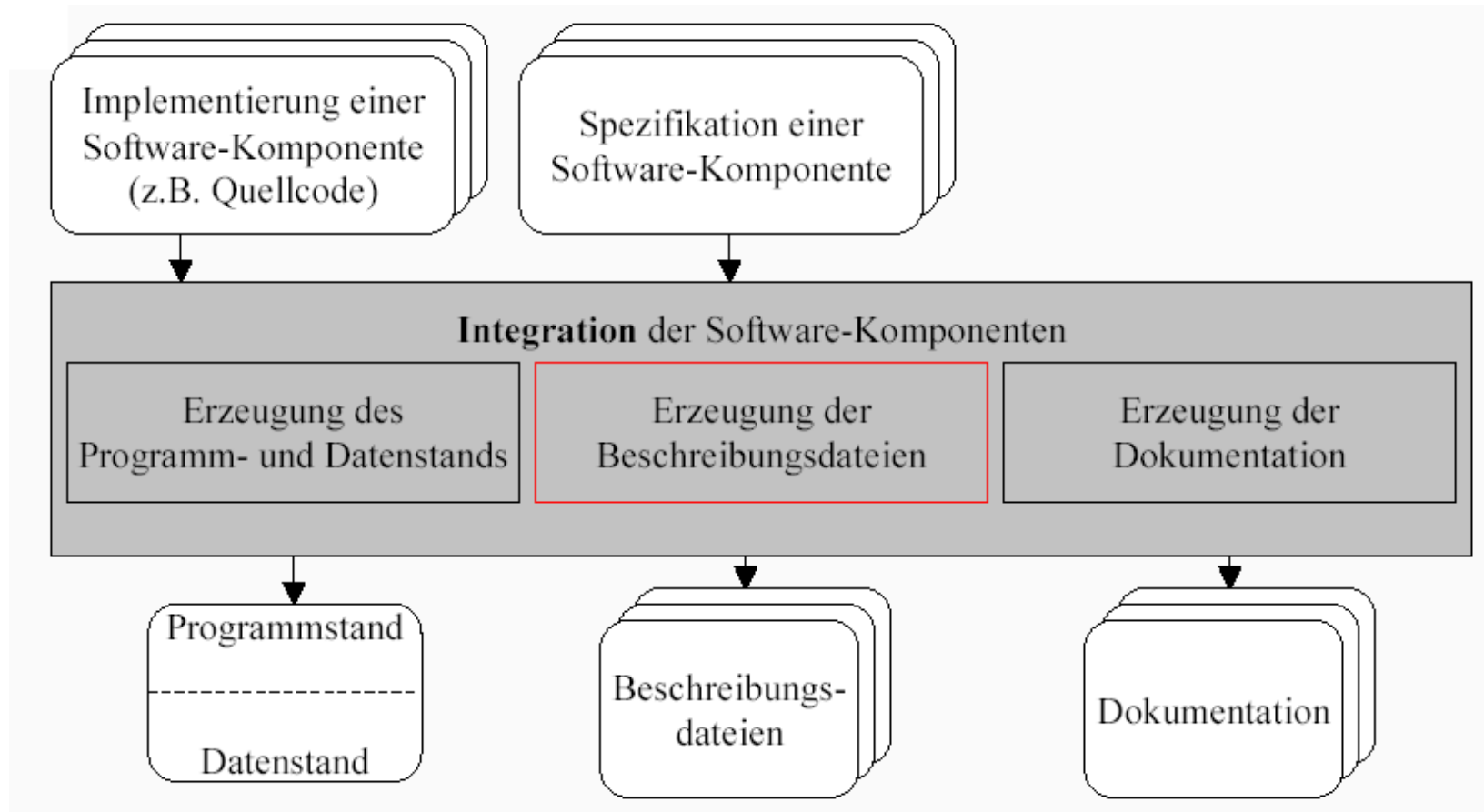
- ◆ Beschreibungsdateien für die Mess- und Kalibrierungswerkzeuge
- ◆ Beschreibungsdateien der On-Board-Kommunikation für Werkzeuge zur Netzwerkentwicklung
- ◆ Beschreibungsdateien der so genannten Bypass-Schnittstelle, falls Rapid-Prototyping-Werkzeuge eingesetzt werden.

# Integration der Software-Komponenten

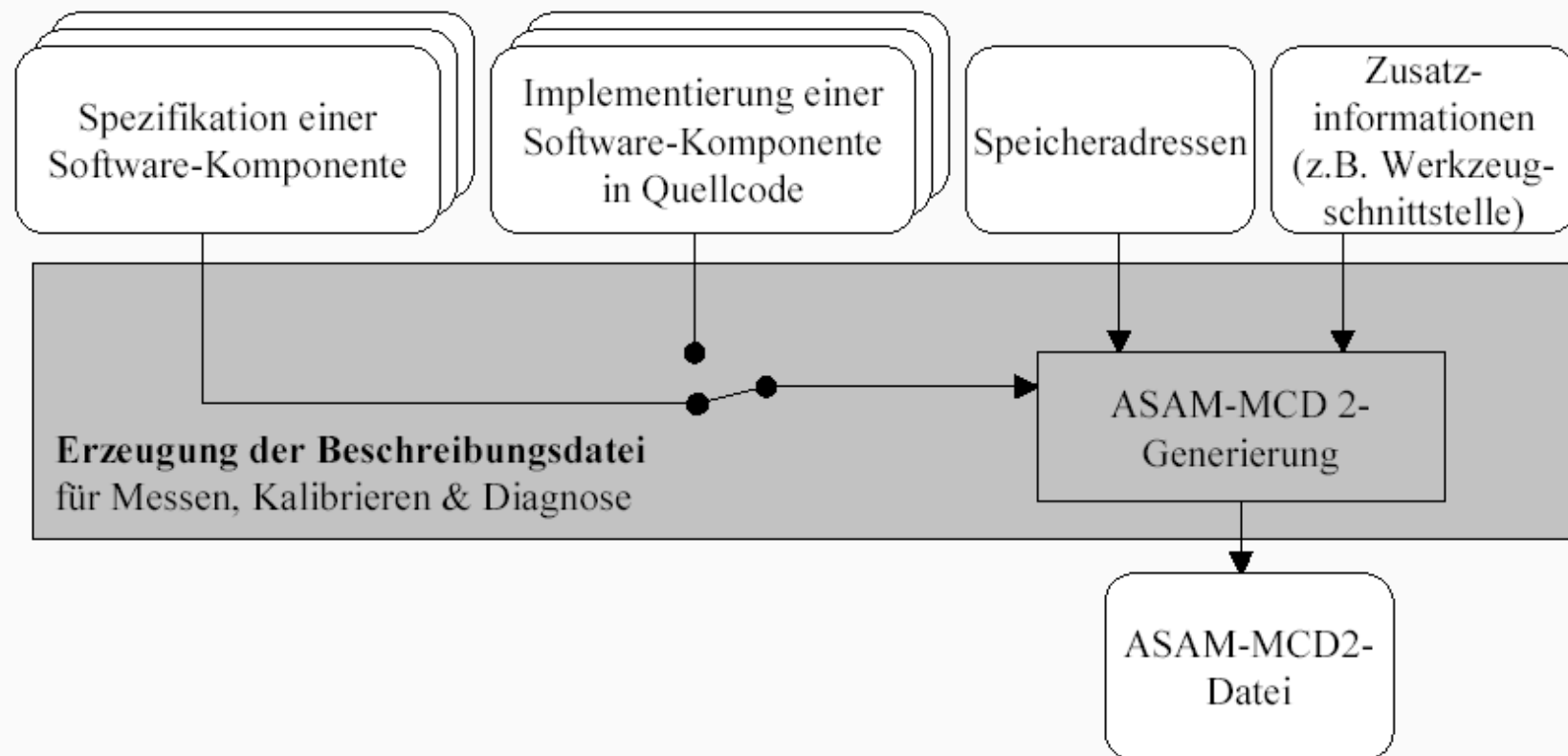


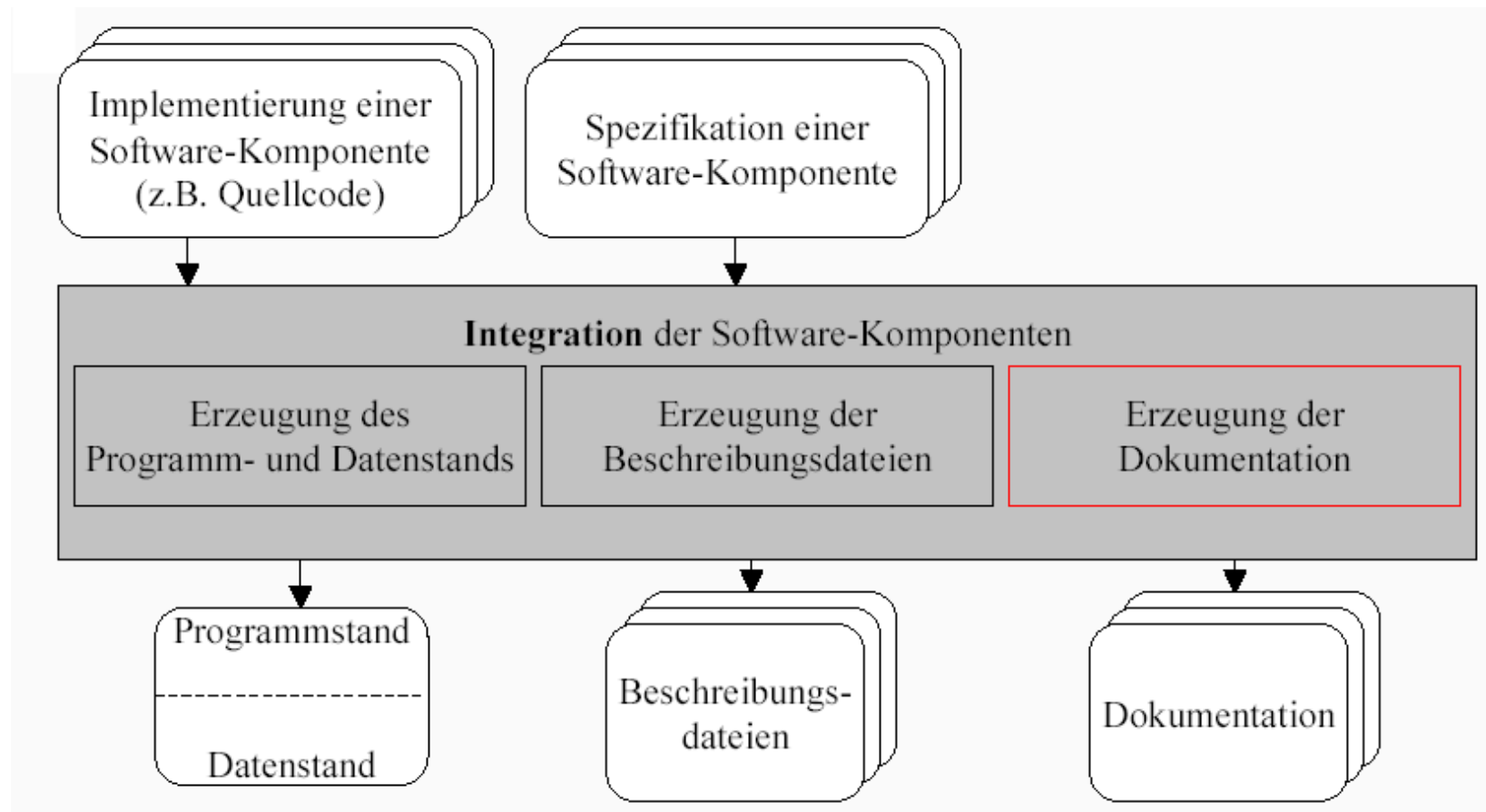
## Erzeugung des Programm- und Datenstands





