

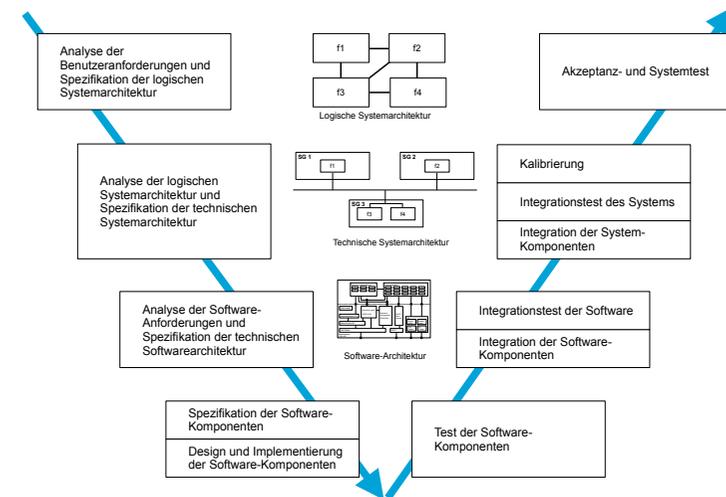
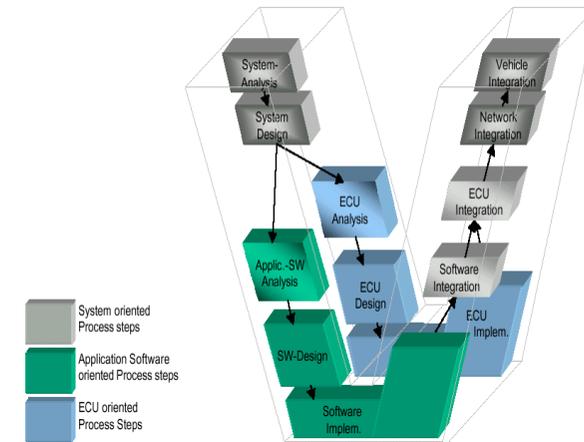
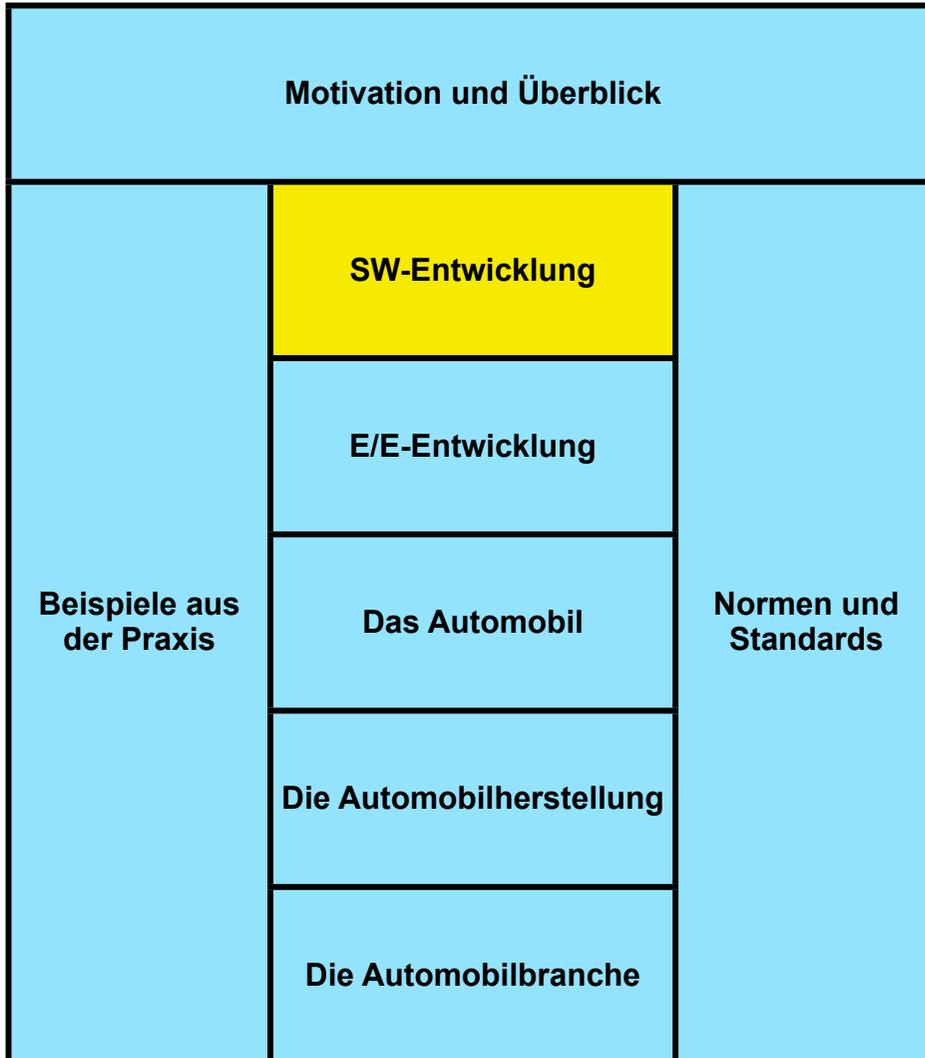
Vorlesung Automotive Software Engineering Teil 6 SW-Entwicklung (3)

TU Dresden, Fakultät Informatik

Sommersemester 2012

Prof. Dr. rer. nat. Bernhard Hohlfeld

bernhard.hohlfeld@daad-alumni.de



6. SW-Entwicklung / 2. Unterstützungsprozesse

Unterstützungsprozesse für die Embedded Software Entwicklung

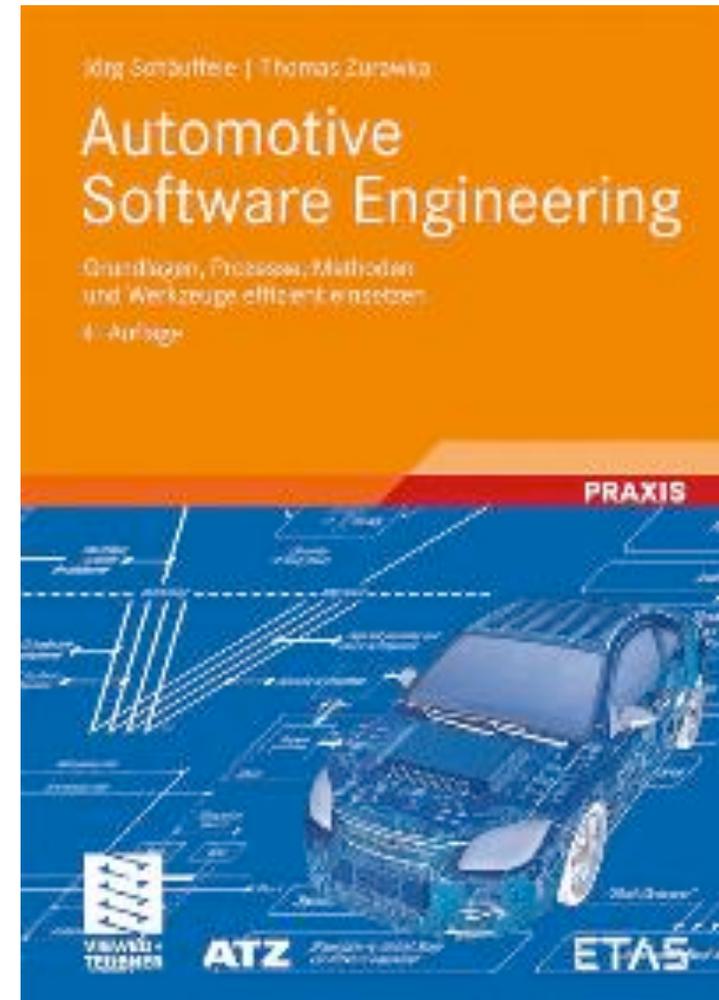
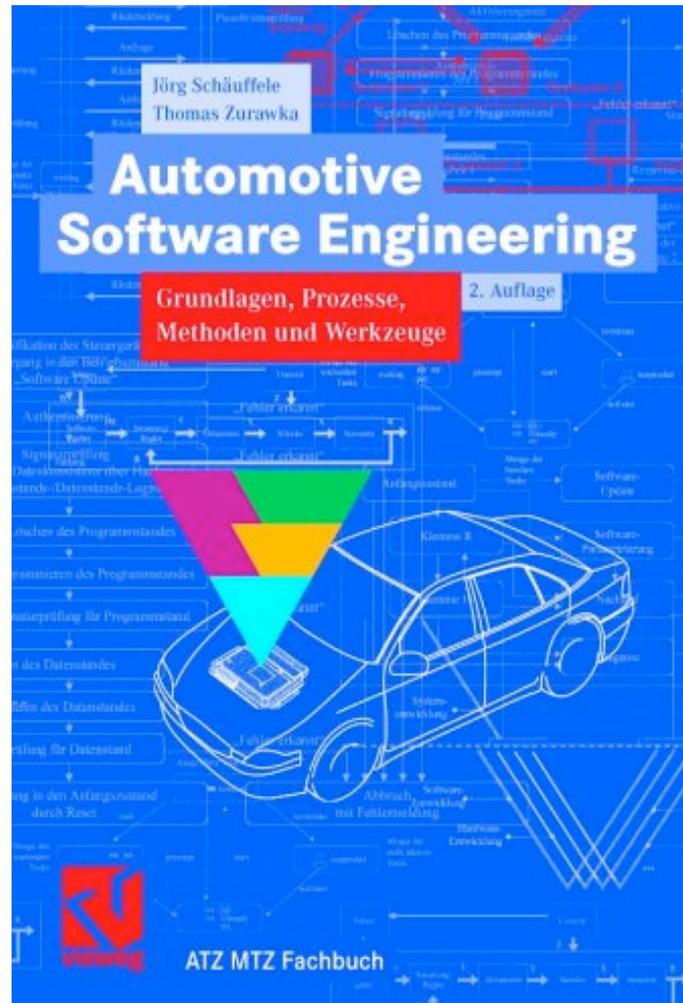


1. Vorgehensmodelle und Standards
2. Konfigurationsmanagement
3. Projektmanagement
4. Lieferantenmanagement
5. Anforderungsmanagement
6. Qualitätssicherung

Quelle



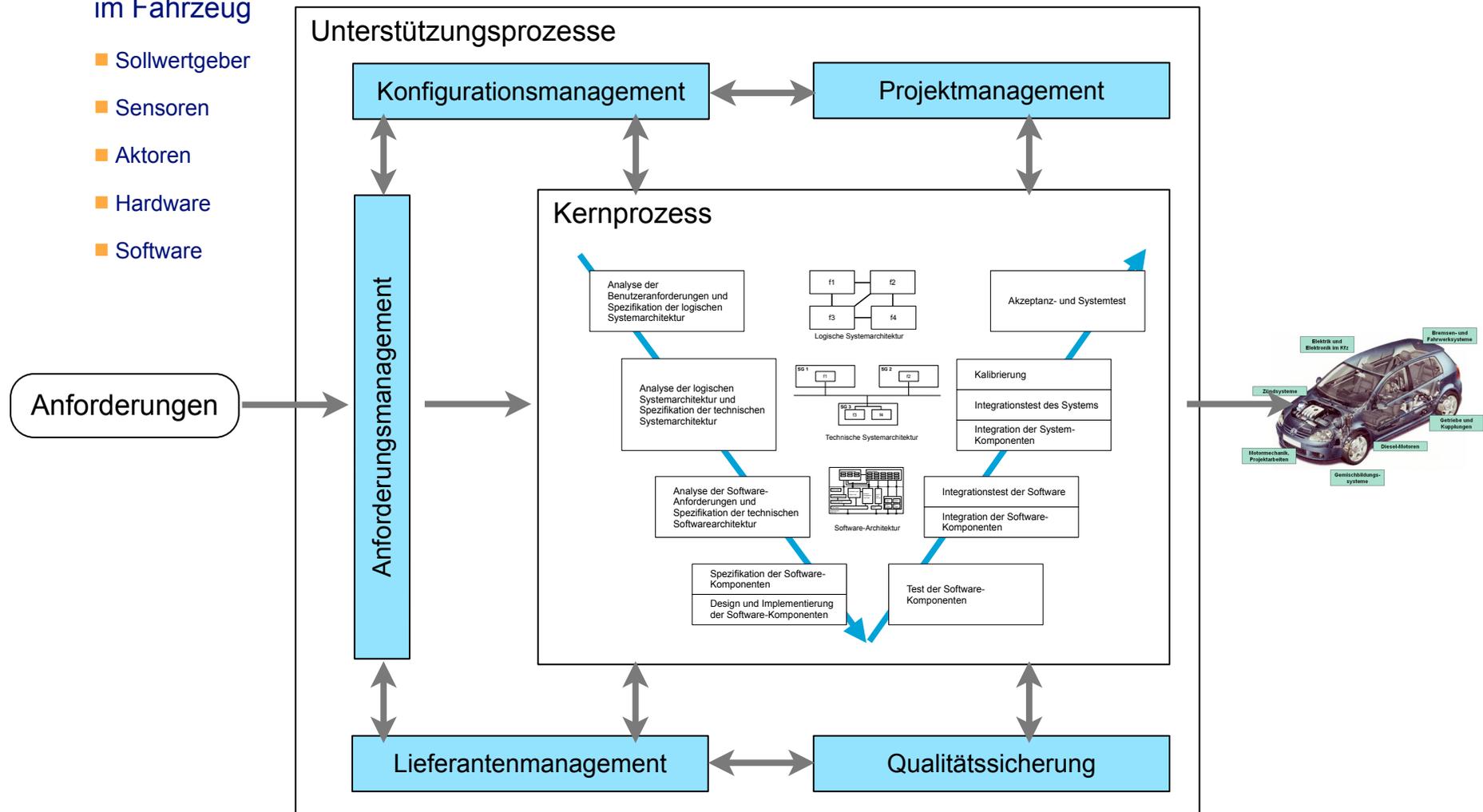
Kapitel 3



Unterstützungsprozesse für die Entwicklung von elektronischen Systemen und Software



- Weitgehend unabhängig von Software
- Anwendbar auf allen Systemebenen im Fahrzeug
- Sollwertgeber
- Sensoren
- Aktoren
- Hardware
- Software



6. SW-Entwicklung / 2. Unterstützungsprozesse

Unterstützungsprozesse für die Embedded Software Entwicklung



1. Vorgehensmodelle und Standards

2. Konfigurationsmanagement

3. Projektmanagement

4. Lieferantenmanagement

5. Anforderungsmanagement

6. Qualitätssicherung

- CMMI Capability Maturity Model Integration
 - <http://www.sei.cmu.edu/cmmi>
- SPICE Software Process Improvement and Capability Determination (ISO/IEC 15504-1)
 - <http://www.sqi.gu.edu.au/spice/>
- V-Modell
 - V-Modell 1997
 - <http://www.v-modell.iabg.de>
 - V-Modell XT 2005
 - http://www.CIO.bund.de/DE/IT-Methoden/V-Modell_XT/v-modell_xt_node.html
 - flyXT
 - Unternehmensspezifische Anpassung des V-Modells XT bei EADS DE
 - Nicht öffentlich

V-Modell und CMMI

- V-Modell:
Tätigkeitsbereiche

- Systemerstellung
- Projektmanagement
- Konfigurationsmanagement
- Qualitätssicherung

- CMMI Reifegradstufe 2:
Key Process Areas

- Anforderungsmanagement
- Konfigurationsmanagement
- Qualitätssicherung
- Projektplanung
- Projektverfolgung
- Lieferantenmanagement

- In dieser Vorlesung:
Unterstützungsprozesse

- Konfigurationsmanagement
- Projektmanagement
- Lieferantenmanagement
- Anforderungsmanagement
- Qualitätssicherung

V-Modell und CMMI

- V-Modell:
Tätigkeitsbereiche
 - Systemerstellung
 - Projektmanagement
 - **Konfigurationsmanagement**
 - Qualitätssicherung

- CMMI Reifegradstufe 2:
Key Process Areas
 - Anforderungsmanagement
 - **Konfigurationsmanagement**
 - Qualitätssicherung
 - Projektplanung
 - Projektverfolgung
 - Lieferantenmanagement

- In dieser Vorlesung:
Unterstützungsprozesse
 - **Konfigurationsmanagement**
 - Projektmanagement
 - Lieferantenmanagement
 - Anforderungsmanagement
 - Qualitätssicherung