## Erzeugung der Beschreibungsdateien für Mess-, Kalibrier- und Diagnosewerkzeuge



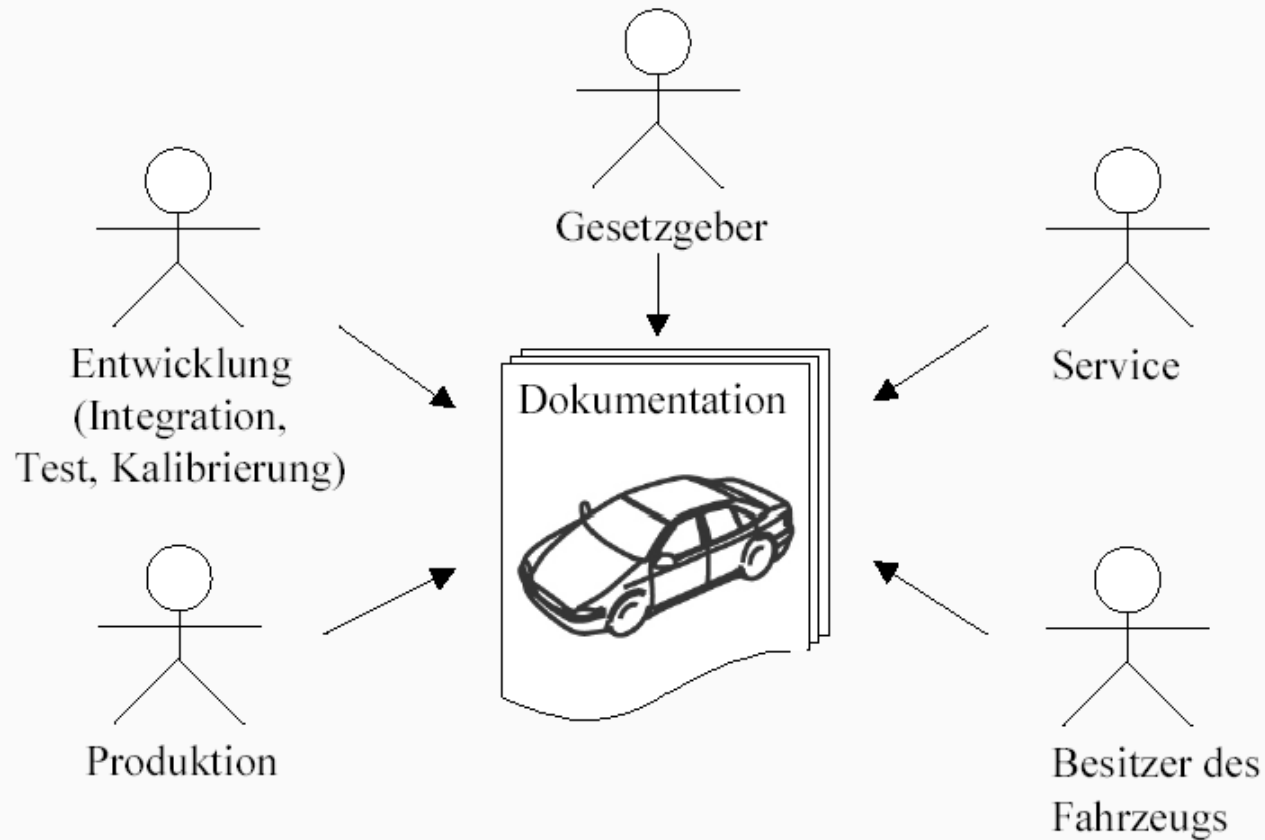


## Dokumentation

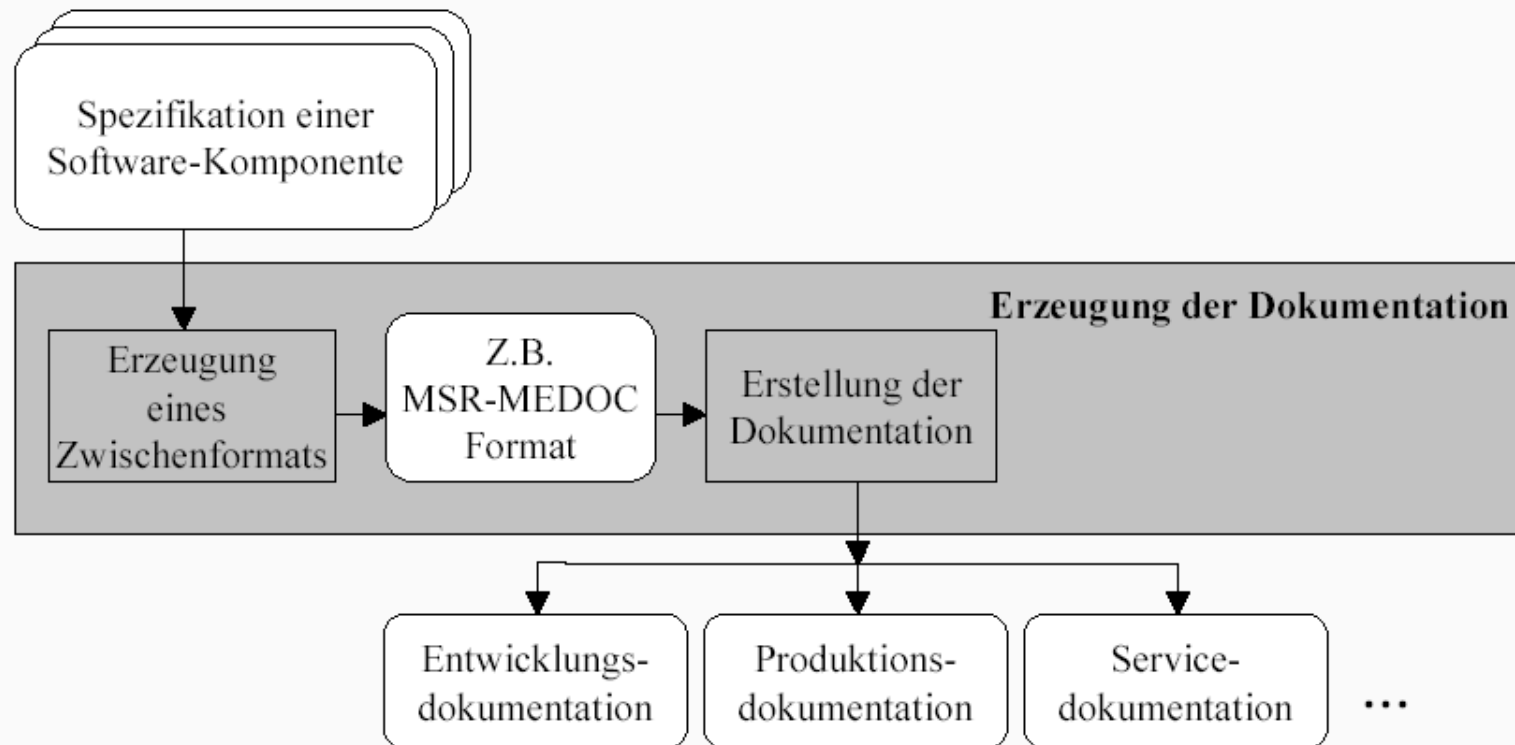
- ◆ Die Dokumentation stellt ein Artefakt dar, das für alle Unterstützungsprozesse in der Software-Entwicklung selbst benötigt wird. Auch die ausgeprägte und oft firmenübergreifende Arbeitsteilung, die langen Produktlebenszyklen, sowie die damit verbundene lange Wartungsphase für die Software erfordern eine ausführliche Dokumentation.
- ◆ Alle folgenden Entwicklungsschritte - wie Integration, Test und Kalibrierung des Systems - benötigen eine Dokumentation.
- ◆ Eine Dokumentation ist für die Fahrzeugproduktion und im weltweiten Service notwendig.
- ◆ Für den Gesetzgeber ist eine Dokumentation notwendig, etwa als Bestandteil für die Beantragung der Zulassung eines Fahrzeugs zum Straßenverkehr.



## Benutzergruppen einer Dokumentation für Software-Funktionen



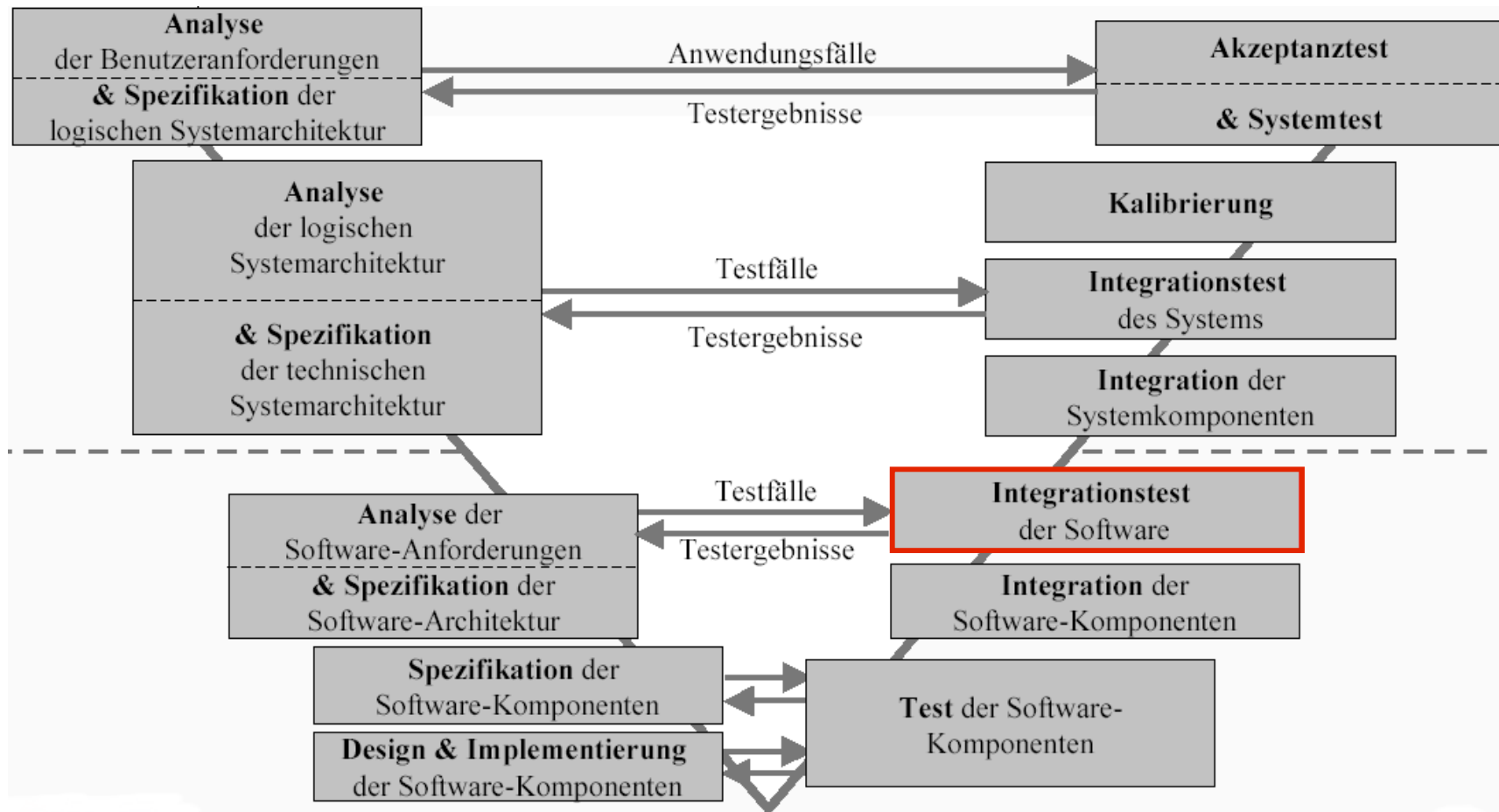
## Erzeugung der Dokumentation



## 4. Kernprozess zur Entwicklung von elektronischen Systemen und Software



1. Grundbegriffe
2. Entwicklungsobjekt: Kombiinstrument
3. Analyse und Spezifikation der Benutzeranforderungen
4. Analyse und Spezifikation der technischen Anforderungen
5. Analyse und Spezifikation der Software-Anforderungen
6. Spezifikation der Software-Komponenten
7. Design und Implementierung der Software-Komponenten
8. Test der Software-Komponenten
9. Integration der Software-Komponenten
- 10. Integrationstest der Software-Komponenten**
11. Integration der System-Komponenten
12. Integrationstest des Systems
13. Kalibrierung
14. Akzeptanz- und Systemtest





## Statistische Prüfung bei der Zusammenführung von Software-Komponenten

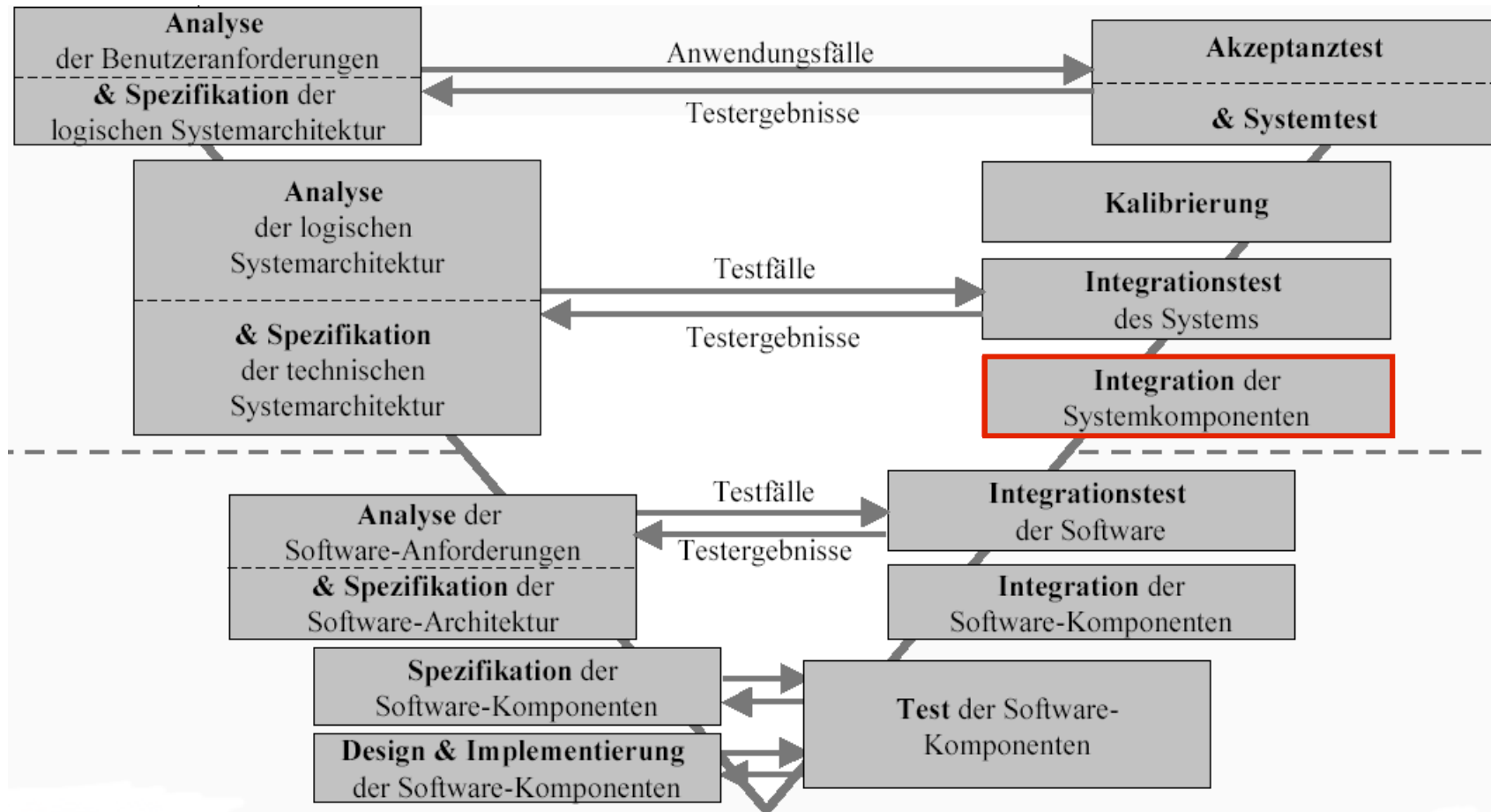
- ◆ teilweise manuell, teilweise automatisiert
- ◆ Schnittstellenspezifikationen eingehalten?
- ◆ Namensraum für Variablen eingehalten?
- ◆ Speicherlayout