V-Modell und CMMI

- V-Modell:
Tätigkeitsbereiche
 - Systemerstellung
 - **Projektmanagement**
 - Konfigurationsmanagement
 - Qualitätssicherung

- CMMI Reifegradstufe 2:
Key Process Areas
 - Anforderungsmanagement
 - Konfigurationsmanagement
 - Qualitätssicherung
 - **Projektplanung**
 - **Projektverfolgung**
 - Lieferantenmanagement

- In dieser Vorlesung:
Unterstützungsprozesse
 - Konfigurationsmanagement
 - **Projektmanagement**
 - Lieferantenmanagement
 - Anforderungsmanagement
 - Qualitätssicherung

V-Modell und CMMI

- V-Modell:
Tätigkeitsbereiche

- Systemerstellung
- Projektmanagement
- Konfigurationsmanagement
- Qualitätssicherung

- CMMI Reifegradstufe 2:
Key Process Areas

- Anforderungsmanagement
- Konfigurationsmanagement
- Qualitätssicherung
- Projektplanung
- Projektverfolgung
- **Lieferantenmanagement**

- In dieser Vorlesung:
Unterstützungsprozesse

- Konfigurationsmanagement
- Projektmanagement
- **Lieferantenmanagement**
- Anforderungsmanagement
- Qualitätssicherung

V-Modell und CMMI

- V-Modell:
Tätigkeitsbereiche
 - Systemerstellung
 - Projektmanagement
 - Konfigurationsmanagement
 - Qualitätssicherung

- CMMI Reifegradstufe 2:
Key Process Areas
 - **Anforderungsmanagement**
 - Konfigurationsmanagement
 - Qualitätssicherung
 - Projektplanung
 - Projektverfolgung
 - Lieferantenmanagement

- In dieser Vorlesung:
Unterstützungsprozesse
 - Konfigurationsmanagement
 - Projektmanagement
 - Lieferantenmanagement
 - **Anforderungsmanagement**
 - Qualitätssicherung

V-Modell und CMMI

- V-Modell:
Tätigkeitsbereiche
 - Systemerstellung
 - Projektmanagement
 - Konfigurationsmanagement
 - **Qualitätssicherung**

- CMMI Reifegradstufe 2:
Key Process Areas
 - Anforderungsmanagement
 - Konfigurationsmanagement
 - **Qualitätssicherung**
 - Projektplanung
 - Projektverfolgung
 - Lieferantenmanagement

- In dieser Vorlesung:
Unterstützungsprozesse
 - Konfigurationsmanagement
 - Projektmanagement
 - Lieferantenmanagement
 - Anforderungsmanagement
 - **Qualitätssicherung**

BMW Group Standard embedded Software (GSeSW)



- Development processes
 - Software Development
 - Software Modification
 - Design Verification process
 - Testprocess
- Organizational processes
 - Project Management
 - Risk Management
 - Role Assignment
 - Documentation
 - Training and Qualification
 - Planning Training and Qualification
- Supporting processes
 - Configuration Management
 - Quality Assurance
 - Problem-solving and Change Management
- In dieser Vorlesung:
Unterstützungsprozesse
 - Konfigurationsmanagement
 - Projektmanagement
 - Lieferantenmanagement
 - Anforderungsmanagement
 - Qualitätssicherung

BMW Group Standard embedded Software (GSeSW)



- Development processes
 - Software Development
 - Software Modification
 - Design Verification process
 - Testprocess
- Organizational processes
 - Project Management
 - Risk Management
 - Role Assignment
 - Documentation
 - Training and Qualification
 - Planning Training and Qualification
- Supporting processes
 - **Configuration Management**
 - Quality Assurance
 - Problem-solving and Change Management

- In dieser Vorlesung:
Unterstützungsprozesse
 - **Konfigurationsmanagement**
 - Projektmanagement
 - Lieferantenmanagement
 - Anforderungsmanagement
 - Qualitätssicherung

BMW Group Standard embedded Software (GSeSW)



- Development processes
 - Software Development
 - Software Modification
 - Design Verification process
 - Testprocess
- Organizational processes
 - **Project Management**
 - **Risk Management**
 - Role Assignment
 - Documentation
 - Training and Qualification
 - Planning Training and Qualification
- Supporting processes
 - Configuration Management
 - Quality Assurance
 - Problem-solving and Change Management

■ In dieser Vorlesung: Unterstützungsprozesse

- Konfigurationsmanagement
- **Projektmanagement**
- Lieferantenmanagement
- Anforderungsmanagement
- Qualitätssicherung

BMW Group Standard embedded Software (GSeSW)



- Development processes
 - Software Development
 - Software Modification
 - Design Verification process
 - Testprocess
- Organizational processes
 - Project Management
 - Risk Management
 - Role Assignment
 - Documentation
 - Training and Qualification
 - Planning Training and Qualification
- Supporting processes
 - Configuration Management
 - Quality Assurance
 - Problem-solving and Change Management

■ In dieser Vorlesung: Unterstützungsprozesse

- Konfigurationsmanagement
- Projektmanagement
- **Lieferantenmanagement**
- Anforderungsmanagement
- Qualitätssicherung

BMW Group Standard embedded Software (GSeSW)



- Development processes
 - Software Development
 - Software Modification
 - Design Verification process
 - Testprocess
- Organizational processes
 - Project Management
 - Risk Management
 - Role Assignment
 - Documentation
 - Training and Qualification
 - Planning Training and Qualification
- Supporting processes
 - Configuration Management
 - Quality Assurance
 - Problem-solving and Change Management

■ In dieser Vorlesung: Unterstützungsprozesse

- Konfigurationsmanagement
- Projektmanagement
- Lieferantenmanagement
- **Anforderungsmanagement**
- Qualitätssicherung

BMW Group Standard embedded Software (GSeSW)



- Development processes
 - Software Development
 - Software Modification
 - Design Verification process
 - Testprocess
- Organizational processes
 - Project Management
 - Risk Management
 - Role Assignment
 - Documentation
 - Training and Qualification
 - Planning Training and Qualification
- Supporting processes
 - Configuration Management
 - **Quality Assurance**
 - Problem-solving and Change Management
- In dieser Vorlesung:
Unterstützungsprozesse
 - Konfigurationsmanagement
 - Projektmanagement
 - Lieferantenmanagement
 - Anforderungsmanagement
 - **Qualitätssicherung**

6. SW-Entwicklung / 2. Unterstützungsprozesse

Unterstützungsprozesse für die Embedded Software Entwicklung



1. Vorgehensmodelle und Standards

2. **Konfigurationsmanagement**

1. **Produkt und Lebenszyklus**

2. Varianten und Skalierbarkeit

3. Versionen und Konfigurationen

3. Projektmanagement

4. Lieferantenmanagement

5. Anforderungsmanagement

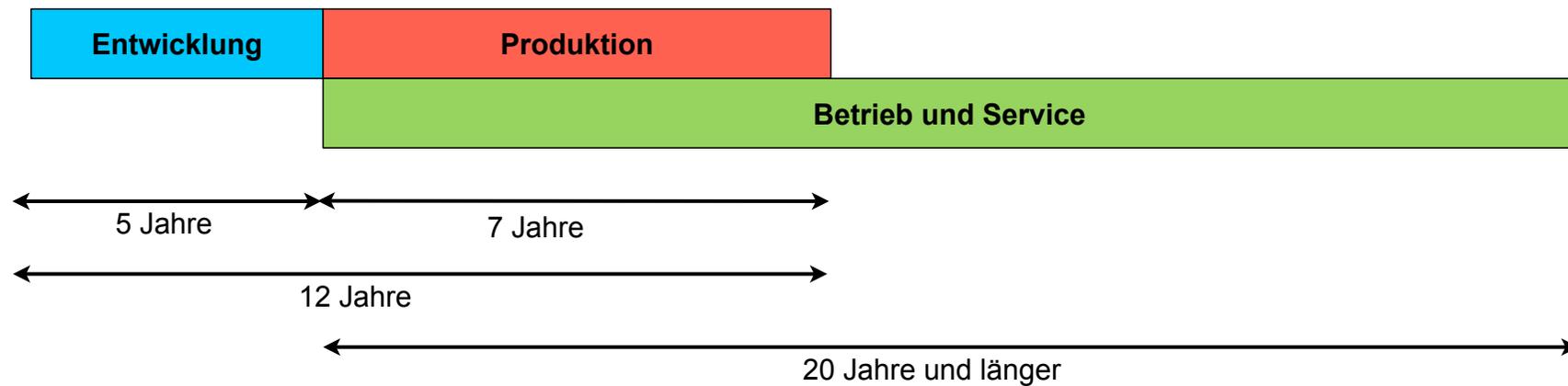
6. Qualitätssicherung

Typischer Produktlebenszyklus eines Fahrzeugs (Nach Schäuuffele, Zurawka)



■ Phasen

- Entwicklung
- Produktion
- Betrieb und Service



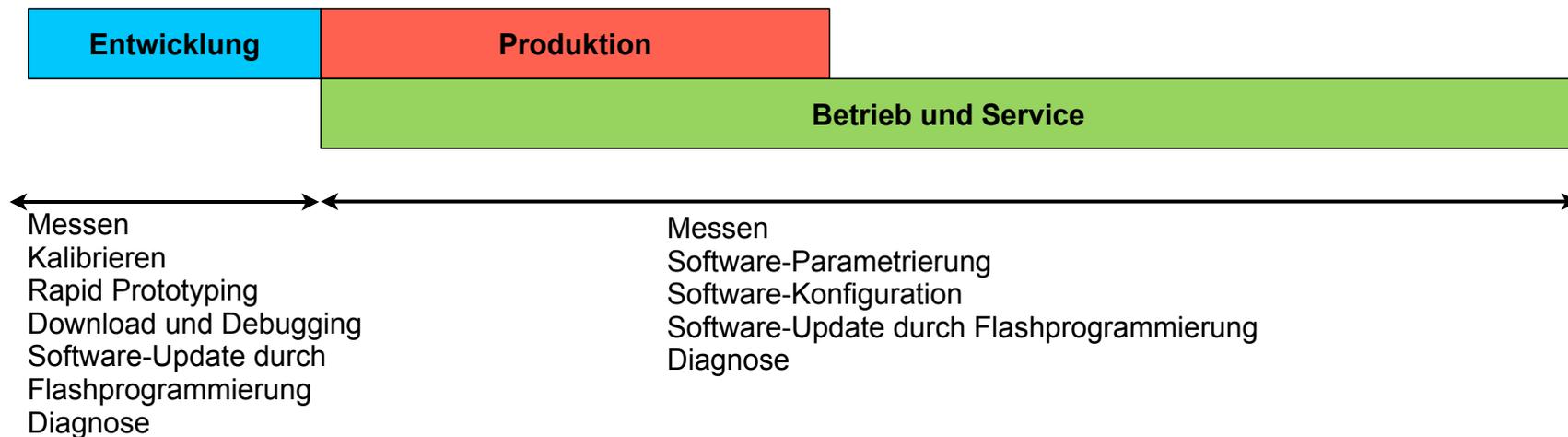
Unterschiedliche Anforderungen an Schnittstellen des Steuergerätes (Nach Schäuffele, Zurawka)



■ Unterschiedliche Lebenszyklen der Komponenten

- Fahrzeug 10 - 20 Jahre
- Steuergeräte 2 Jahre
- Unterhaltungselektronik 6 Monate

■ Unterschiedliche Anforderungen an Schnittstellen des Steuergerätes in Entwicklung, Produktion, Betrieb und Service



6. SW-Entwicklung / 2. Unterstützungsprozesse

Unterstützungsprozesse für die Embedded Software Entwicklung



1. Vorgehensmodelle und Standards

2. Konfigurationsmanagement

1. Produkt und Lebenszyklus

2. Varianten und Skalierbarkeit

3. Versionen und Konfigurationen

3. Projektmanagement

4. Lieferantenmanagement

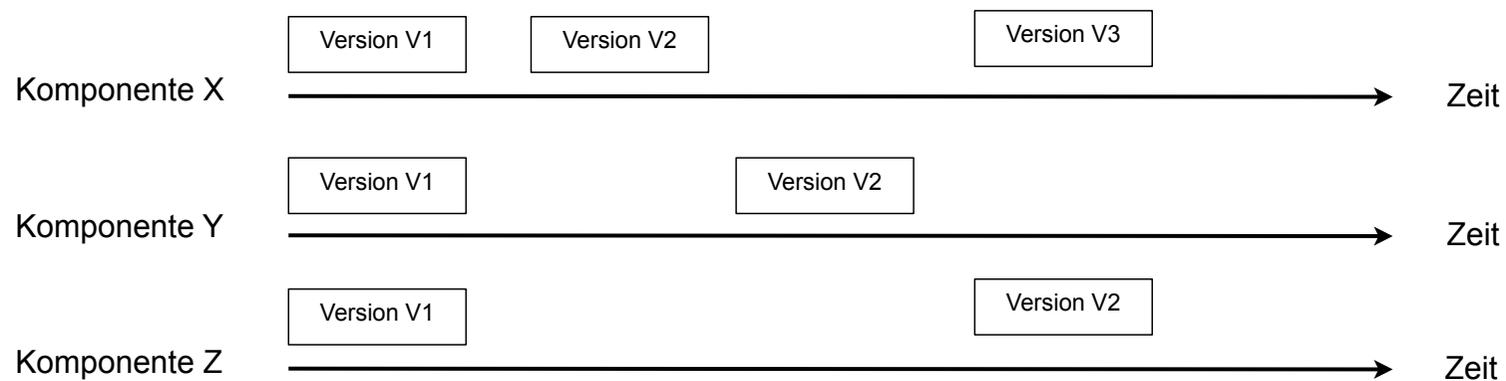
5. Anforderungsmanagement

6. Qualitätssicherung

Versionen (Nach Schäuffele, Zurawka)



- Bestehende elektronische Systeme werden weiterentwickelt
- Zusätzliche elektronische Systeme werden eingeführt
- Komponentenebene
 - Zu bestimmten Zeitpunkten verschiedene Versionen
- Systemebene
 - Zusätzlich Verwaltung der Beziehungen (Referenzen) zu den enthaltenen Komponenten
- Version:
 - Verwaltung der Komponente an sich
 - Änderung der Komponente (Zeit)



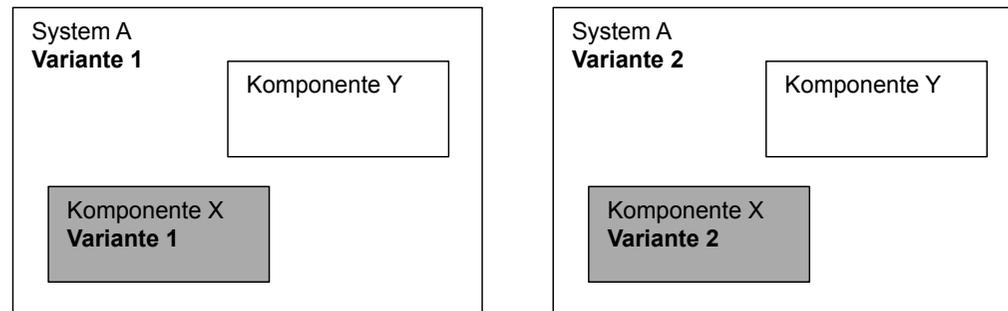
Verschiedene Versionen auf Komponentenebene

Varianten und Skalierbarkeit

Variantenbildung für Komponenten (Nach Schäuuffele, Zurawka)



- Zunehmende Anzahl von Fahrzeugvarianten
- Gestiegene Kundenerwartungen
 - **Individualisierung**
 - Erweiterbarkeit
- **Beispiel**
 - Variante 1:
Automatikgetriebe
 - Variante 2:
Schaltgetriebe



Bildung von Systemvarianten durch Komponentenvarianten

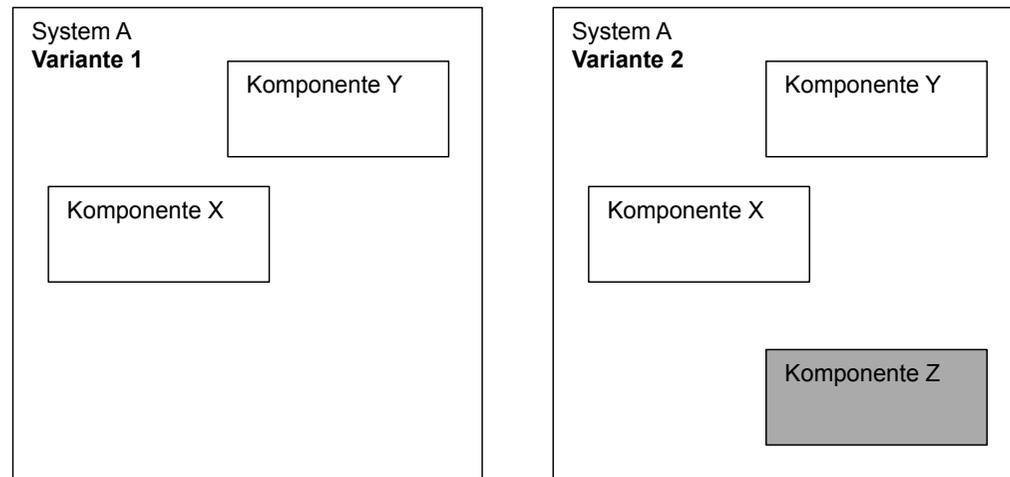
Varianten und Skalierbarkeit

Skalierbare Systemarchitekturen

(Nach Schäuuffele, Zurawka)



- Zunehmende Anzahl von Fahrzeugvarianten
- Gestiegene Kundenerwartungen
 - Individualisierung
 - **Erweiterbarkeit**
- Beispiel
 - Komponente Z: Schiebedach



Bildung von Systemvarianten durch Skalierung

6. SW-Entwicklung / 2. Unterstützungsprozesse

Unterstützungsprozesse für die Embedded Software Entwicklung



1. Vorgehensmodelle und Standards

2. **Konfigurationsmanagement**

1. Produkt und Lebenszyklus

2. Varianten und Skalierbarkeit

3. Versionen und Konfigurationen

3. Projektmanagement

4. Lieferantenmanagement

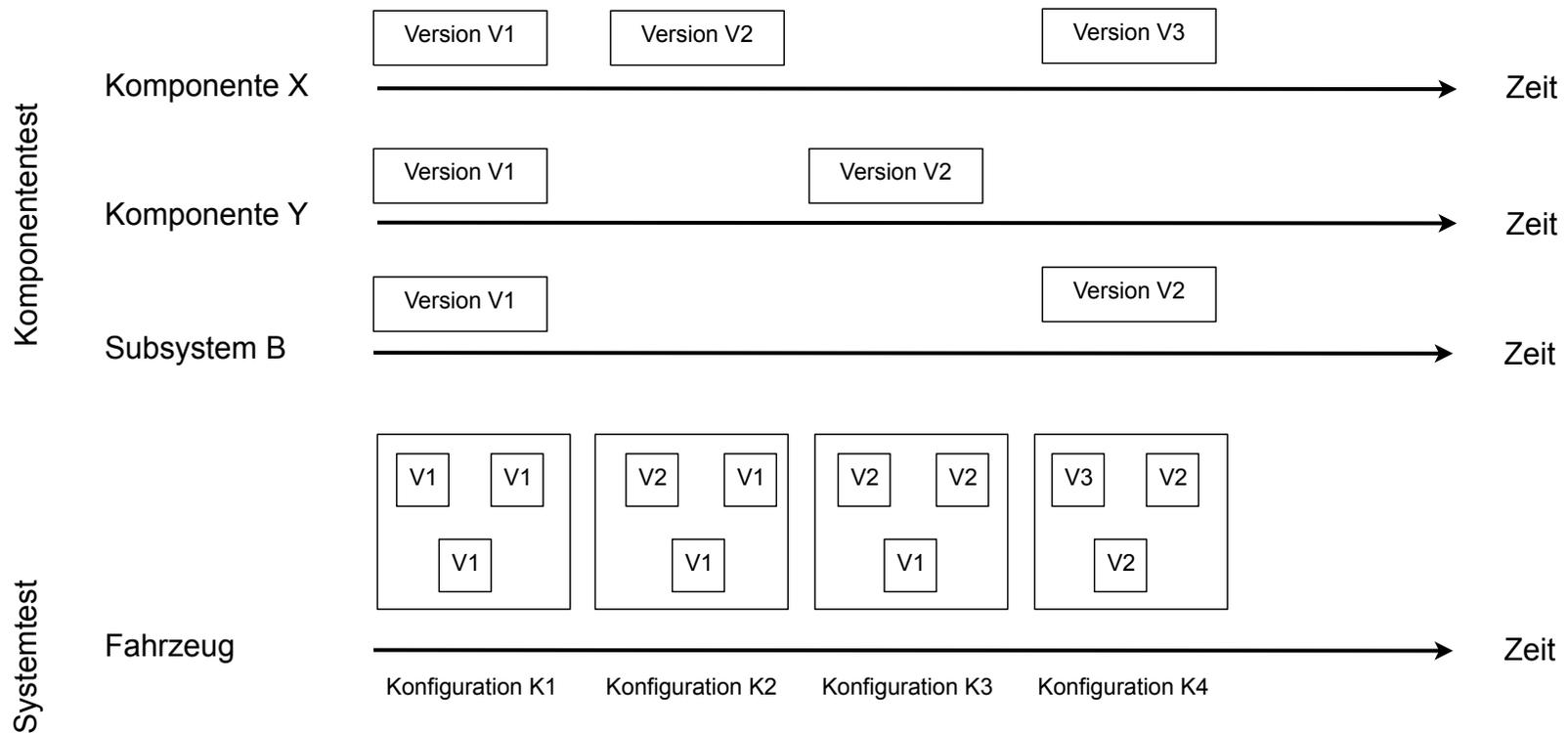
5. Anforderungsmanagement

6. Qualitätssicherung

Integration der System-Komponenten (Nach Schäuuffele, Zurawka)



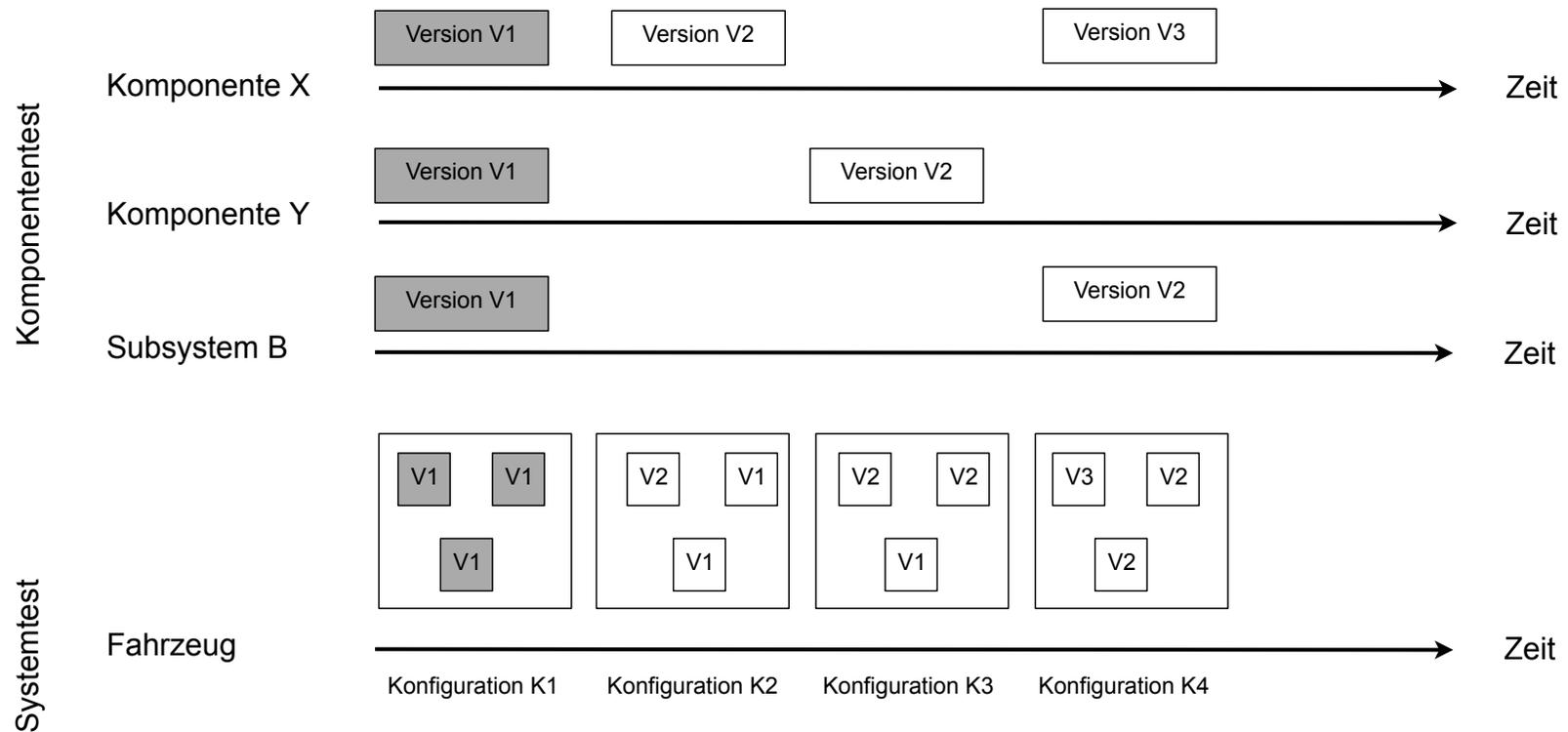
■ Abhängigkeiten zwischen Komponenten- und Systemtest



Integration der System-Komponenten (Nach Schäuffele, Zurawka)



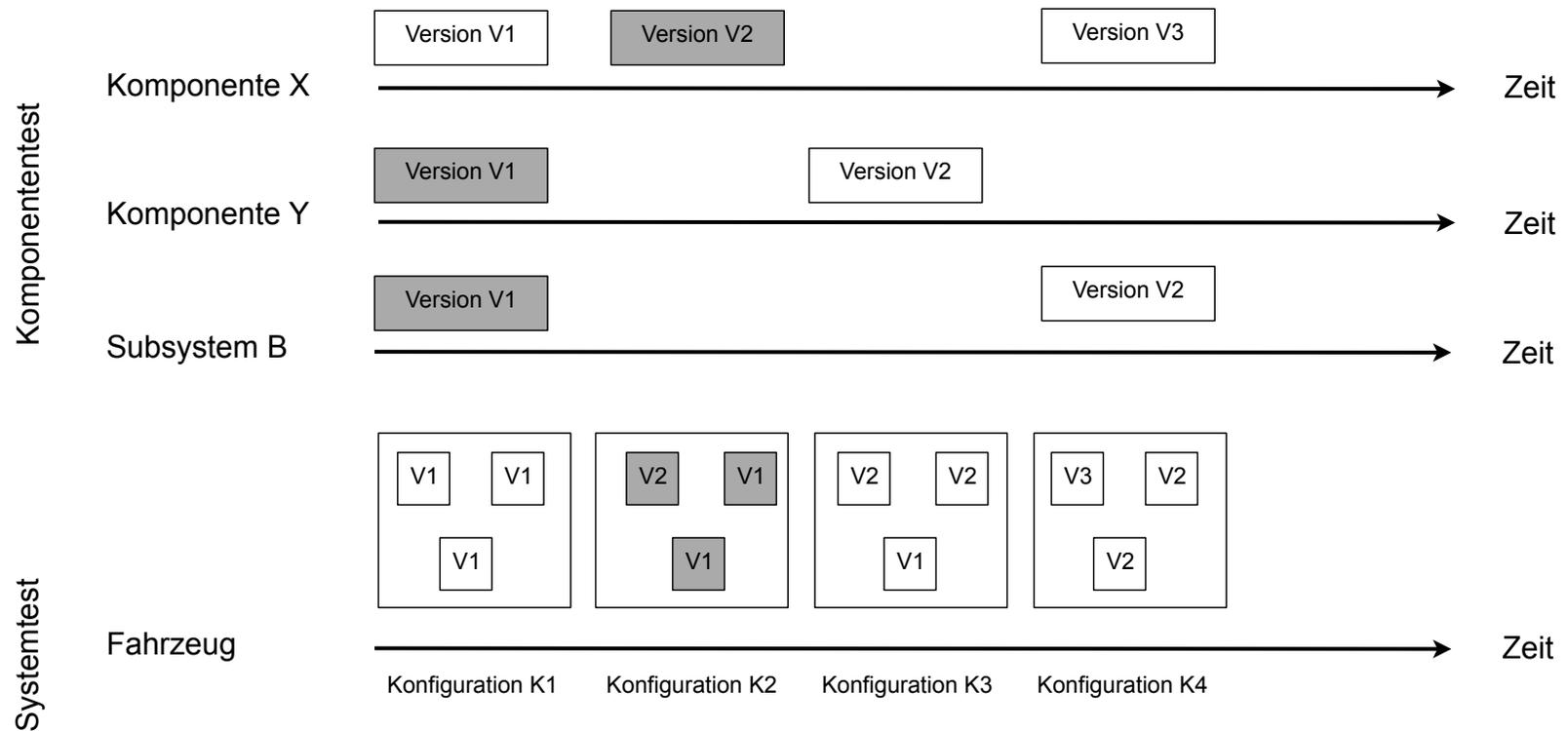
■ Abhängigkeiten zwischen Komponenten- und Systemtest



Integration der System-Komponenten (Nach Schäuffele, Zurawka)



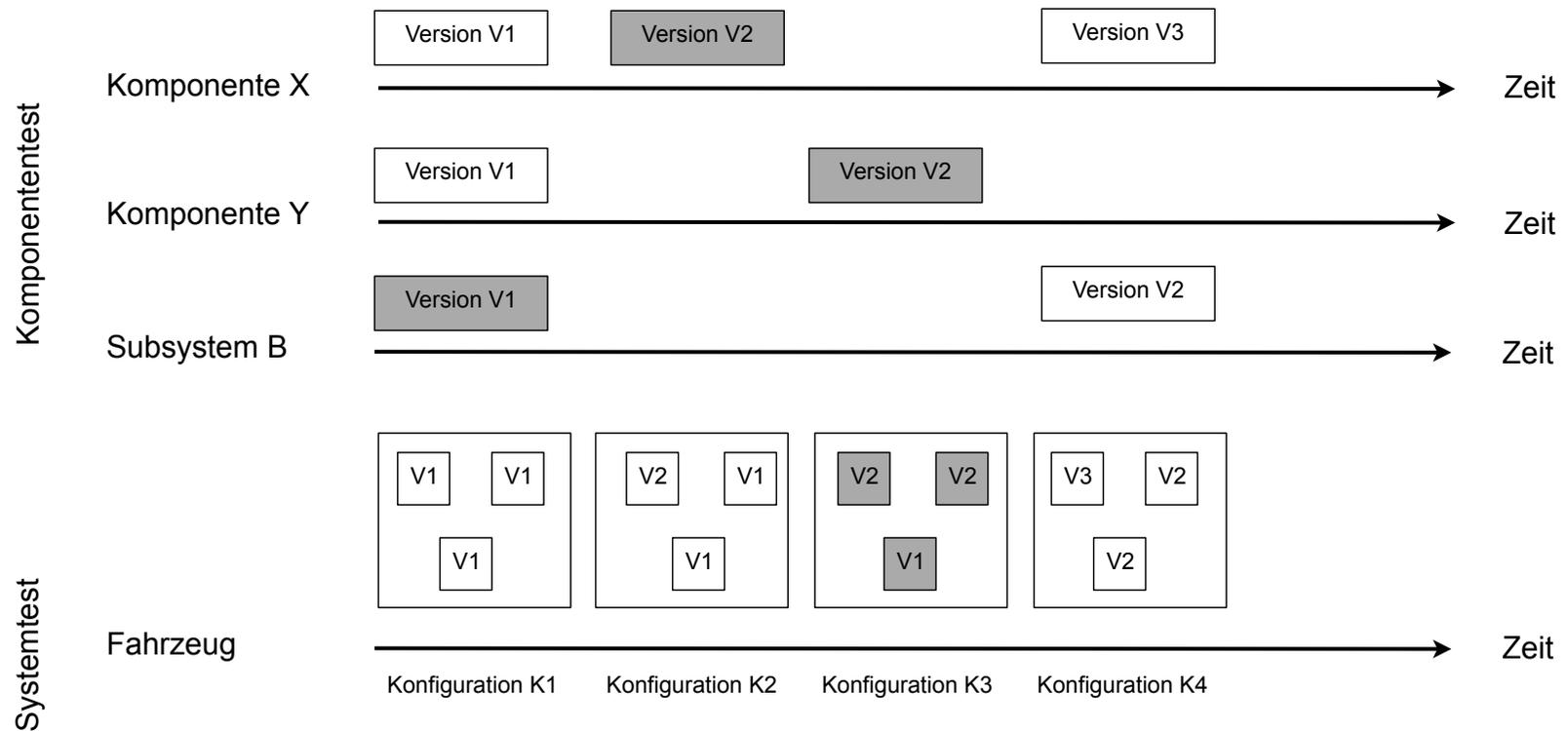
■ Abhängigkeiten zwischen Komponenten- und Systemtest