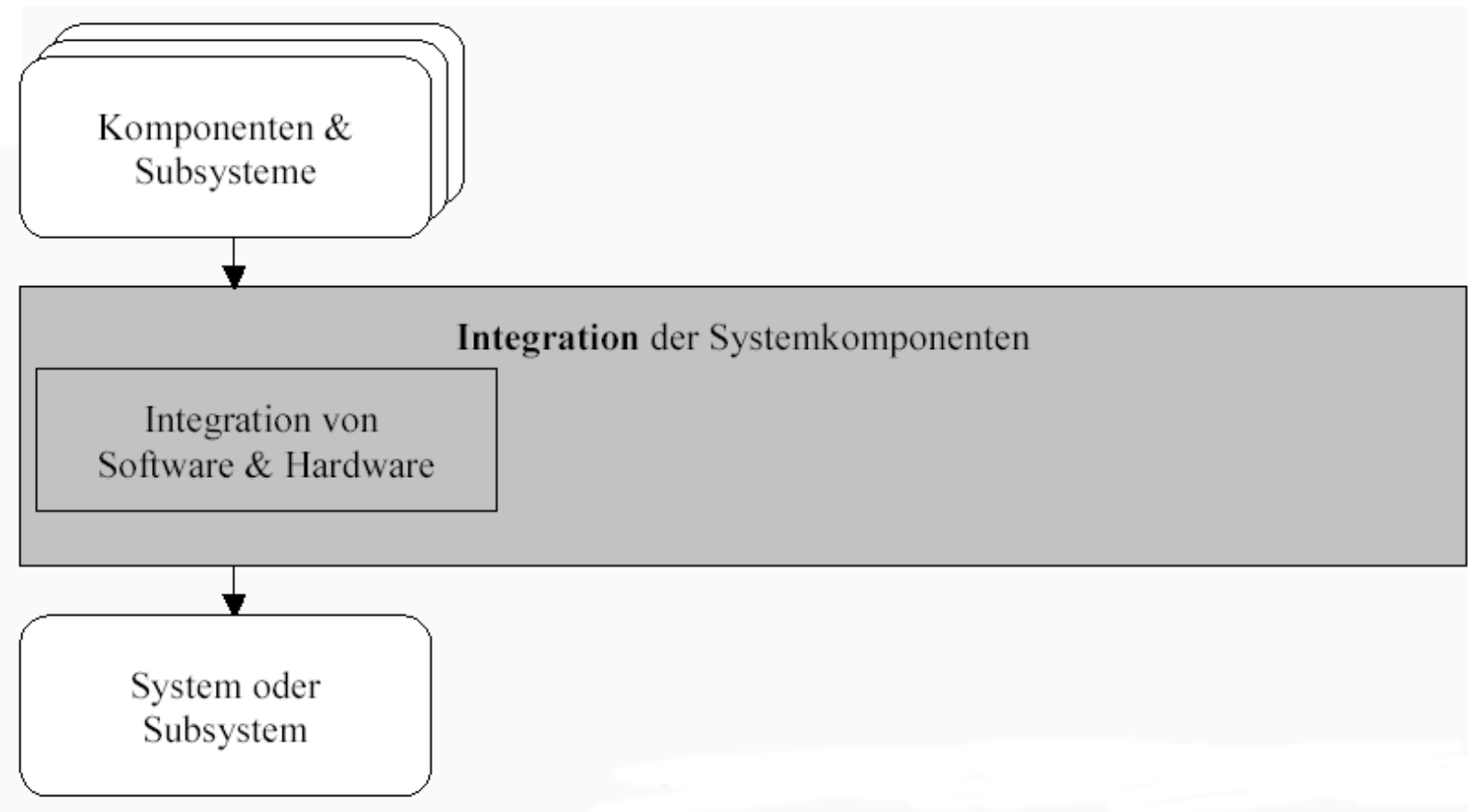
## 4. Kernprozess zur Entwicklung von elektronischen Systemen und Software



1. Grundbegriffe
2. Entwicklungsobjekt: Kombiinstrument
3. Analyse und Spezifikation der Benutzeranforderungen
4. Analyse und Spezifikation der technischen Anforderungen
5. Analyse und Spezifikation der Software-Anforderungen
6. Spezifikation der Software-Komponenten
7. Design und Implementierung der Software-Komponenten
8. Test der Software-Komponenten
9. Integration der Software-Komponenten
10. Integrationstest der Software-Komponenten
- 11. Integration der System-Komponenten**
12. Integrationstest des Systems
13. Kalibrierung
14. Akzeptanz- und Systemtest

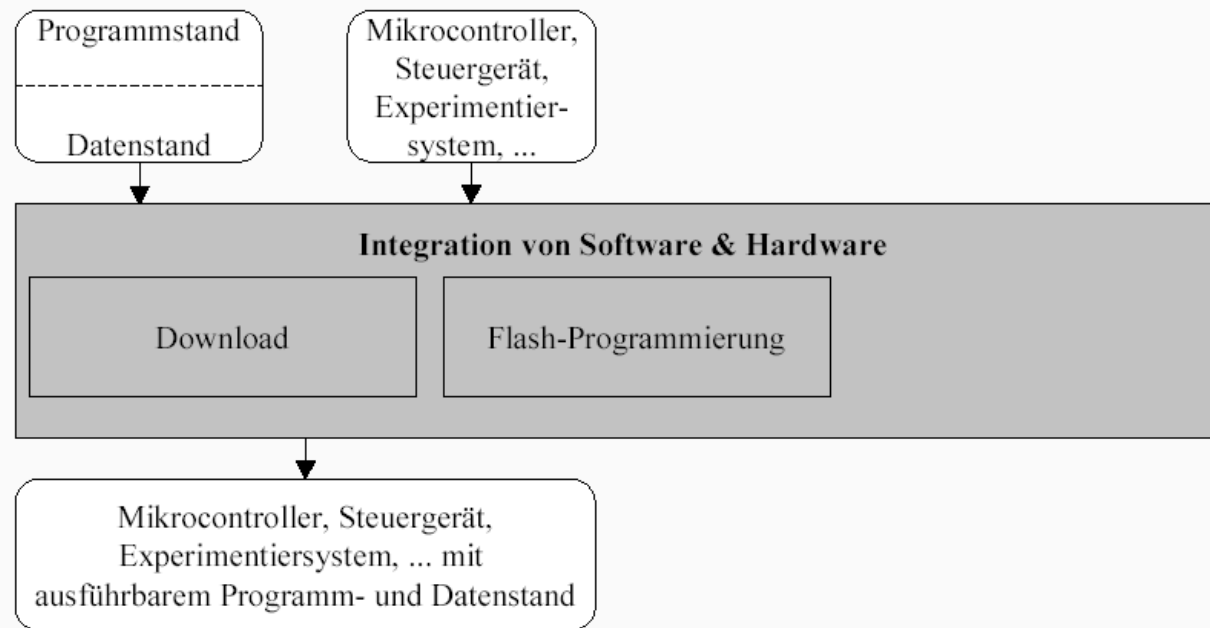
# Integration der System-Komponenten







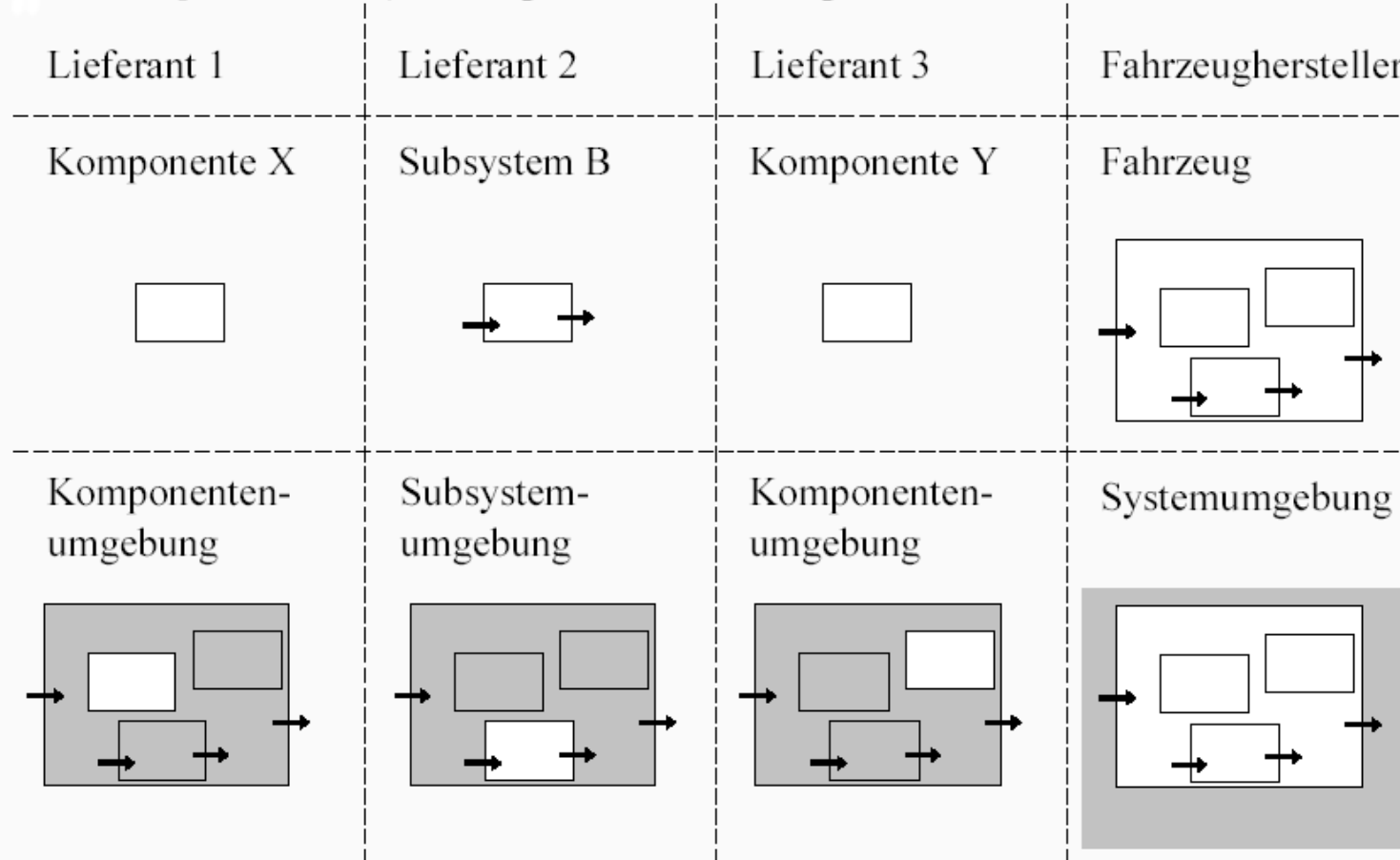
## Integration von Software und Hardware



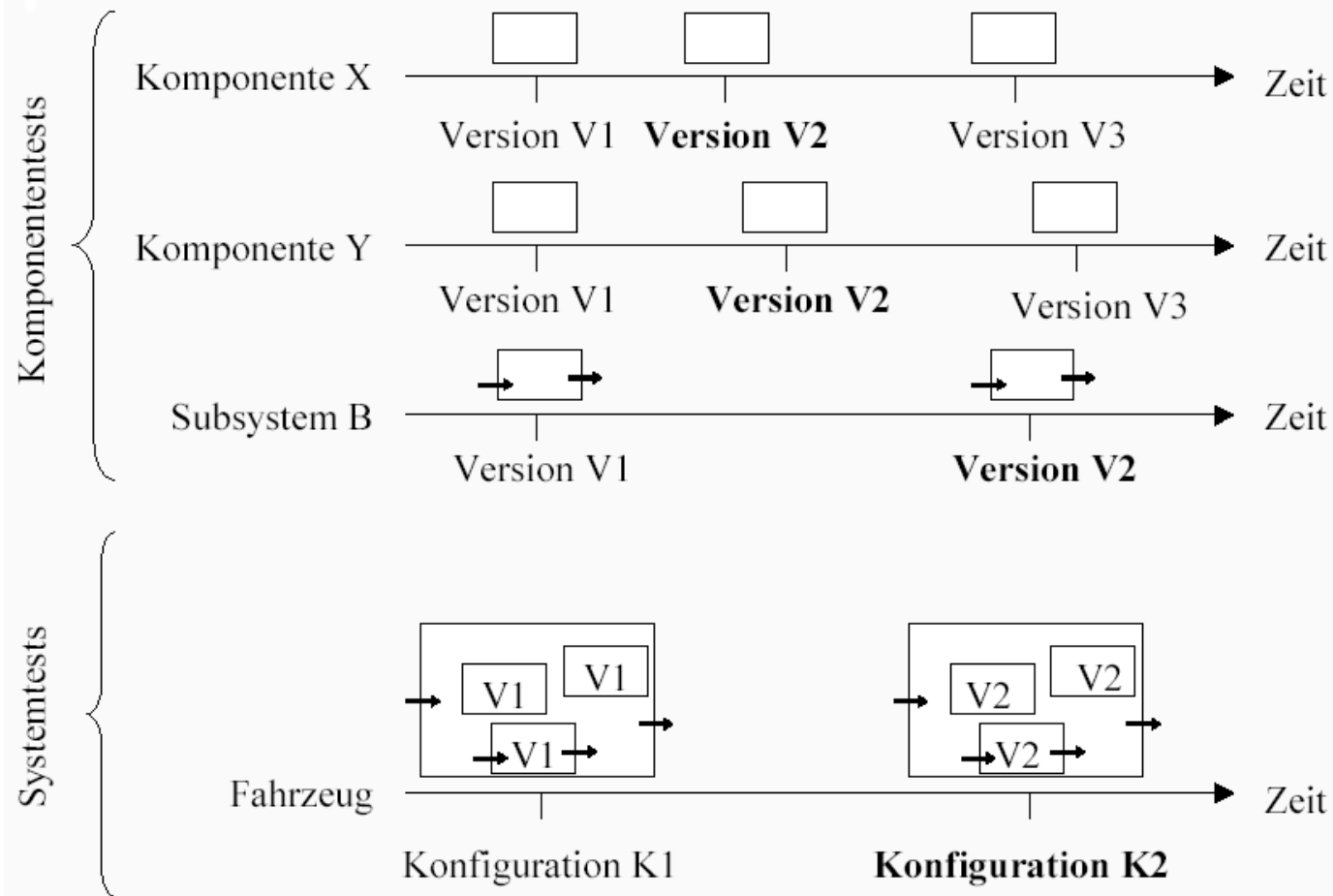
## Integration von Steuergeräten, Sollwertgebern, Sensoren und Aktuatoren

- ◆ Abnahmetests für vom Zulieferer gelieferte Komponenten oder Subsysteme
- ◆ nur eingeschränkte Testmöglichkeiten beim Zulieferer
- ◆ Komponentenintegration als Synchronisationspunkt

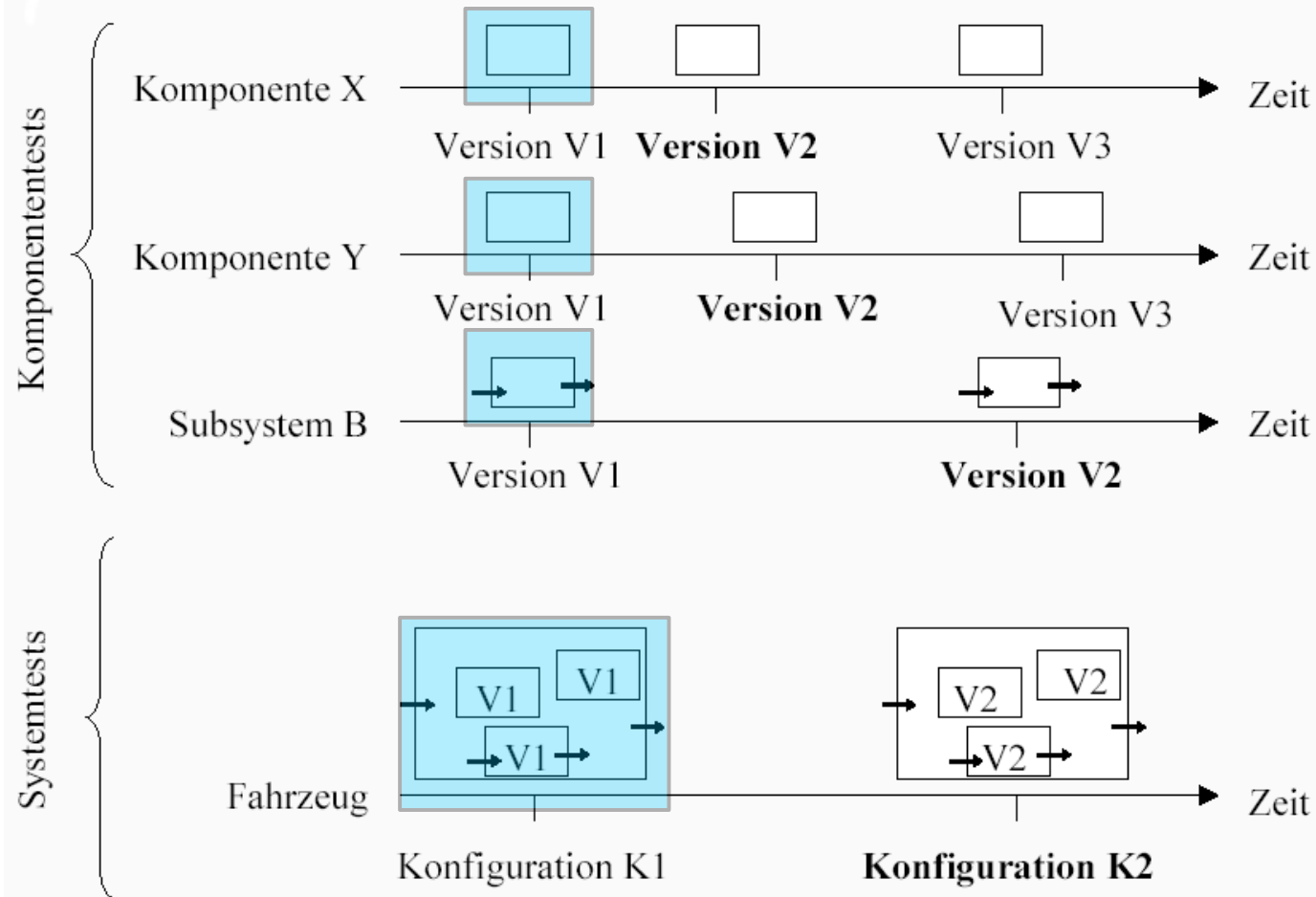
## Unterschiedliche Umgebungen für Komponenten, Subsysteme und Systeme