Integration der System-Komponenten (Nach Schäuffele, Zurawka)



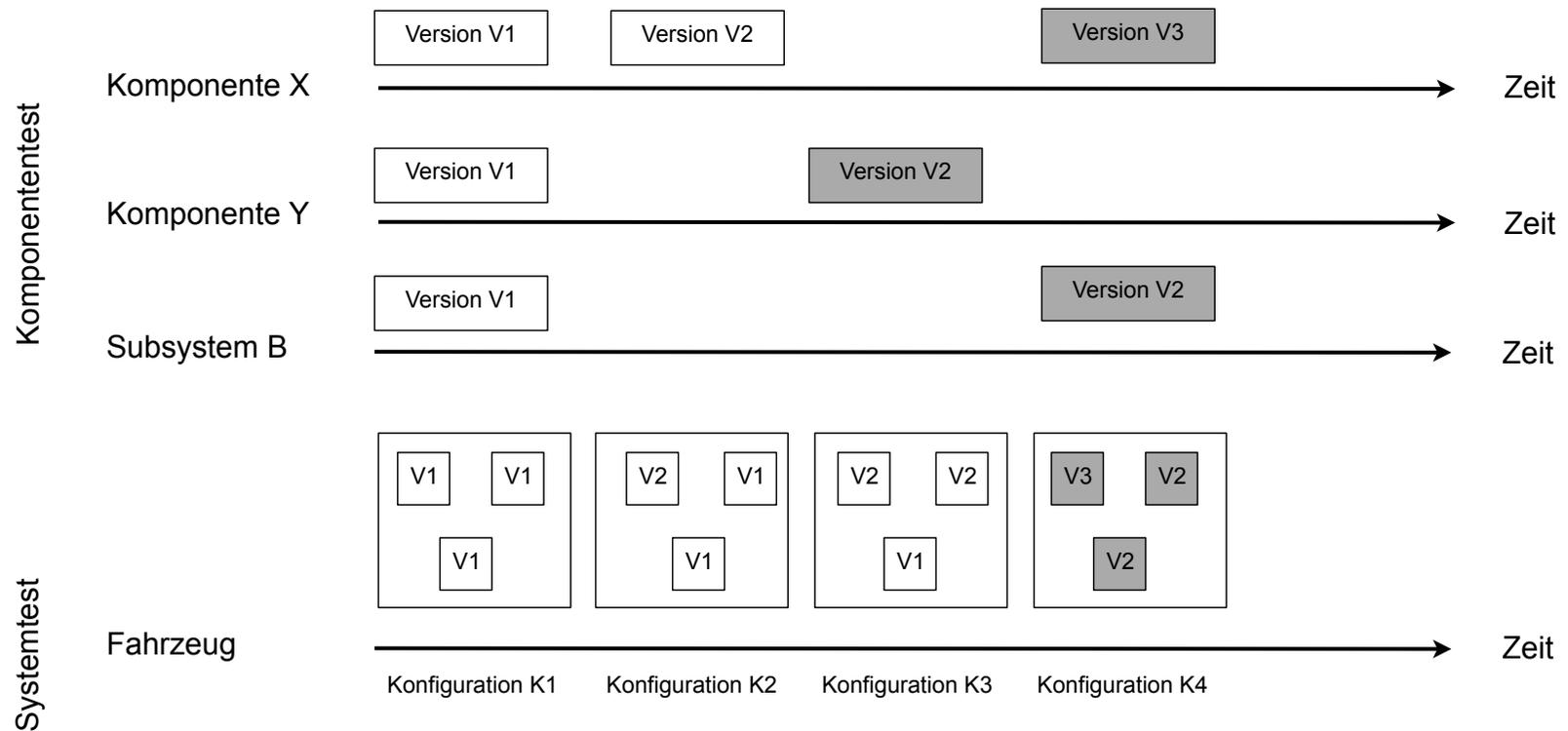
■ Abhängigkeiten zwischen Komponenten- und Systemtest



Integration der System-Komponenten (Nach Schäuuffele, Zurawka)



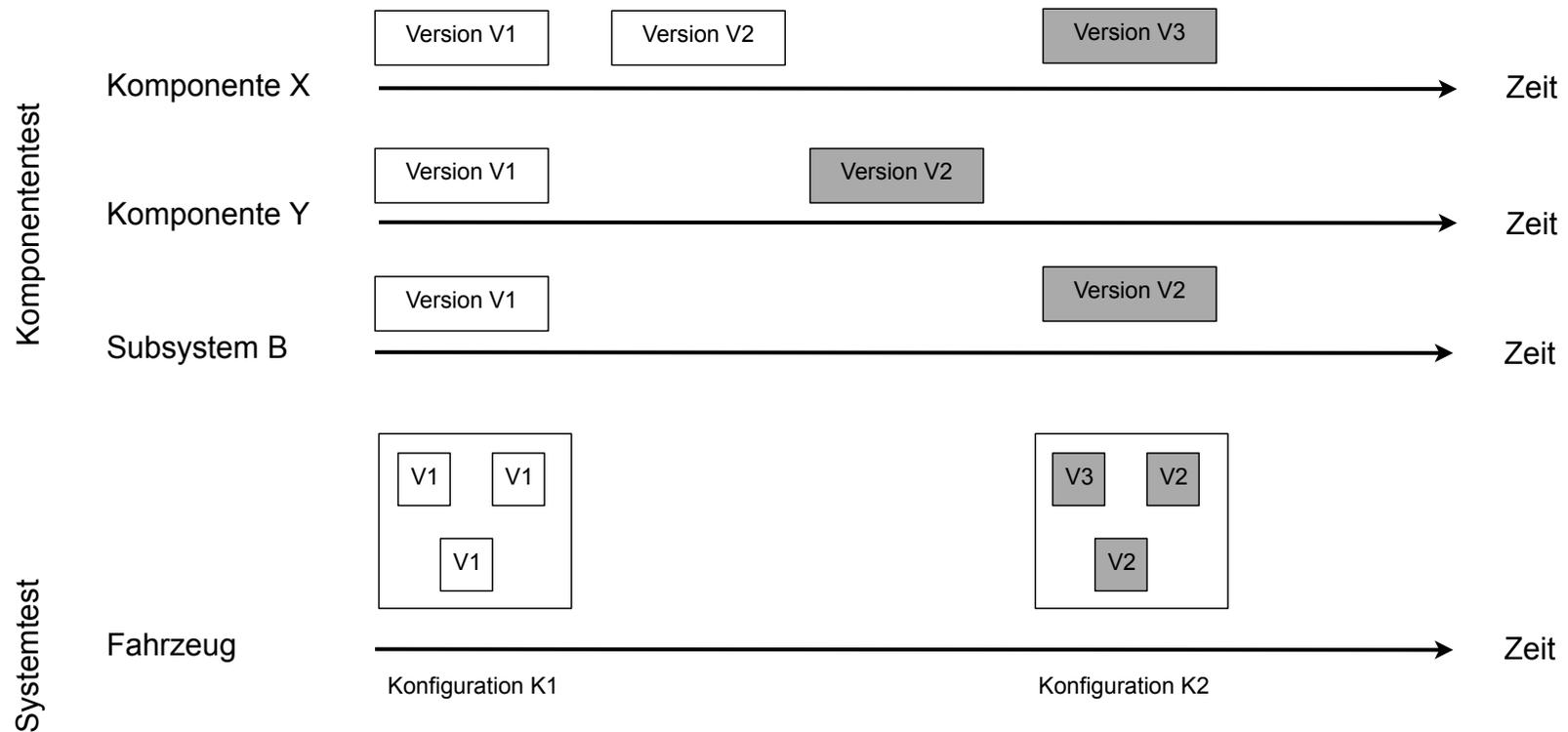
■ Abhängigkeiten zwischen Komponenten- und Systemtest



Integration der System-Komponenten (Nach Schäuuffele, Zurawka)



■ Abhängigkeiten zwischen Komponenten- und Systemtest

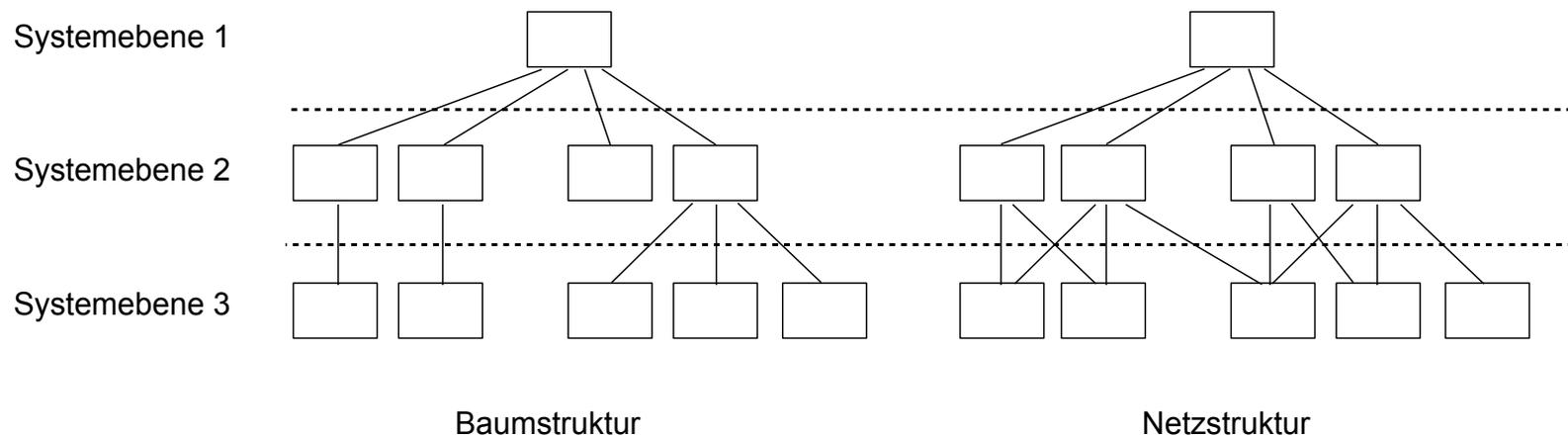


■ In der Praxis meist Bündelung: Weniger Konfigurationen

Versionen und Konfigurationen (Nach Schäuffele, Zurawka)



- Systemvarianten können auf allen Systemebenen auftreten
- Hierarchiebeziehungen
 - Baumstrukturen
 - Jede Komponente gehört zu genau einem System
 - Netzstrukturen
 - Eine Komponente kann zu mehreren Systemen gehören
- Das Versions- und Konfigurationsmanagement (Verwaltung von Versionen und Konfigurationen) geht von Netzstrukturen aus
 - Warum?

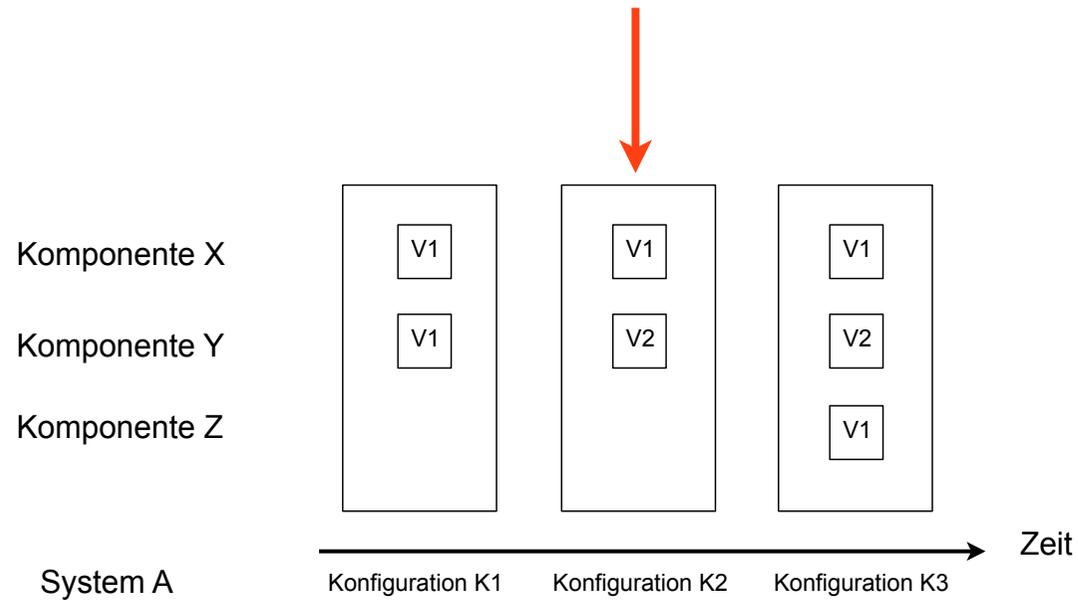


- Eine Konfiguration ist eine versionierbare Komponente, die eine Menge anderer versionierbarer Komponenten referenziert.
- Verwaltung der Beziehungen zu den enthaltenen Komponentenversionen
- Referenzierte Versionen dürfen nicht mehr geändert werden :-)
 - ... müssen über die Lebenszeit der Konfiguration verfügbar sein

Konfiguration (Nach Schäuuffele, Zurawka)



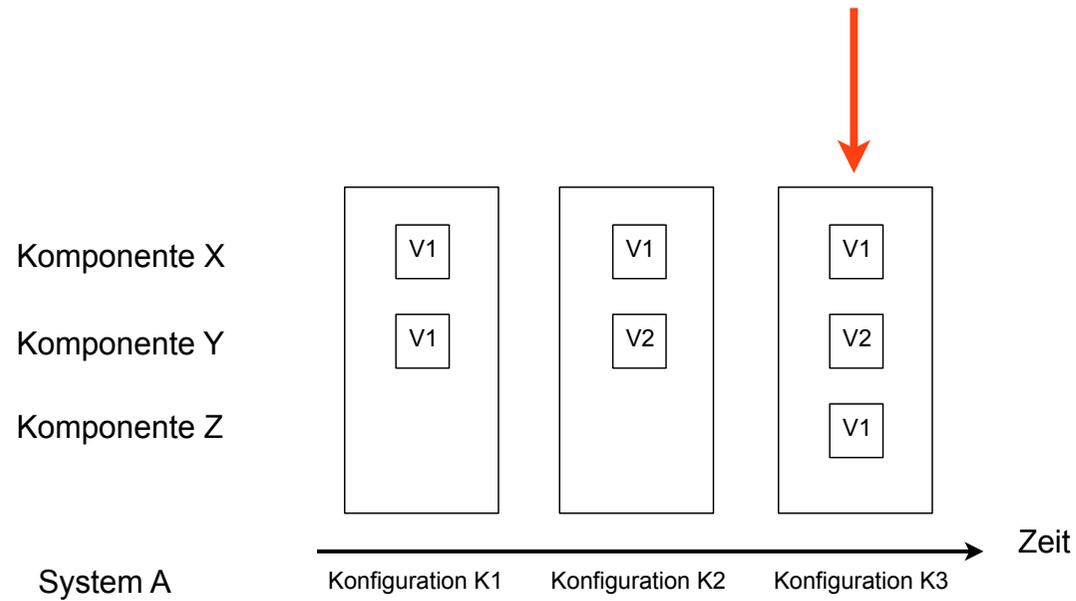
- Versionen einer Konfiguration
 - Versionen der Komponenten
 - Änderung der Hierarchiebeziehungen



Konfiguration (Nach Schäuuffele, Zurawka)



- Versionen einer Konfiguration
 - Versionen der Komponenten
 - **Änderung der Hierarchiebeziehungen**



- **Eigentlich:**
 - Versions- und Konfigurationsmanagement
- **Verwaltet die Beziehungen zwischen Systemen und Komponenten und deren Änderungen**
 - Weiterentwicklungen von Komponenten
 - Änderung der Hierarchien von Systemen
- **Wichtiger Bestandteil von**
 - Entwicklungsprozessen
 - Produktionsprozessen
 - Serviceprozessen
- **Das Konfigurationsmanagement verwaltet**
 - Anforderungen
 - Spezifikationen
 - Implementierungen (Programmstände, Datenstände)
 - Beschreibungsdateien für Diagnose, Software-Update, Software-Parametrierung
 - Dokumentation