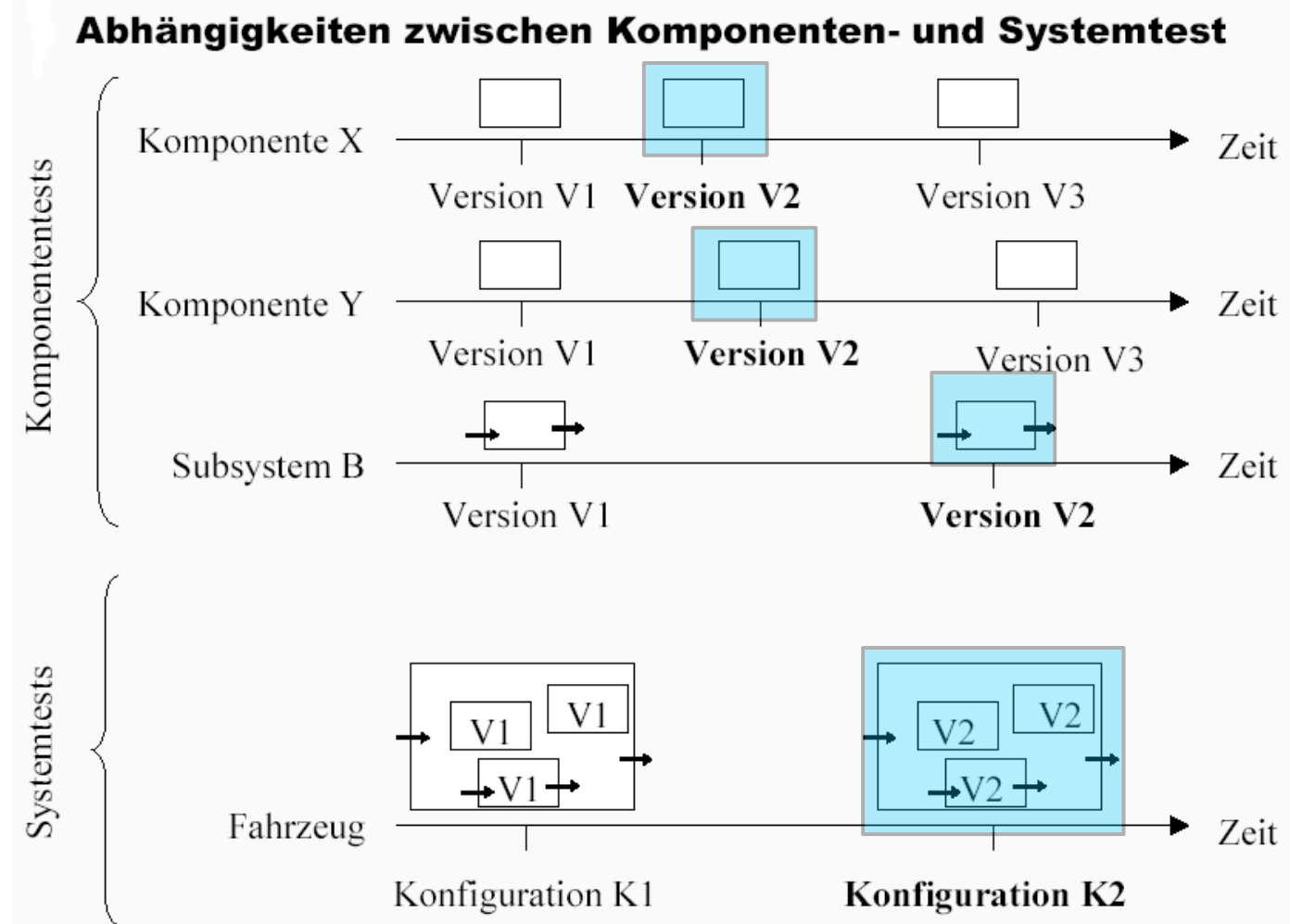


## Abhängigkeiten zwischen Komponenten- und Systemtest



## Abhängigkeiten zwischen Komponenten- und Systemtest



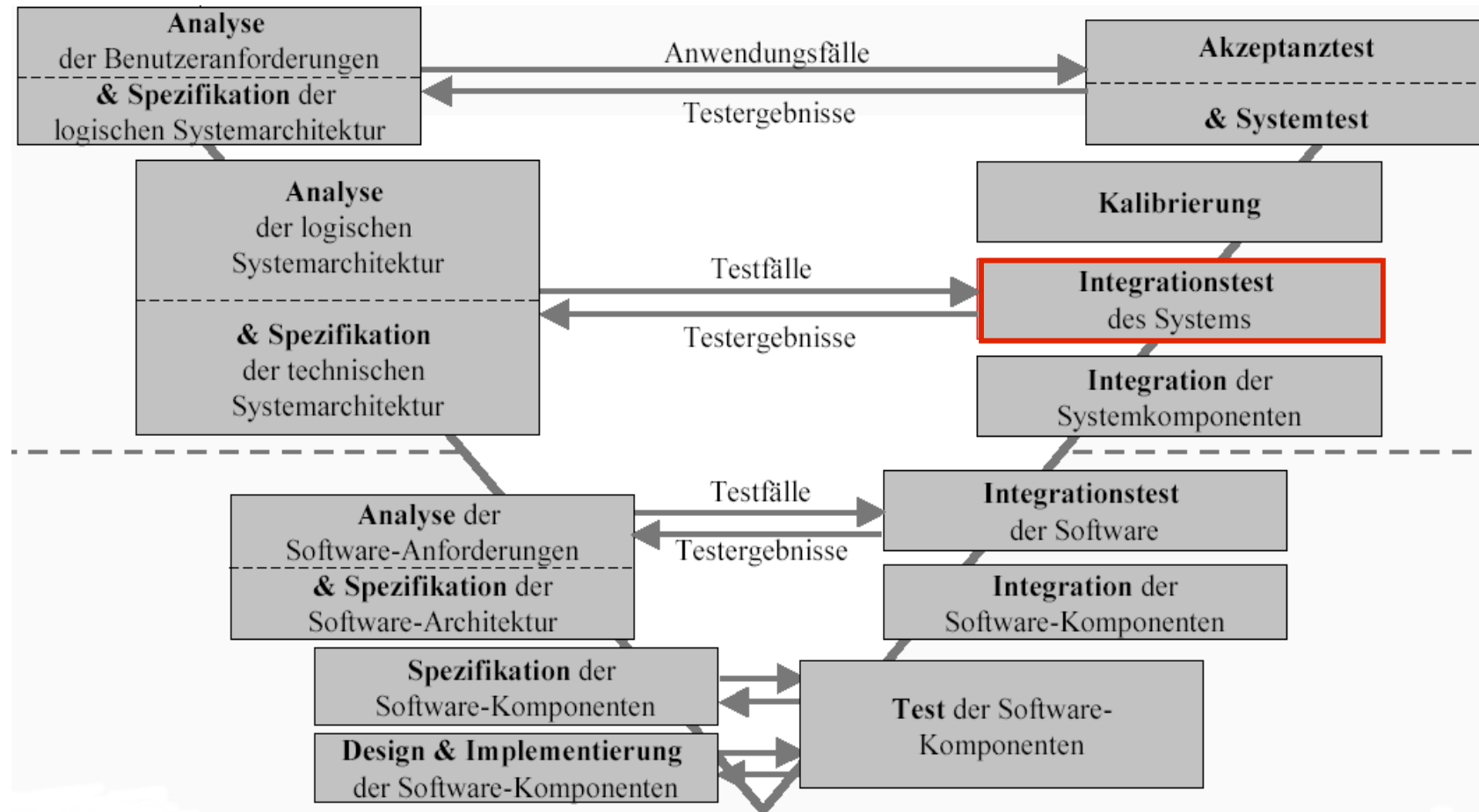


## 4. Kernprozess zur Entwicklung von elektronischen Systemen und Software



1. Grundbegriffe
2. Entwicklungsobjekt: Kombiinstrument
3. Analyse und Spezifikation der Benutzeranforderungen
4. Analyse und Spezifikation der technischen Anforderungen
5. Analyse und Spezifikation der Software-Anforderungen
6. Spezifikation der Software-Komponenten
7. Design und Implementierung der Software-Komponenten
8. Test der Software-Komponenten
9. Integration der Software-Komponenten
10. Integrationstest der Software-Komponenten
11. Integration der System-Komponenten
- 12. Integrationstest des Systems**
13. Kalibrierung
14. Akzeptanz- und Systemtest

# Integrationstest des Systems

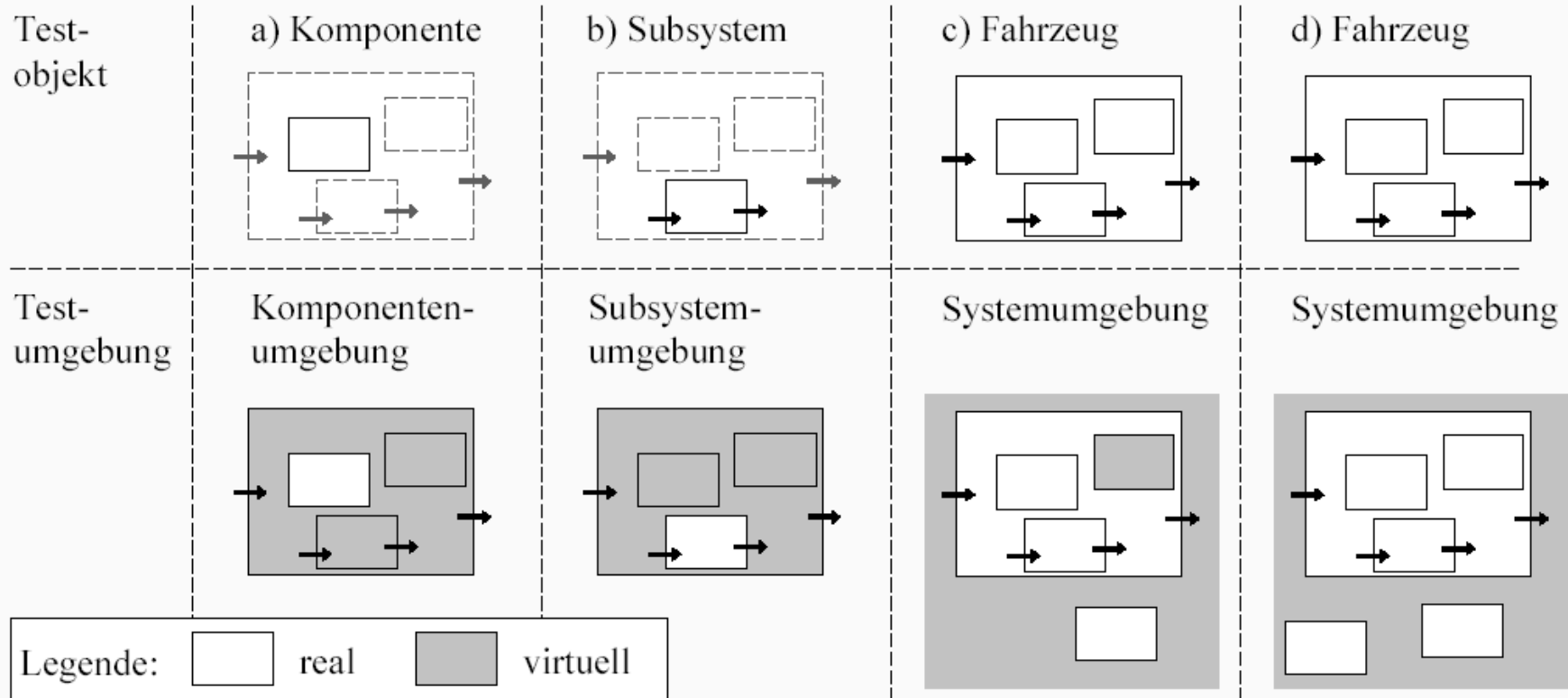


# Integrationstest des Systems





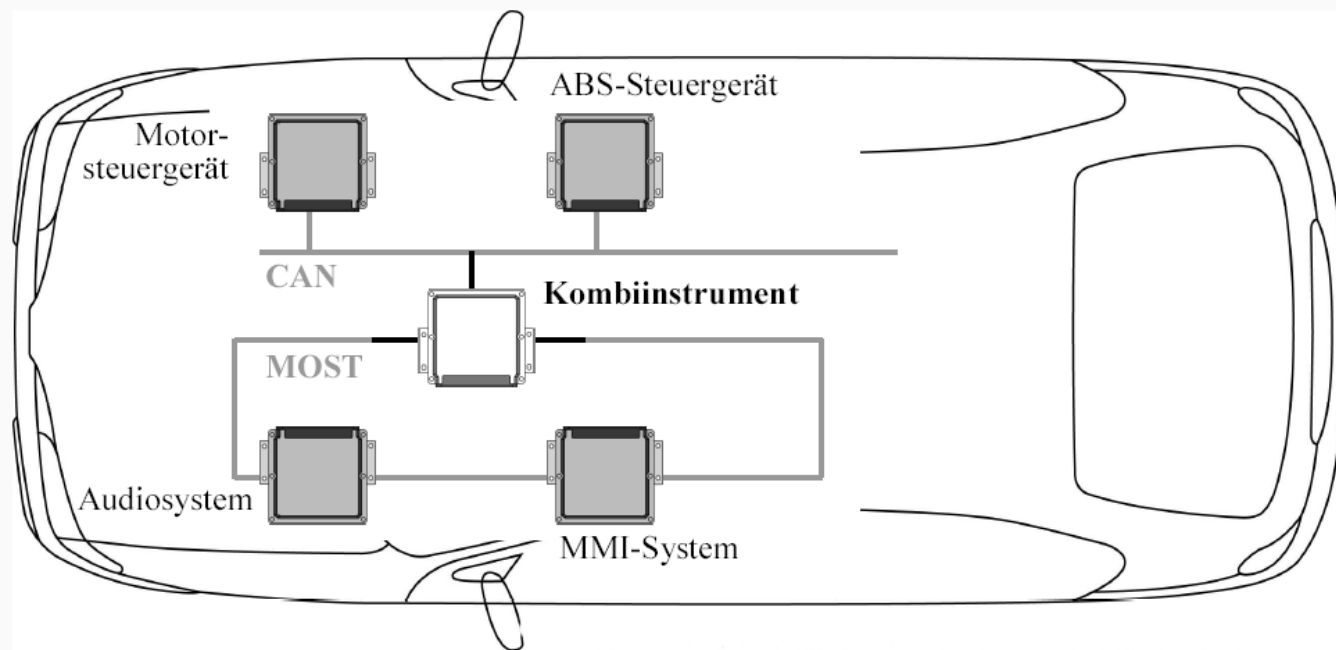
## Testobjekt und Testumgebung



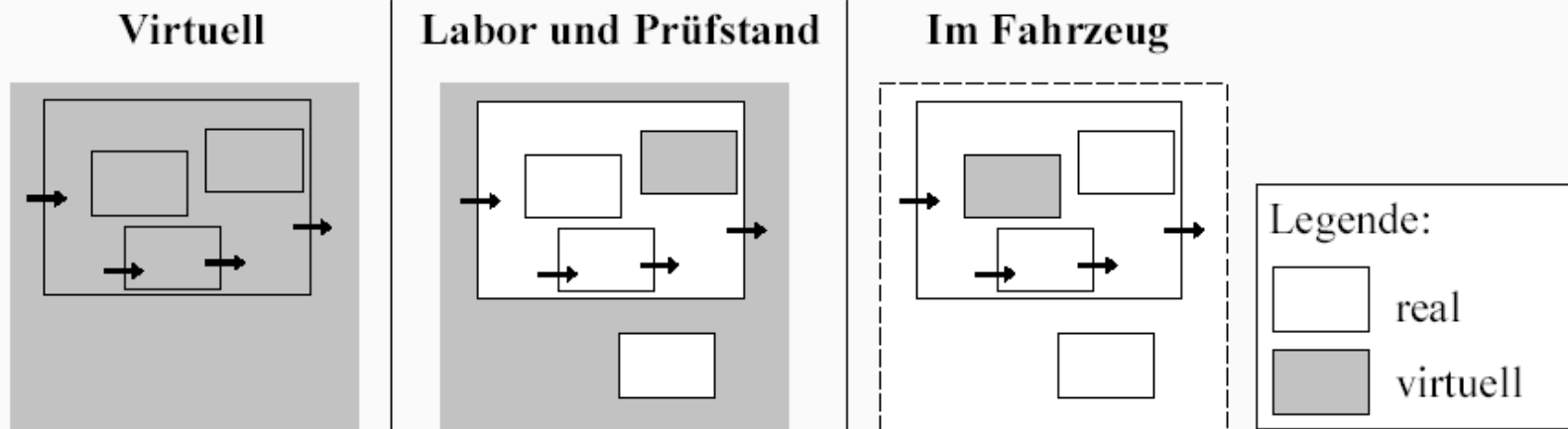
## Vorteile virtueller Testumgebungen

- ◆ Prüfschritte im Labor oder am Prüfstand statt im Fahrzeug
- ◆ Reproduzierbarkeit, Automatisierung
- ◆ Extremsituationen ohne Gefährdung von Testfahrern oder Prototypen

## Virtuelle Netzwerkumgebung für das Kombiinstrument



## Durchgängiger Integrations- und Testprozess

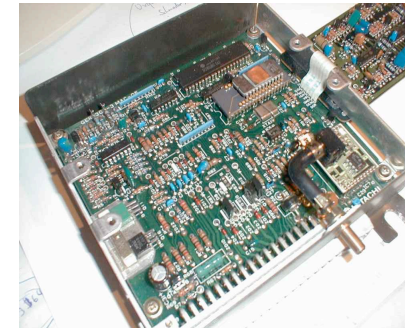


# Entwicklungsprozesse “Software/Hardware in the Loop”



## ■ Begriffsdefinitionen:

- HIL Hardware in the Loop
  - Reale SG-Hardware wird in simulierter Fahrzeugumgebung gespeist, d.h. mit rechnergenerierten Sensor- und Bussignalen angesteuert.
- SIL Software in the Loop
  - SG-Software „läuft“ auf simuliertem SG, das von simulierter Fahrzeugumgebung gespeist wird.



Steuergerät

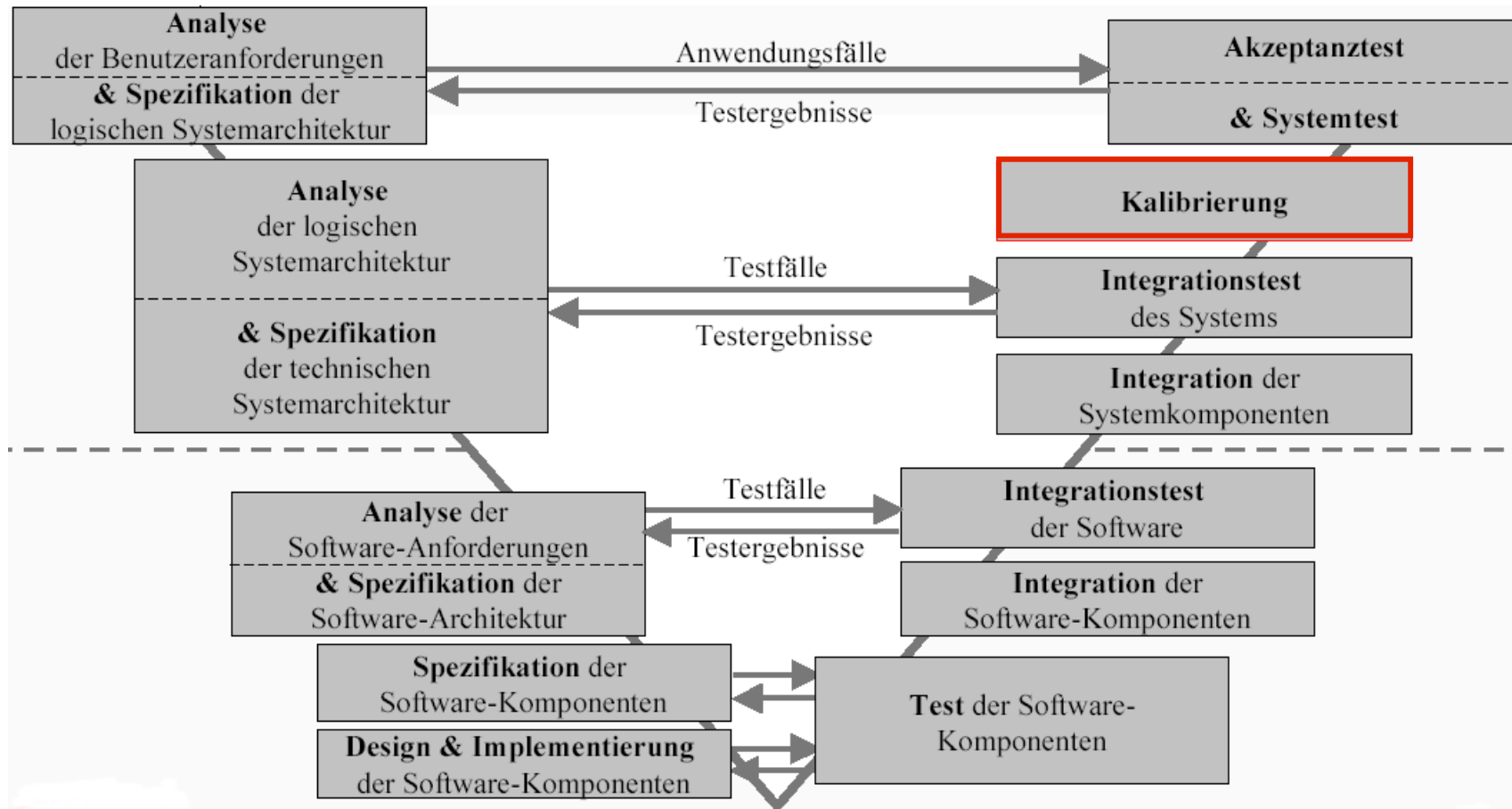


|           |                 |                    |
|-----------|-----------------|--------------------|
|           | simuliert       | real               |
| simuliert | <b>SIL</b>      | <b>HIL</b>         |
| real      | <b>Prototyp</b> | <b>Fahrversuch</b> |