6. SW-Entwicklung / 2. Unterstützungsprozesse

Unterstützungsprozesse für die Embedded Software Entwicklung



1. Vorgehensmodelle und Standards

2. Konfigurationsmanagement

3. Projektmanagement

1. **Projektplanung**

2. Projektverfolgung und Risikomanagement

4. Lieferantenmanagement

5. Anforderungsmanagement

6. Qualitätssicherung

- Als Projekte werden Aufgabenstellungen bezeichnet, die durch folgende Merkmale gekennzeichnet sind
 - Aufgabenstellung mit Risiko und einer gewissen Einmaligkeit - keine Routineaufgaben
 - Eindeutige Aufgabenstellung
 - Verantwortung und Zielsetzung für ein zu lieferndes Gesamtergebnis
 - Zeitliche Befristung mit klarem Anfangs- und Endtermin
 - Begrenzter Ressourceneinsatz
 - Besondere auf das Projekt abgestimmte Organisation
 - Häufig: Teilaufgaben und Organisationseinheiten
 - Verschiedenartig
 - Untereinander verbunden
 - Wechselseitig voneinander abhängig

Projektziele



- **Qualitätsziele**

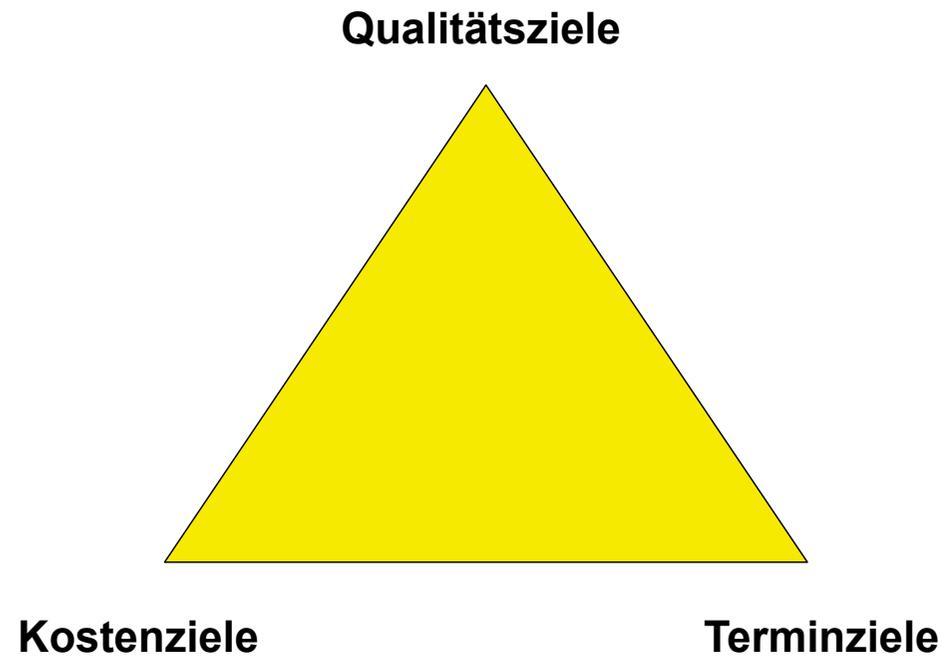
- Welche Anforderungen sollen vom Gesamtergebnis erfüllt werden?

- **Kostenziele**

- Wie viel darf die Erarbeitung des Gesamtergebnisses kosten?

- **Terminziele**

- Wann soll das Gesamtergebnis vorliegen?



Projektmanagement

Projektplanung



- Planung der Umsetzung der Projektziele
 - Qualitätsplanung
 - Kostenplanung
 - Terminplanung
- Organisationsplanung
- Einsatzplanung
- Risikoanalyse

6. SW-Entwicklung / 2. Unterstützungsprozesse

Unterstützungsprozesse für die Embedded Software Entwicklung



1. Vorgehensmodelle und Standards

2. Konfigurationsmanagement

3. Projektmanagement

1. Projektplanung

2. Projektverfolgung und Risikomanagement

4. Lieferantenmanagement

5. Anforderungsmanagement

6. Qualitätssicherung

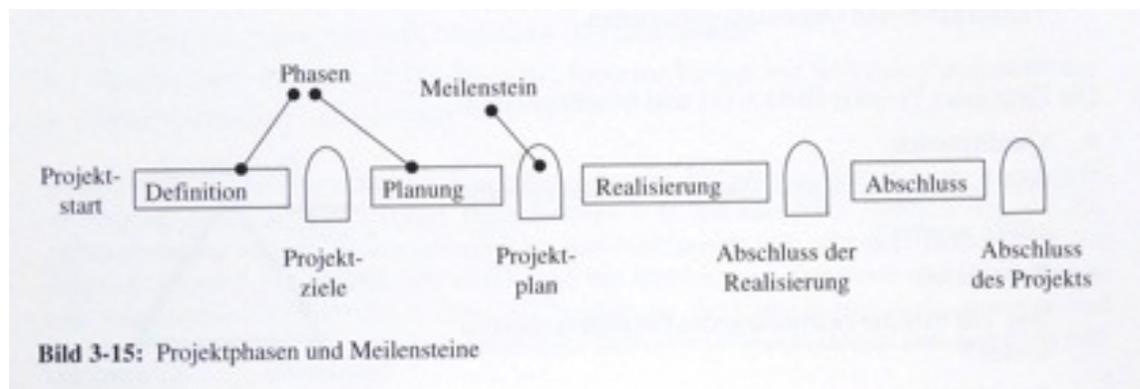
Projektmanagement

Projektverfolgung / Projektsteuerung



- Verfolgung und Überwachung von
 - Qualität
 - Kosten
 - Terminen
- Risikomanagement
 - Beobachtung von auftretenden Risiken und Gegensteuerung

- Definition der Teilaufgaben eines Projektes
- Meilenstein
 - Abschluss einer Teilaufgabe
 - Termin für
 - Teillieferungen
 - Tests
 - Teilzahlungen
- Projektphase
 - Zeitraum, in dem eine Teilaufgabe bearbeitet wird
 - Im allgemeinen vier Projektphasen (die in der Realität weiter untergliedert werden)



■ Qualitätsplanung

- Festlegung der Massnahmen zur Erreichung des Gesamtergebnisses
- Für alle Projektphasen
 - Richtlinien zur Qualitätssicherung
 - Massnahmen zur Qualitätsprüfung
- Phasenübergreifende Zusammenfassung in einem Qualitätsplan

■ Kostenplanung

- Personalkosten
 - Einsatzpläne für Mitarbeiter
- Sachkosten
 - Materialkosten
 - Raumkosten
 - Reisekosten
- Massnahmen
 - Z.B. Wiederverwendung von Ergebnissen aus anderen Projekten

■ Terminplanung

- Festlegung des Zeitraums für die Durchführung der Projektphasen
 - Anfangstermin
 - Endtermin / Meilenstein
- Herausforderungen
 - Einsatz von Mitarbeitern in verschiedenen Projekten
 - Gleichzeitige Durchführung mehrerer Projekte
 - Abhängigkeiten zwischen den Projektphasen
- Abarbeitung
 - Sequentiell bei abhängigen Aufgaben
 - Parallel bei unabhängigen Aufgaben
 - Verkürzung der Entwicklungszeit
 - Simultaneous Engineering
 - Planung und und Synchronisation zeitlich paralleler Entwicklungsschritte

Beispiel: Terminplan für Fahrzeugentwicklung



Abstimmung von Fahrzeug-, Elektronik- und Software-Entwicklung

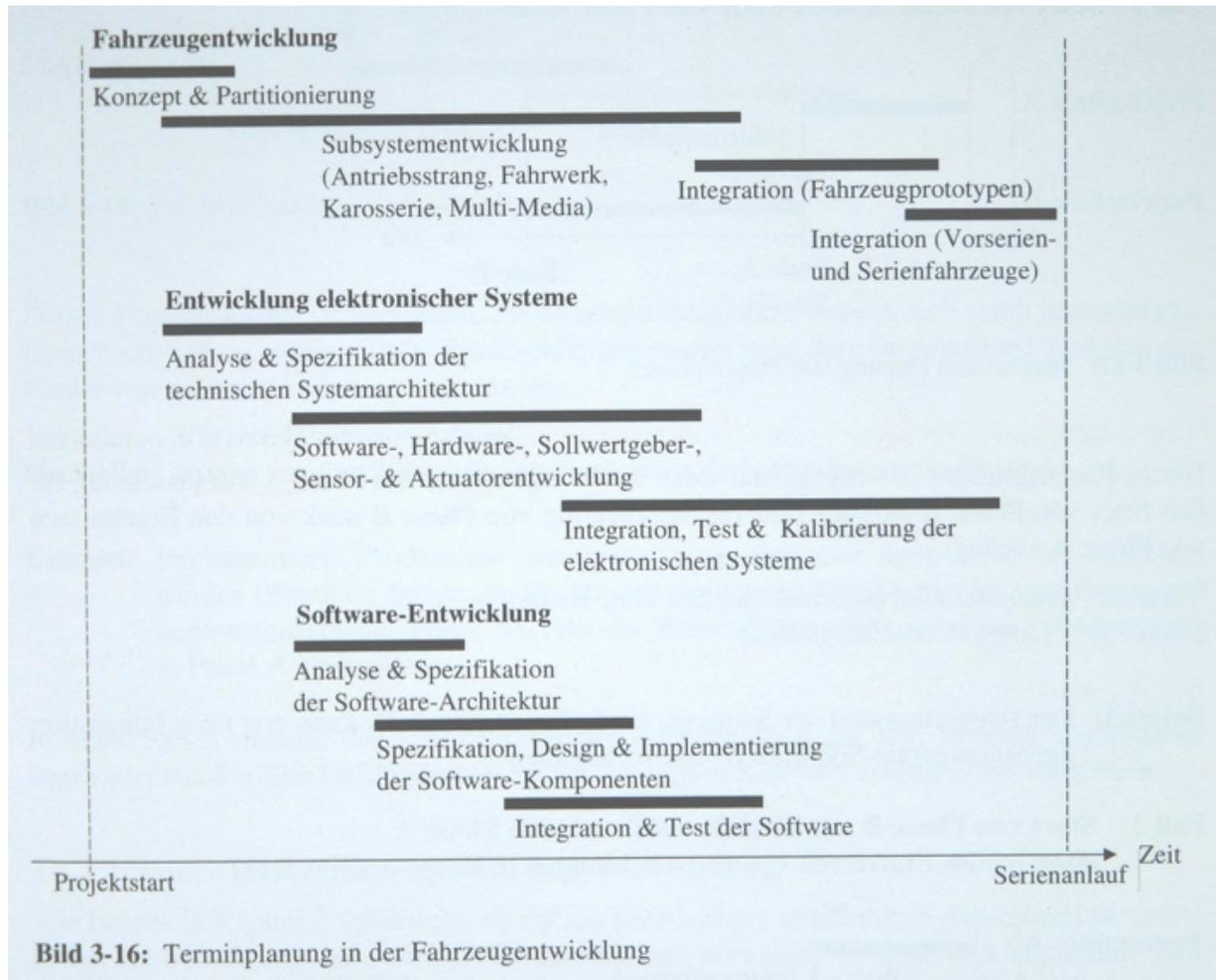
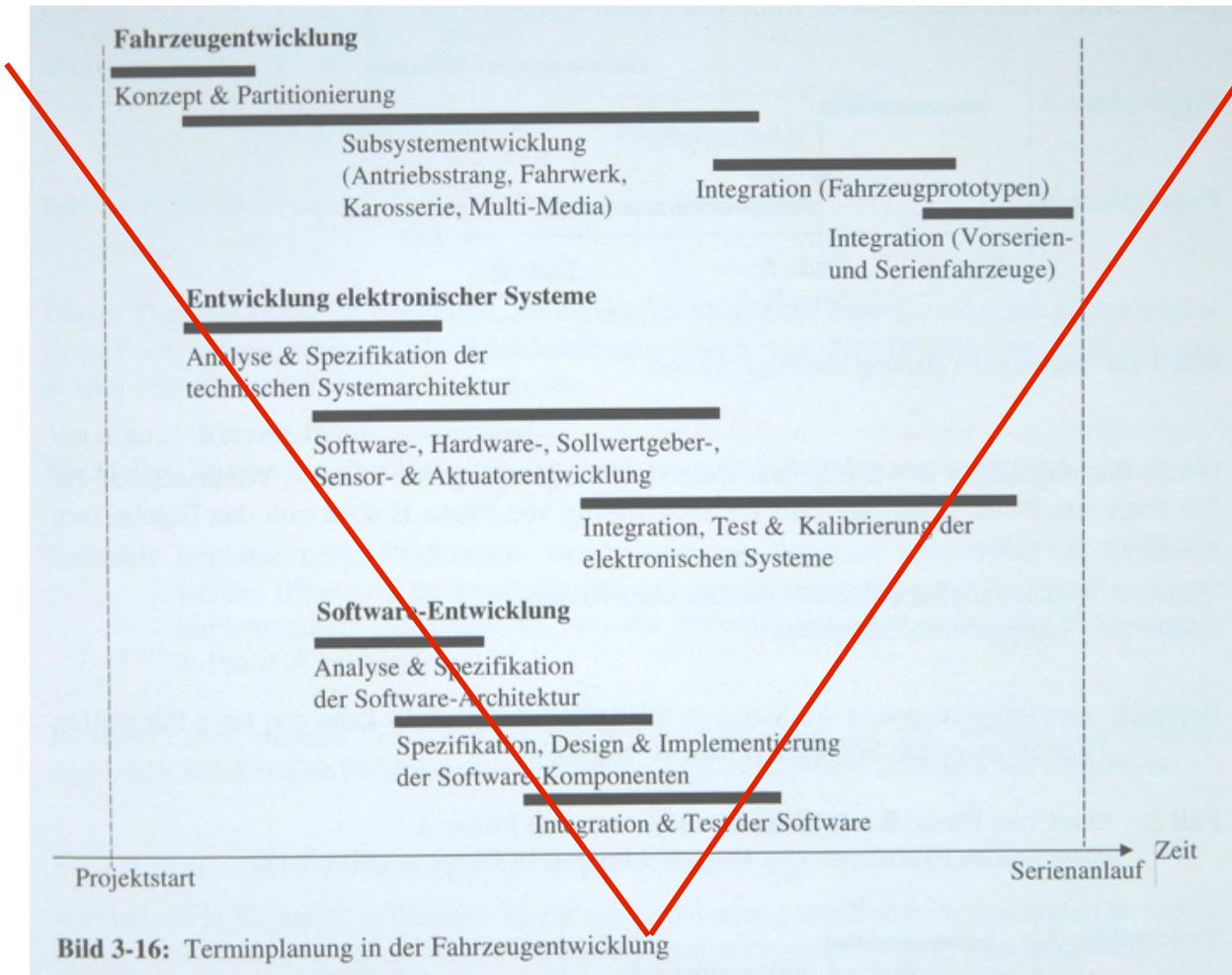


Bild 3-16: Terminplanung in der Fahrzeugentwicklung

Beispiel: Terminplan für Fahrzeugentwicklung



Abstimmung von Fahrzeug-, Elektronik- und Software-Entwicklung



Sequentielle Planung

- Vorteile: Sequentieller Informationsfluss ohne Risiko
- Nachteile: Lange Bearbeitungszeit
- Beispiel: Integrationstest (Phase B) nach Integration (Phase A)

Fall 1: Start von Phase B nach Ende von Phase A (Bild 3-17)

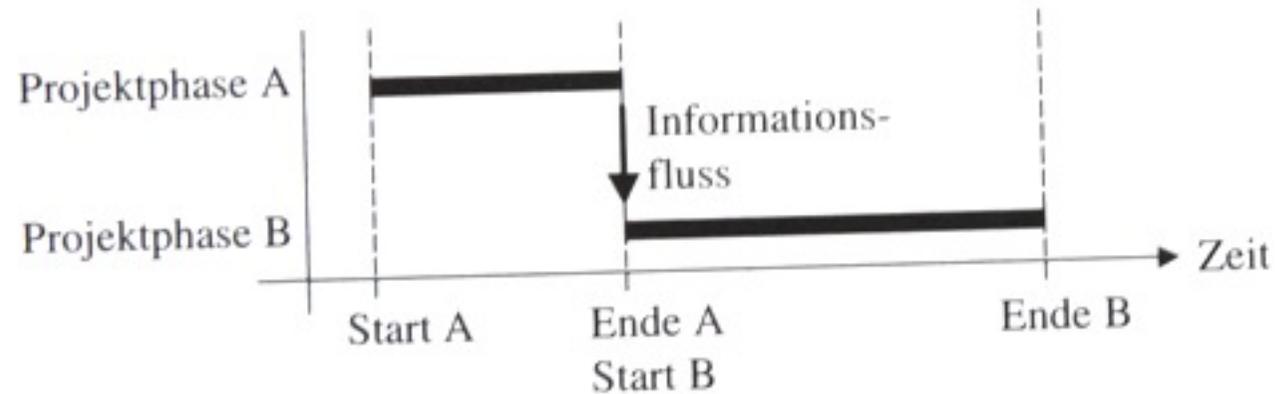


Bild 3-17: Sequentielle Planung von Projektphasen

Parallele Planung ohne Einfrieren

- Vorteile: Kürzere Bearbeitungsdauer
- Nachteile: Risiko von Verzögerungen durch Iterationen in der Phase B
- Beispiel: Anwendungssoftware (Phase B) wird mit Rapid Prototyping System vor Fertigstellung der Plattformsoftware (Phase A) begonnen

Fall 2: Start von Phase B mit Teilinformationen von Phase A ohne frühes Einfrieren von Entscheidungen in Phase A (Bild 3-18)

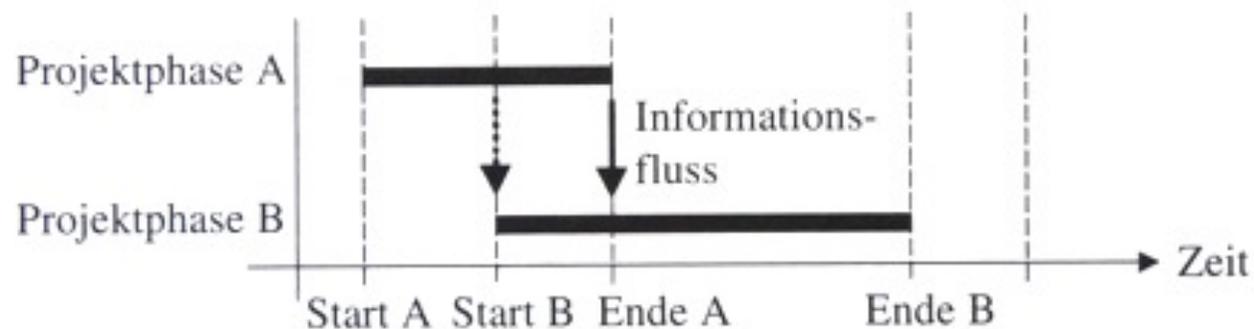


Bild 3-18: Parallele Planung von Projektphasen

Parallele Planung mit Einfrieren

- Vorteile: Kürzere Bearbeitungsdauer
- Nachteile: Risiko von Qualitätseinbussen durch frühe Einschränkungen in der Phase A
- Beispiel: Kalibrierung von implementierten Funktionen der Anwendungssoftware (Phase B) vor Fertigstellung aller Funktionen (Phase A)

Fall 3: Start von Phase B mit Teilinformationen von Phase A mit frühem Einfrieren einiger Entscheidungen in Phase A (Bild 3-19)

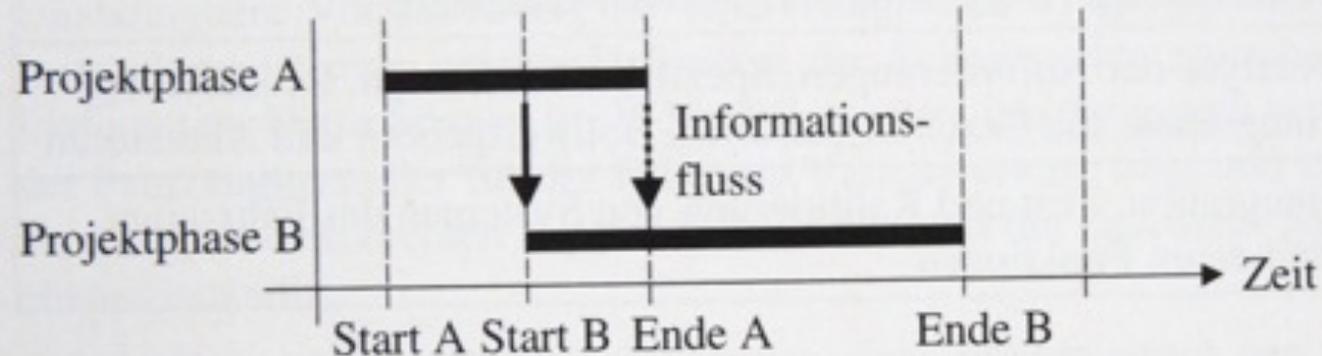


Bild 3-19: Parallele Planung von Projektphasen

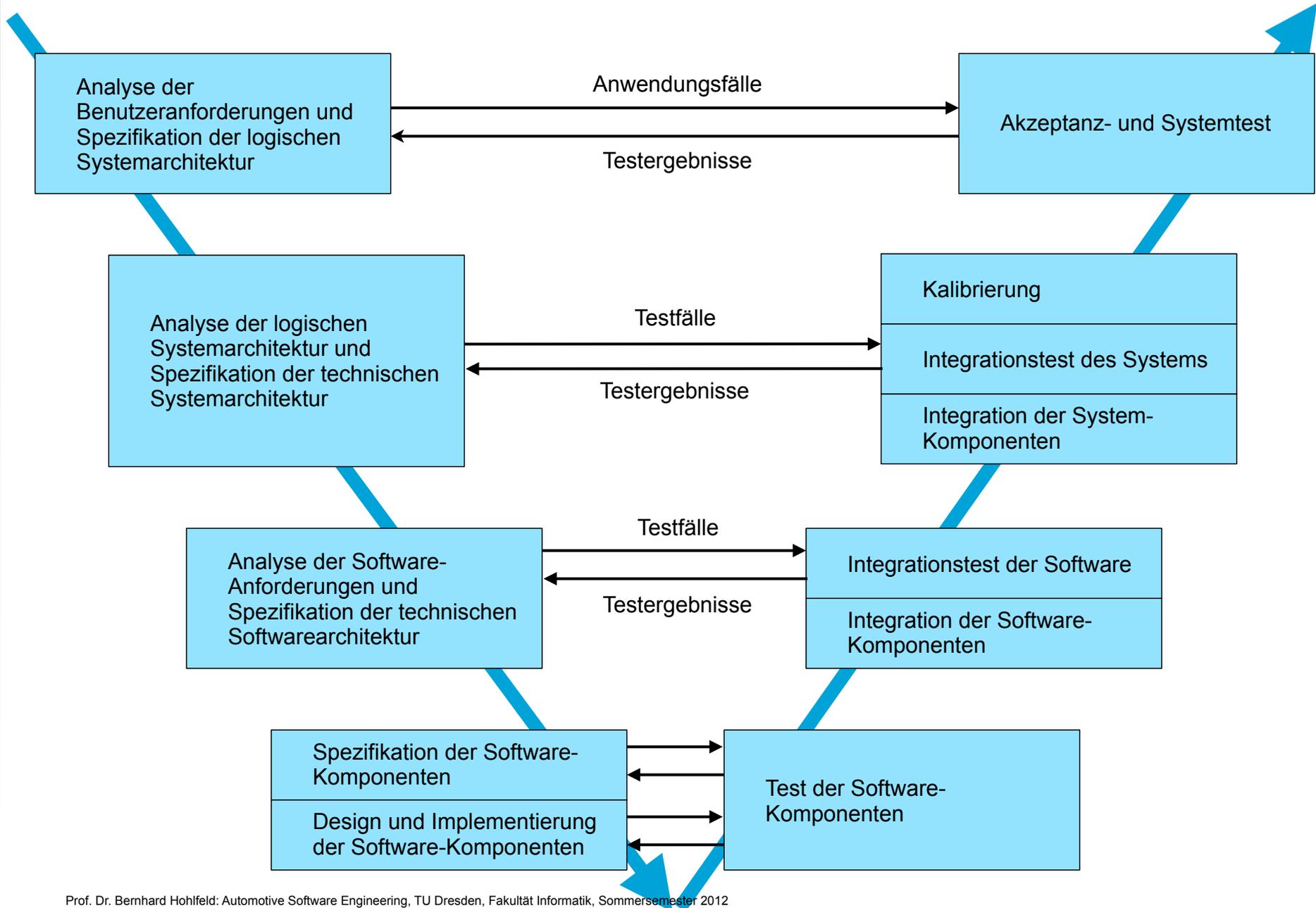
Rollen und Aufgabengebiete in der Entwicklung
siehe 6. SW-Entwicklung / 1. Kernprozess



Tabelle 3-1: Rollen und Aufgabengebiete im Entwicklungsprozess

Rolle	Aufgabengebiet
Funktions-entwicklung	Analyse der Benutzeranforderungen und Spezifikation der logischen Systemarchitektur
System-Entwicklung	Analyse der logischen Systemarchitektur und Spezifikation der technischen Systemarchitektur
Software-Entwicklung	Analyse der Software-Anforderungen, Spezifikation, Design, Implementierung, Integration und Test der Software
Hardware-Entwicklung	Analyse der Hardware-Anforderungen, Spezifikation, Design, Realisierung, Integration und Test der Hardware
Sensor-/Sollwertgeber-/Aktuatorenentwicklung	Analyse der Anforderungen, Spezifikation, Design, Realisierung, Integration und Test von Sensoren, Sollwertgebern und Aktuatoren
Integration, Erprobung und Kalibrierung	Integration, Test und Kalibrierung von Systemen des Fahrzeugs und deren Funktionen

Kernprozess zur Entwicklung von elektronischen Systemen und Software (Nach Schäuuffele, Zurawka)



6. SW-Entwicklung / 2. Unterstützungsprozesse

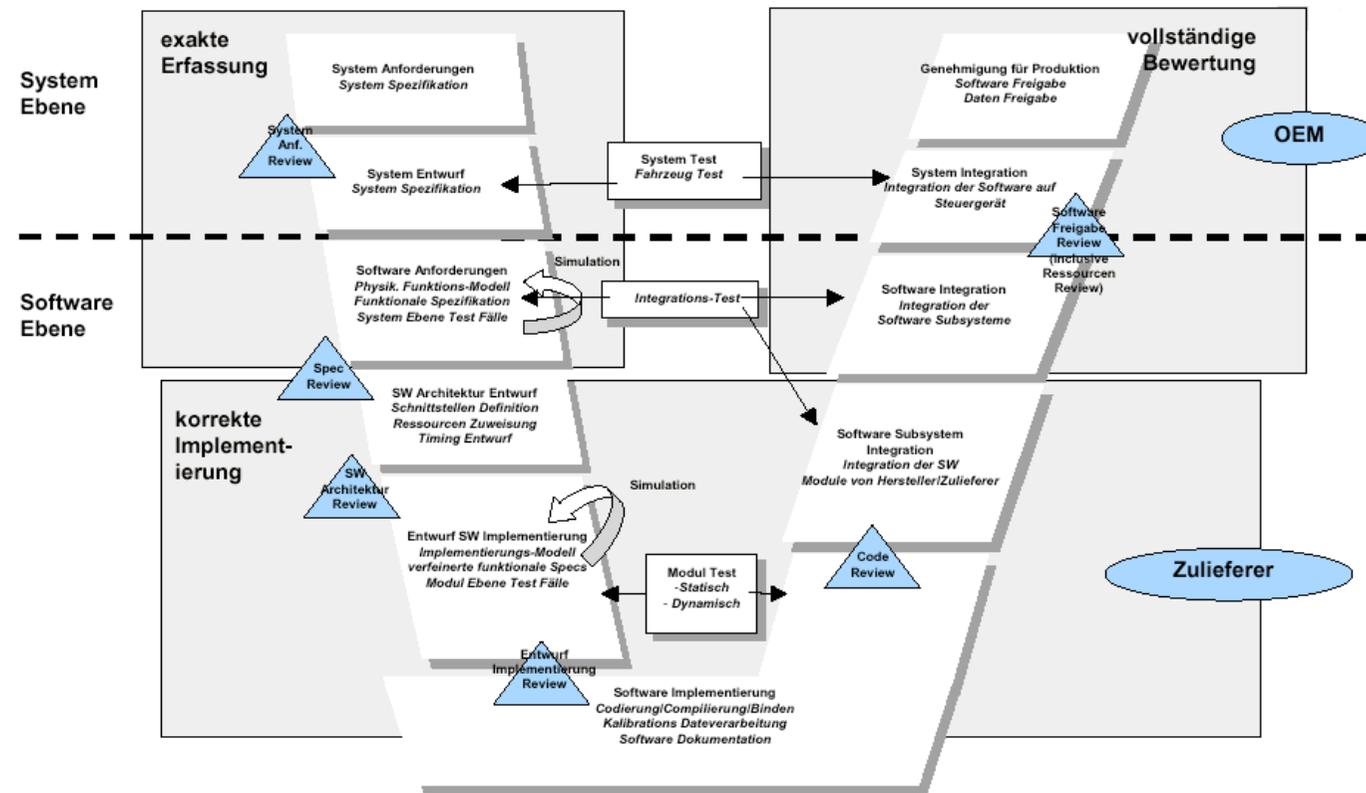
Unterstützungsprozesse für die Embedded Software Entwicklung