## 4. Kernprozess zur Entwicklung von elektronischen Systemen und Software

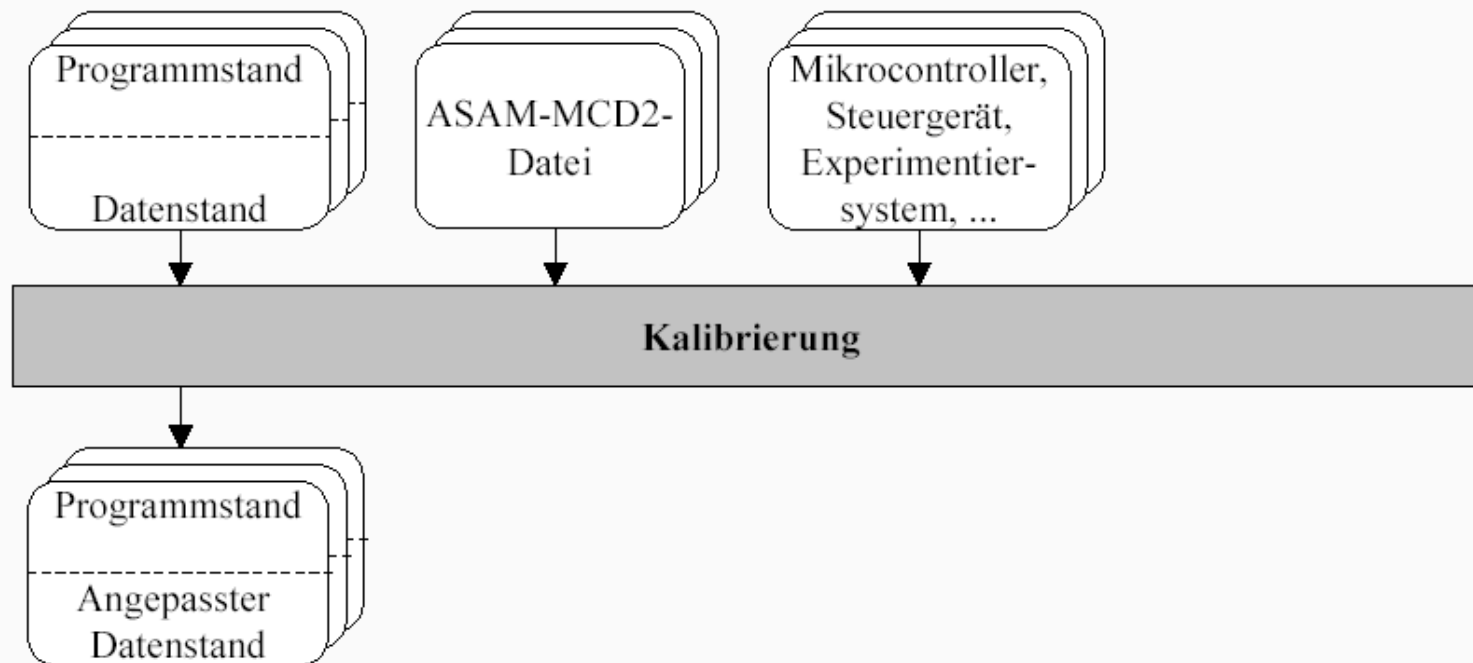


1. Grundbegriffe
2. Entwicklungsobjekt: Kombiinstrument
3. Analyse und Spezifikation der Benutzeranforderungen
4. Analyse und Spezifikation der technischen Anforderungen
5. Analyse und Spezifikation der Software-Anforderungen
6. Spezifikation der Software-Komponenten
7. Design und Implementierung der Software-Komponenten
8. Test der Software-Komponenten
9. Integration der Software-Komponenten
10. Integrationstest der Software-Komponenten
11. Integration der System-Komponenten
12. Integrationstest des Systems
- 13. Kalibrierung**
14. Akzeptanz- und Systemtest



## Kalibrierung

- ◆ fahrzeugindividuelle Einstellung der Parameter der Software-Funktionen
- ◆ häufig nur direkt im Fahrzeug bei laufenden Systemen
- ◆ *Kalibriersystem*: Steuergerät mit Off-Board-Schnittstelle zu einem Meß- und Kalibrierwerkzeug
- ◆ Datenstände in einem Festwertspeicher (ROM, EEPROM oder Flash)

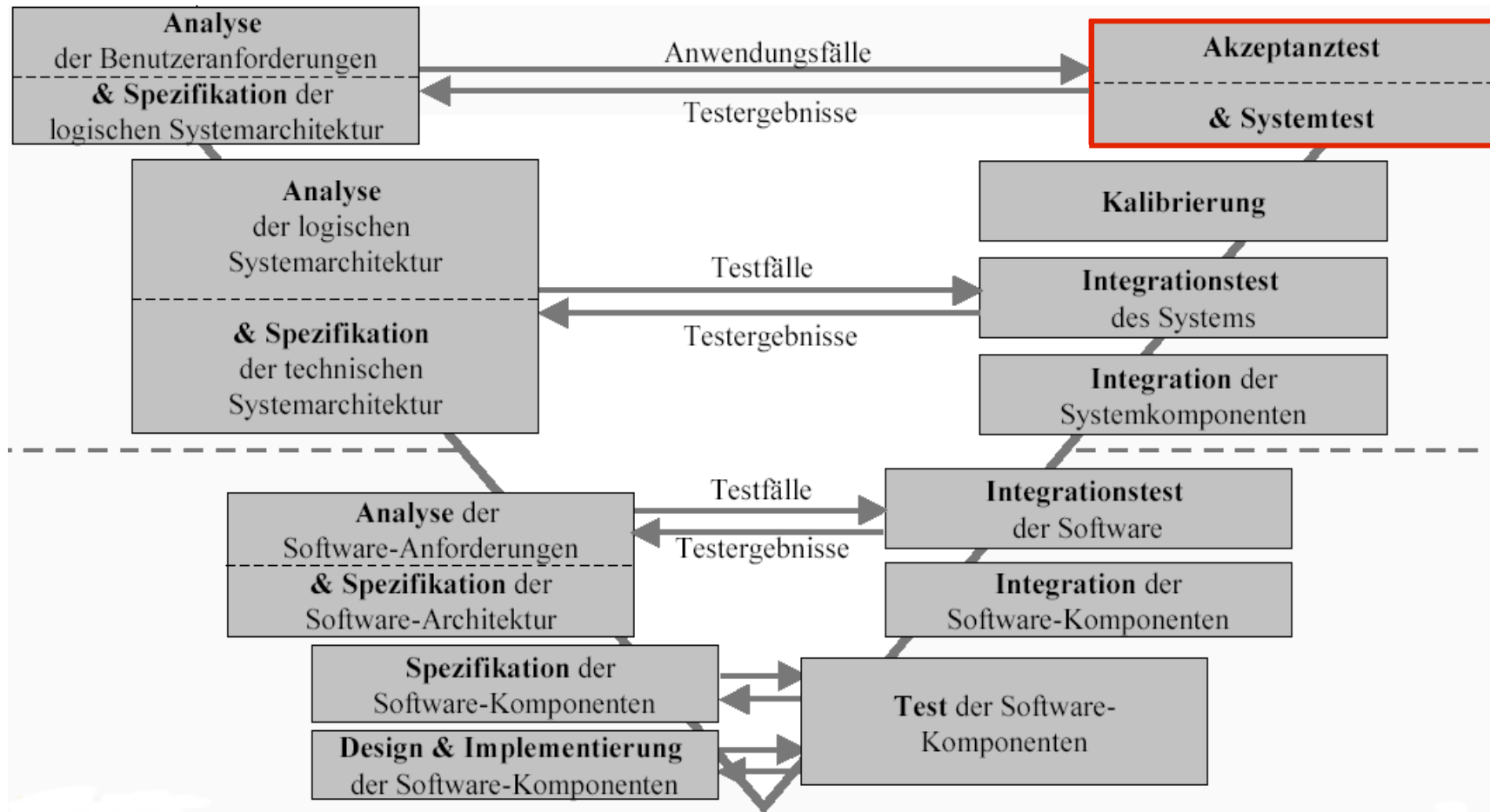


## 4. Kernprozess zur Entwicklung von elektronischen Systemen und Software



1. Grundbegriffe
2. Entwicklungsobjekt: Kombiinstrument
3. Analyse und Spezifikation der Benutzeranforderungen
4. Analyse und Spezifikation der technischen Anforderungen
5. Analyse und Spezifikation der Software-Anforderungen
6. Spezifikation der Software-Komponenten
7. Design und Implementierung der Software-Komponenten
8. Test der Software-Komponenten
9. Integration der Software-Komponenten
10. Integrationstest der Software-Komponenten
11. Integration der System-Komponenten
12. Integrationstest des Systems
13. Kalibrierung
- 14. Akzeptanz- und Systemtest**





## Prüfschritte im Fahrzeug

- ◆ Restrisiko durch Simulationsmodelle
- ◆ Systemtest im Fahrversuch:  
Akzeptanztest in der realen Betriebsumgebung
- ◆ fahrzeugtauglicher Zugang zu Steuergeräten und Netzwerken
- ◆ mobile Messtechnik