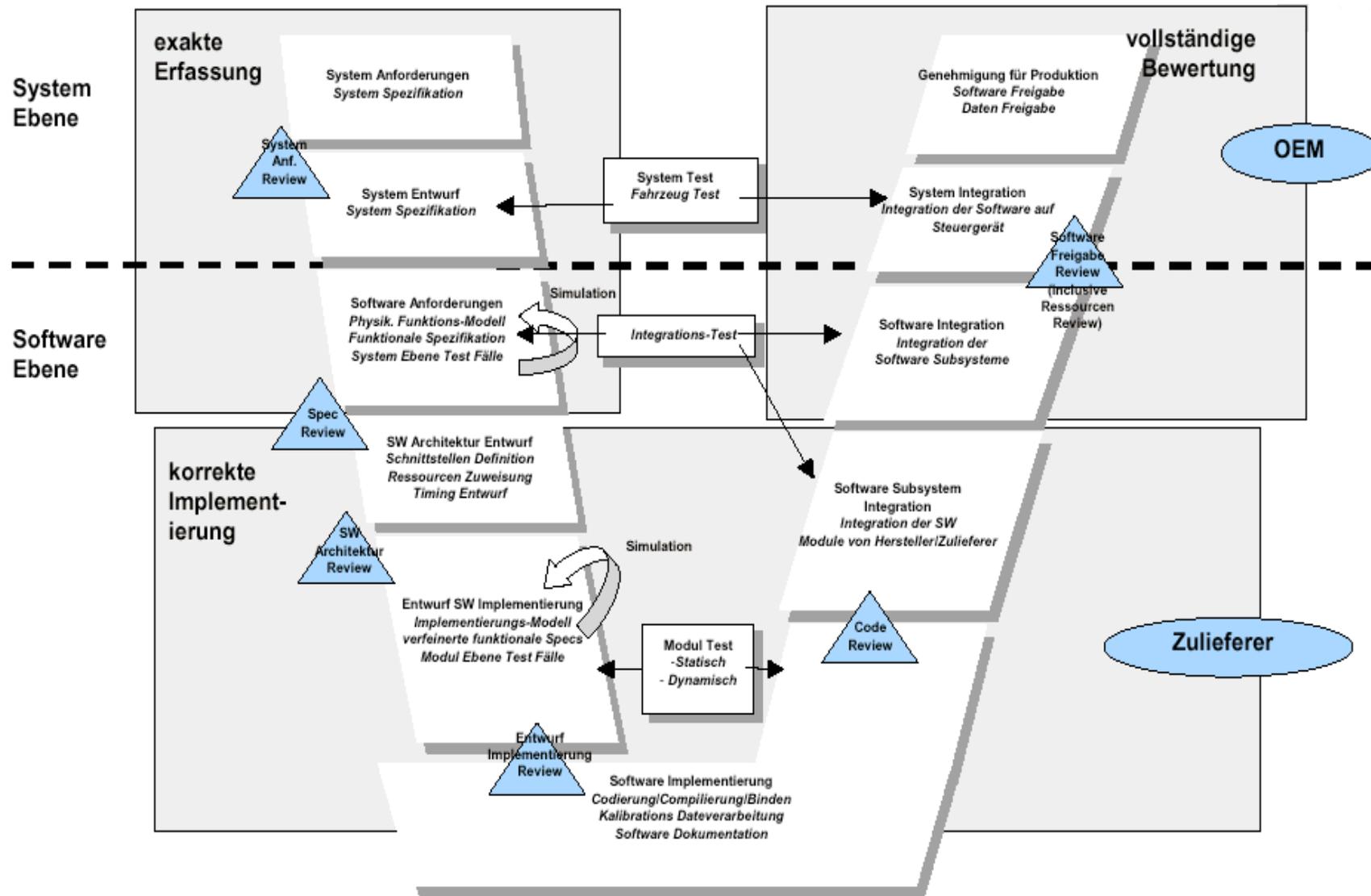


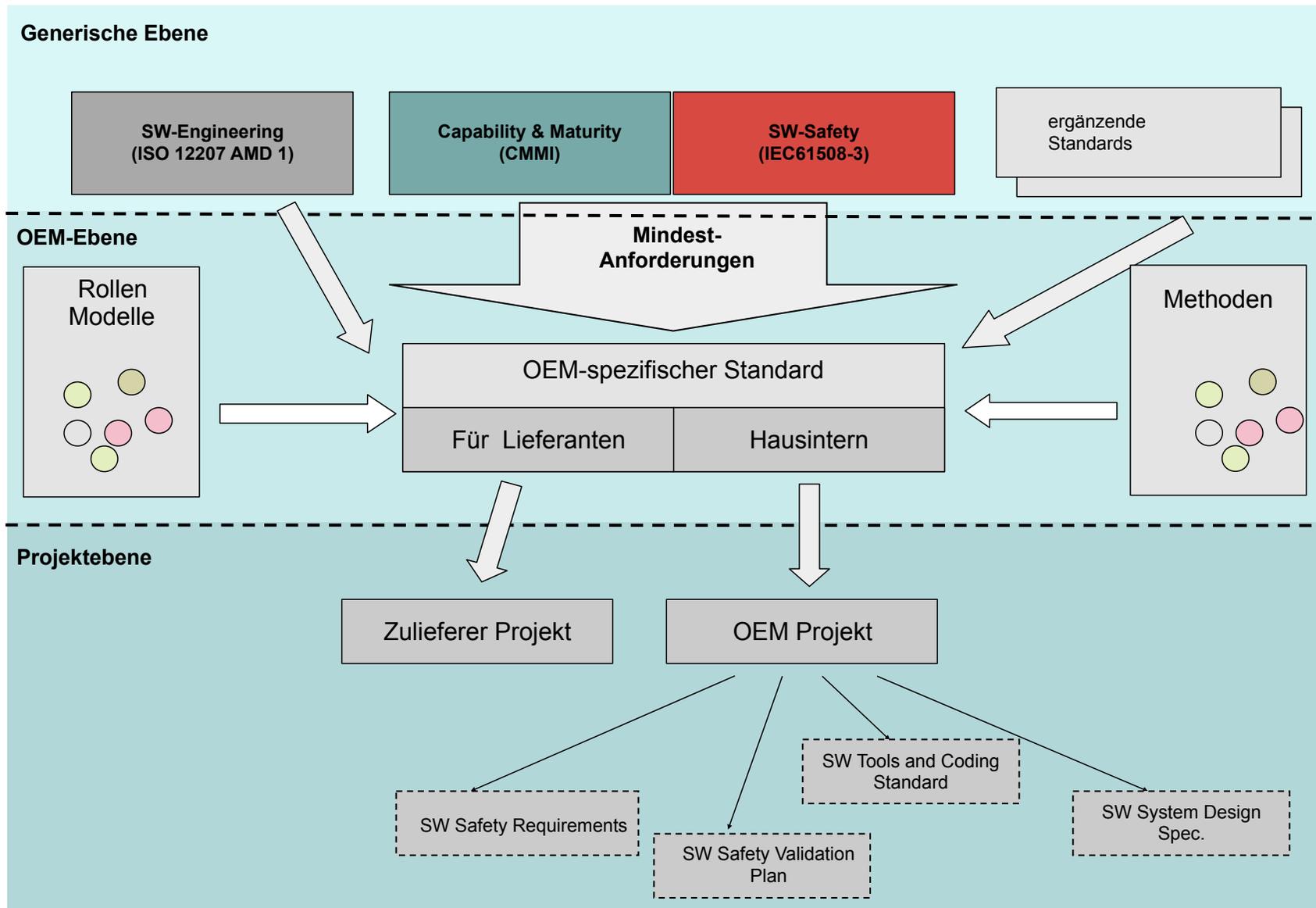
1. Vorgehensmodelle und Standards
2. Konfigurationsmanagement
3. Projektmanagement
- 4. Lieferantenmanagement**
 - 1. System- und Komponentenverantwortung**
 2. Schnittstellen für Spezifikation und Integration
 3. Festlegung des firmenübergreifenden Entwicklungsprozesses
5. Anforderungsmanagement
6. Qualitätssicherung

- Starke Arbeitsteilung zwischen Fahrzeugherstellern und Zulieferern
- Festlegung der Anforderungen an eine zu entwickelnde Funktion
 - Fahrzeughersteller
- Realisierung der Funktion durch elektronische Systeme
 - Zulieferer
- Abstimmung und Abnahme der Funktion im Fahrzeug
 - Fahrzeughersteller
- Firmenübergreifende Zusammenarbeit
 - Technische Aspekte
 - Organisatorische Aspekte
 - Rechtliche Aspekte
- Lieferantenmanagement
 - Alle Aufgaben, die im Rahmen der Systementwicklung an den Schnittstellen zwischen Fahrzeughersteller und Zulieferern beachtet werden müssen

- Präzise Definition der Schnittstellen zwischen Fahrzeughersteller und Zulieferern
- Orientierung am V-Modell
- Fahrzeughersteller
 - Gesamtfahrzeug
- Zulieferer
 - Komponenten







6. SW-Entwicklung / 2. Unterstützungsprozesse

Unterstützungsprozesse für die Embedded Software Entwicklung



1. Vorgehensmodelle und Standards
2. Konfigurationsmanagement
3. Projektmanagement
- 4. Lieferantenmanagement**
 1. System- und Komponentenverantwortung
 - 2. Schnittstellen für Spezifikation und Integration**
 3. Festlegung des firmenübergreifenden Entwicklungsprozesses
5. Anforderungsmanagement
6. Qualitätssicherung

Schnittstellen für Spezifikation und Integration



- Zwei Arten von Schnittstellen
 - Spezifikationsschnittstelle im linken Ast des V-Modells
 - Integrationsschnittstelle im rechten Ast des V-Modells
- Grosse Komplexität
 - Hersteller
 - n Komponenten von n verschiedenen Zulieferern
 - 1:n-Beziehung an der Spezifikationsschnittstelle
 - 1:n-Beziehung an der Integrationsschnittstelle
 - Zulieferer
 - 1 Komponente an m verschiedene Hersteller
 - 1:m-Beziehung an der Spezifikationsschnittstelle
 - 1:m-Beziehung an der Integrationsschnittstelle

Herstellersicht

Baugruppenverantwortlicher Türe



■ Ansprechpartner

- Baugruppenverantwortlicher Karosserie
- Baugruppenverantwortlicher Sitze
- Baugruppenverantwortlicher Kombi-Instrument
- Baugruppenverantwortlicher Blinker
- Baugruppenverantwortlicher Mittelkonsole
- Baugruppenverantwortlicher Soundsystem
- Baugruppenverantwortlicher Seitenairbag
- Verantwortlicher Passive Sicherheit
- Verantwortlicher EMV
- Verantwortlicher Verkabelung
- Verantwortlicher Vernetzung
- Verantwortlicher Telematik

■ Zulieferer

- **Schliesssystem**
- **Scheiben**
- **Fensterheber**
- **Aussenspiegel**
- **Türsteuergerät**
- **Schalter**

■ Schnittstellen

- Mechanik
- Energie
- Information

Zulieferersicht (Prinzipdarstellung)



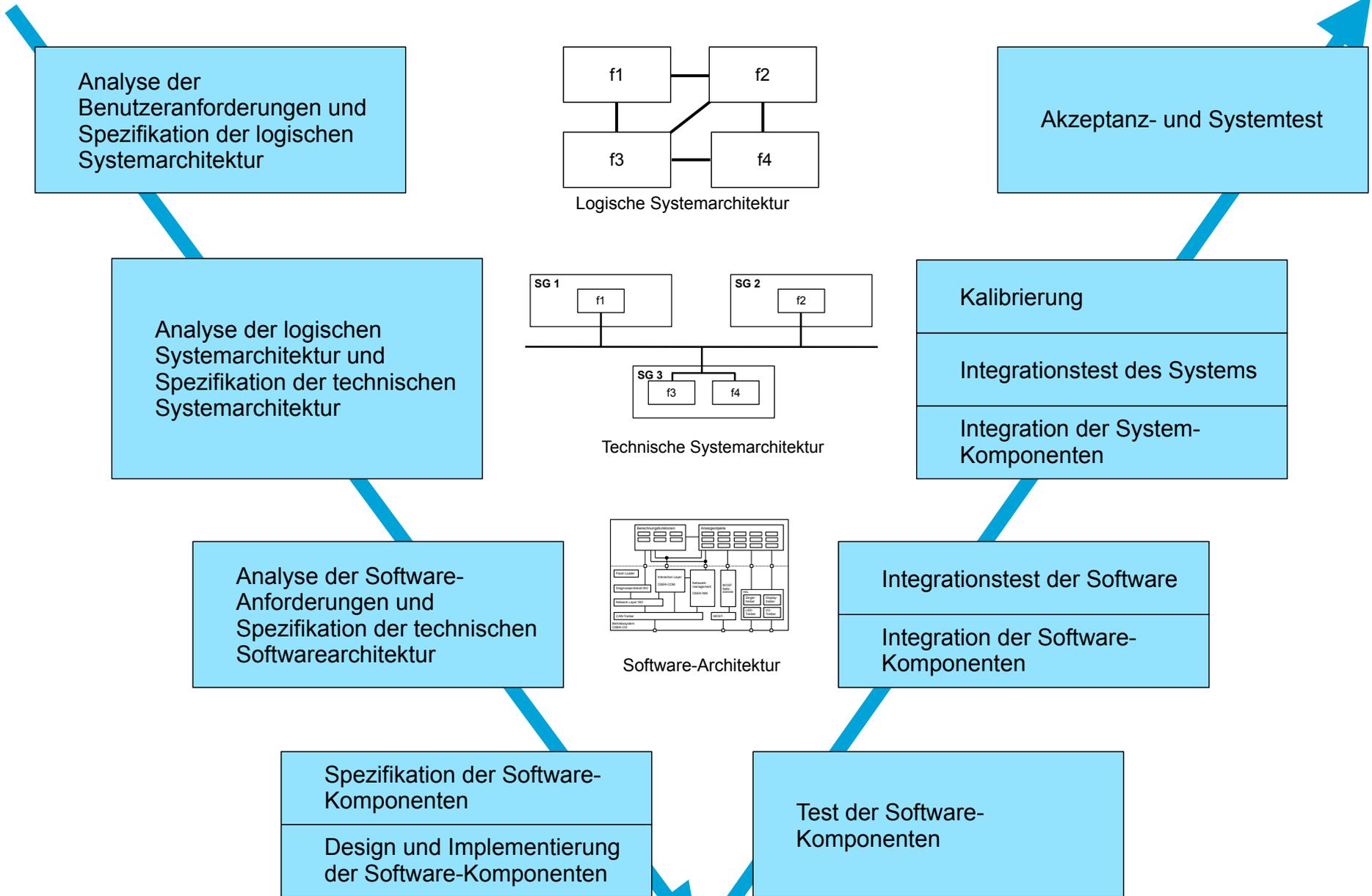
	BMW	Mer- cedes	Audi	Porsche	VW	Ford	Opel
Schliesssystem	X	X	X				
Scheiben			X				
Fensterheber			X	X	X		
Aussenspiegel	X	X	X	X	X	X	X
Türsteuergerät		X					
Schalter						X	X

Herstellersicht (Prinzipdarstellung)



	Bosch	Conti	Kostal	Dräxl- meier	Magna	Delphi	TRW
Schliesssystem		X	X				
Scheiben			X				
Fensterheber			X	X	X		
Aussenspiegel					X		X
Türsteuergerät	X	X					
Schalter						X	X

Schnittstellen für Spezifikation und Integration (Nach Schäuuffele, Zurawka)



6. SW-Entwicklung / 2. Unterstützungsprozesse

Unterstützungsprozesse für die Embedded Software Entwicklung



1. Vorgehensmodelle und Standards

2. Konfigurationsmanagement

3. Projektmanagement

4. Lieferantenmanagement

1. System- und Komponentenverantwortung

2. Schnittstellen für Spezifikation und Integration

3. Festlegung des firmenübergreifenden Entwicklungsprozesses

5. Anforderungsmanagement

6. Qualitätssicherung

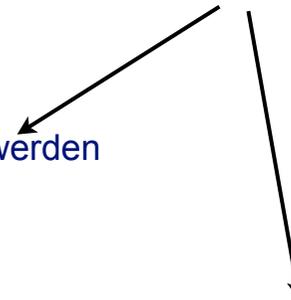
Festlegung des firmenübergreifenden Entwicklungsprozesses



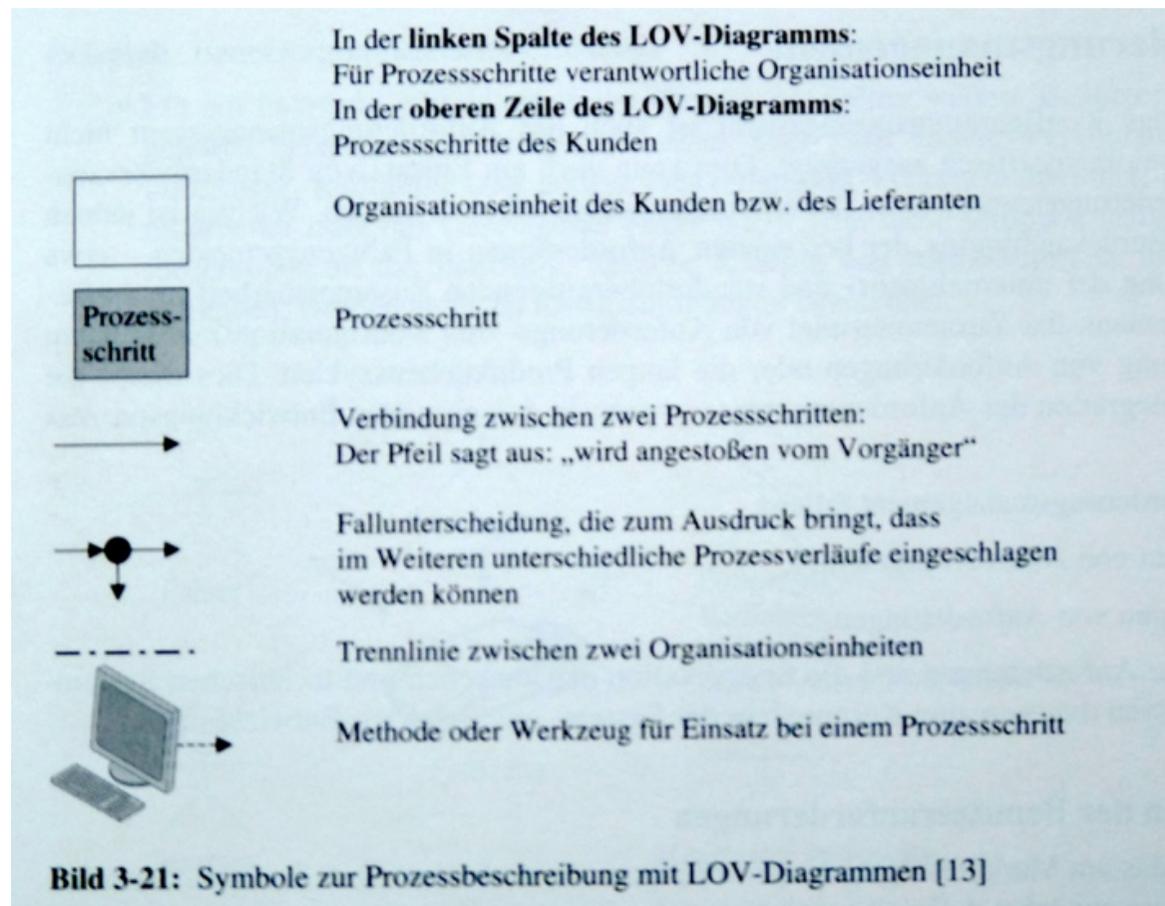
- Elektronische Steuergeräte sind eingebettete Systeme, die für den Benutzer nicht unmittelbar in Erscheinung treten
 - Beispiel: Türsteuergerät
- Für den Benutzer sind die Funktionen sichtbar
 - Beispiele:
 - Fensterheber
 - Spiegelverstellung
 - Sitzverstellung
- Grundfunktionen können herstellerübergreifend entwickelt werden
 - Beispiel:
 - Steuerung von Fensterheber, Spiegelverstellung, Sitzverstellung
- Wettbewerbsdifferenzierende Zusatzfunktionen werden herstellerepezifisch entwickelt
 - Beispiel:
 - Automatische Spiegelverstellung und Sitzverstellung (Memory, Ergonomie)

- Elektronische Steuergeräte sind eingebettete Systeme, die für den Benutzer nicht unmittelbar in Erscheinung treten
 - Beispiel: Türsteuergerät
- Für den Benutzer sind die Funktionen sichtbar
 - Beispiele:
 - Fensterheber
 - Spiegelverstellung
 - Sitzverstellung
- Grundfunktionen können herstellerübergreifend entwickelt werden
 - Beispiel:
 - Steuerung von Fensterheber, Spiegelverstellung, Sitzverstellung
- Wettbewerbsdifferenzierende Zusatzfunktionen werden herstellerspezifisch entwickelt
 - Beispiel:
 - Automatische Spiegelverstellung und Sitzverstellung (Memory, Ergonomie)

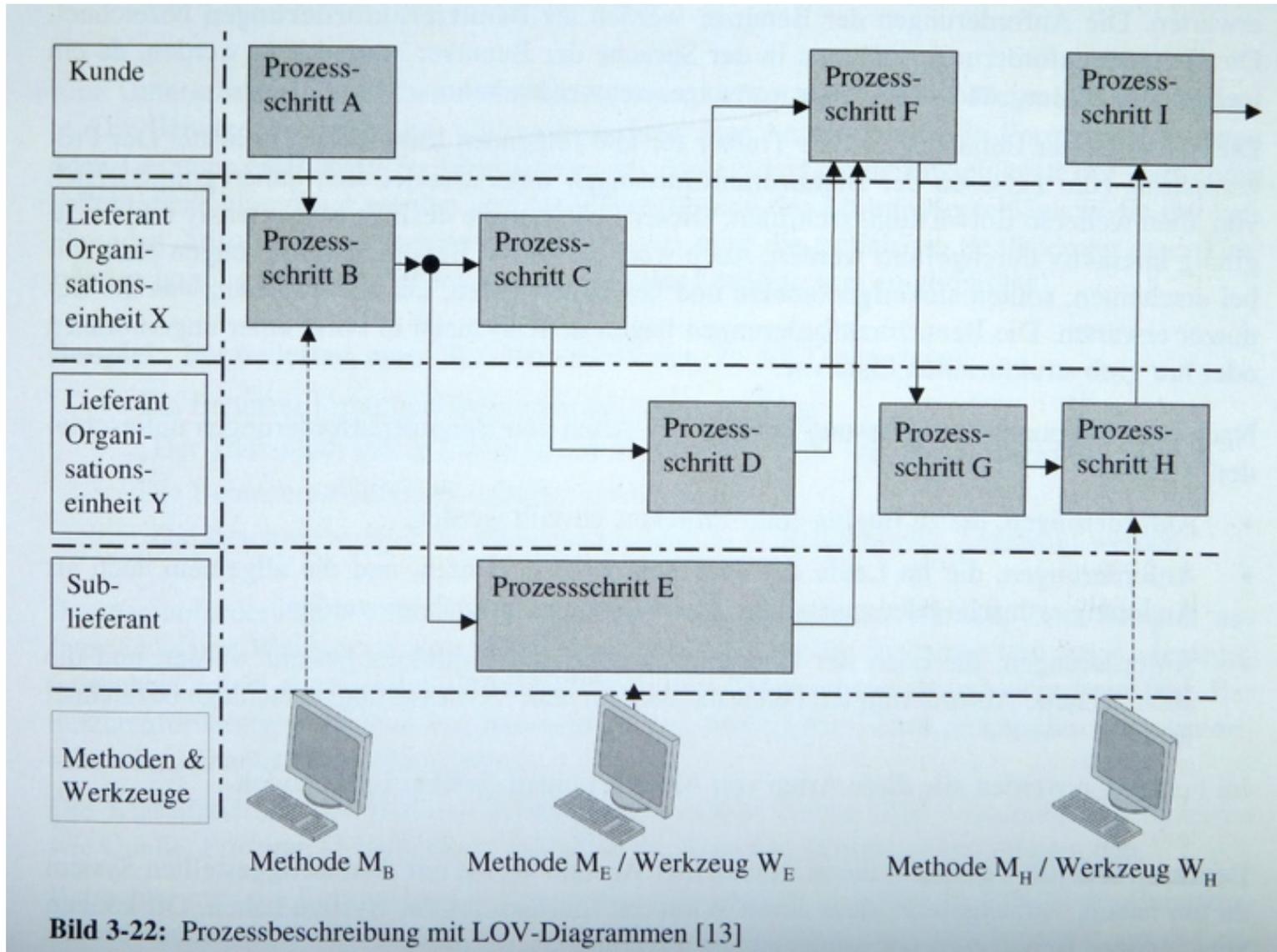
Grundidee von AUTOSAR



- LOV: Line of Visibility
- Grafische Beschreibung der komplexen Beziehungen zwischen Herstellern und Zulieferern



LOV-Diagramme: Beispiel



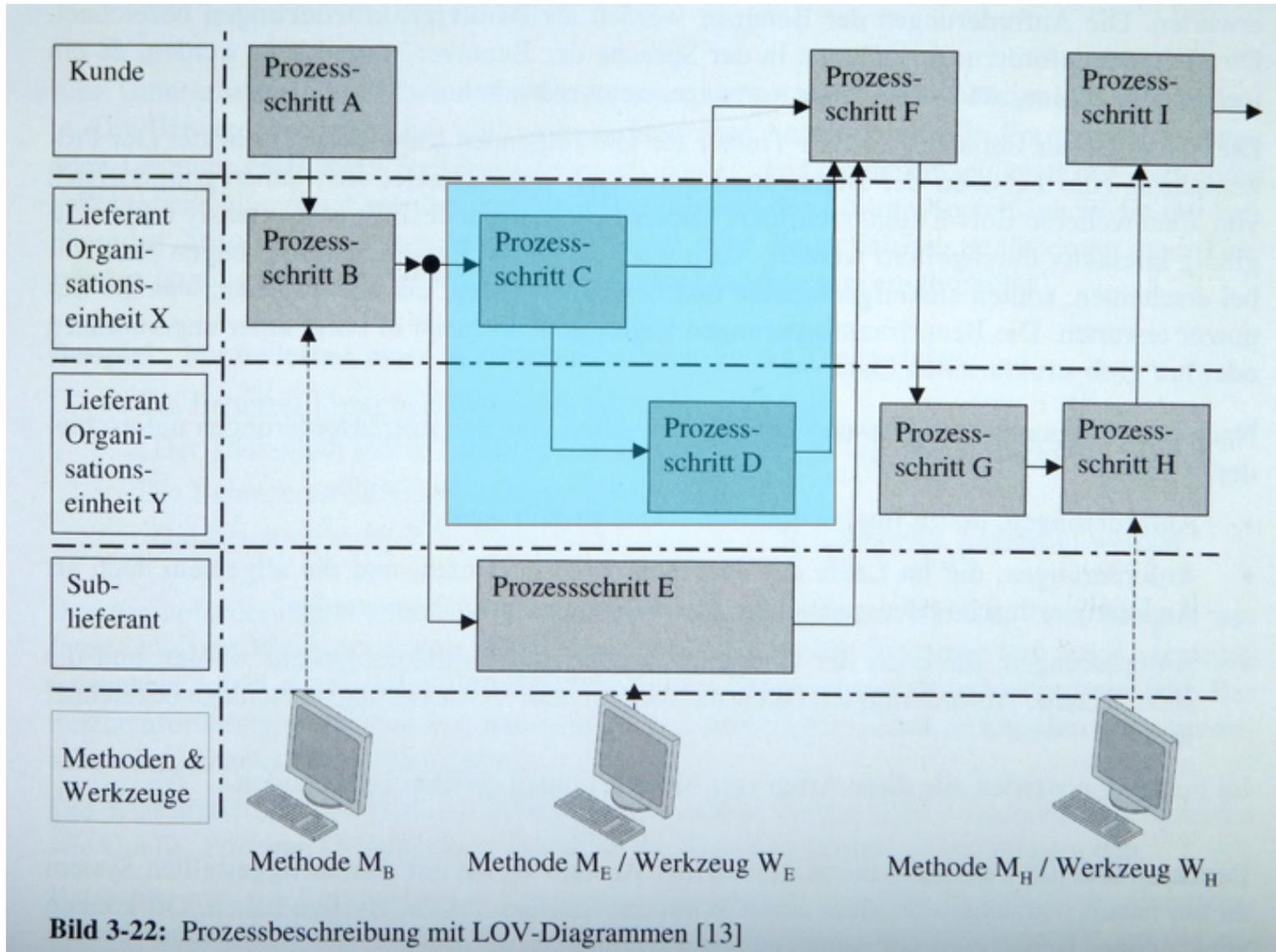
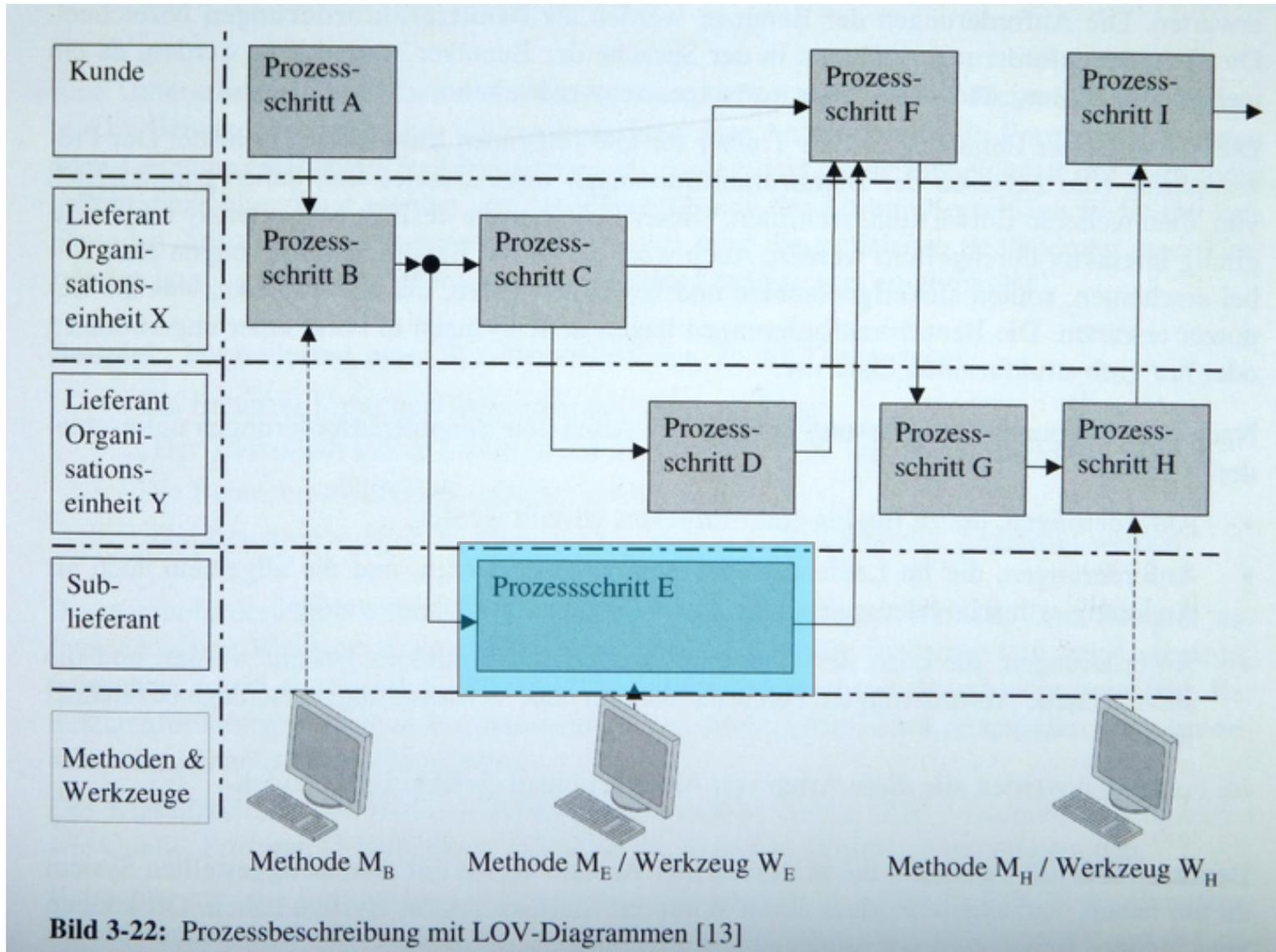


Bild 3-22: Prozessbeschreibung mit LOV-Diagrammen [13]

LOV-Diagramme: Alternative 2



6. SW-Entwicklung / 2. Unterstützungsprozesse

Unterstützungsprozesse für die Embedded Software Entwicklung



1. Vorgehensmodelle und Standards
2. Konfigurationsmanagement
3. Projektmanagement
4. Lieferantenmanagement
- 5. Anforderungsmanagement**
 1. Erfassen von Benutzeranforderungen
 2. Verfolgen von Anforderungen
6. Qualitätssicherung

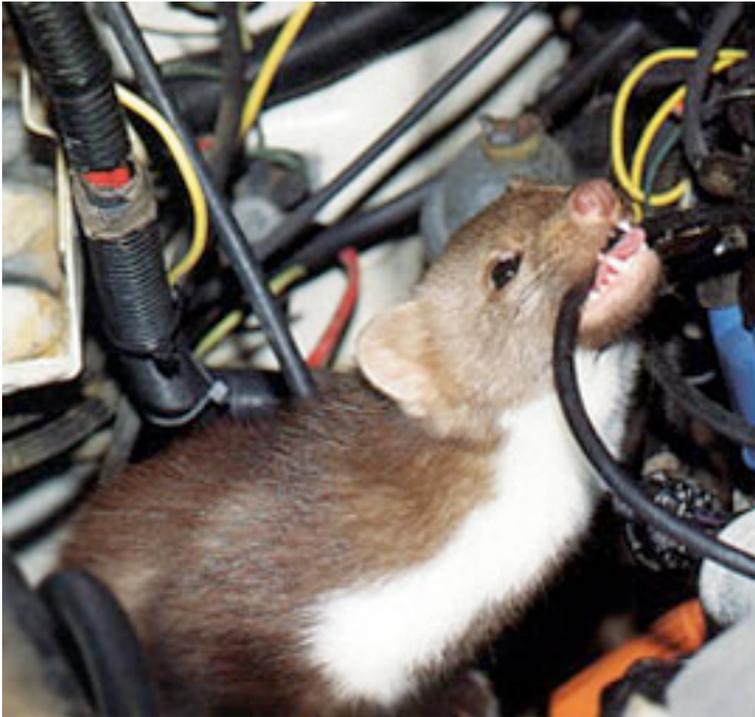
- Anforderungsmanagement ist - wie Konfigurationsmanagement - nicht grundsätzlich automobilspezifisch
 - Einsatz von Standardwerkzeugen wie Telelogic DOORS (Jetzt IBM)
 - <http://www-01.ibm.com/software/awdtools/doors/>
- Aber: Berücksichtigung von besonderen Anforderungen in Fahrzeugprojekten
 - Unterstützung der unternehmens- und standortübergreifenden Zusammenarbeit im Anforderungsmanagement
 - Versionierung von Anforderungen
 - Zusammenspiel von Anforderungsmanagement und Konfigurationsmanagement
 - Lange Produktlebenszyklen
 - Änderung der Anforderungen
- Unterstützungsprozess Anforderungsmanagement
 - Erfassen von Benutzeranforderungen
 - Verfolgen von Anforderungen
- Kernprozess der System- und Softwareentwicklung
 - Analyse der Anforderungen
 - Spezifikation der logischen und technischen Systemarchitektur

- Benutzer sind alle Personen, deren Wünsche Einfluss auf das Fahrzeug haben können
- Benutzergruppen eines Fahrzeugs
 - Fahrer
 - Insassen
 - Andere Verkehrsteilnehmer
 - Fahrzeuge
 - Fahrer
 - Radfahrer
 - Reiter
 - Fussgänger
 - ...
 - ...
 - ...
 - Mitarbeiter im Service
 - Gesetzgeber
 - Gesetzliche Vorgaben werden auch als Randbedingungen bezeichnet

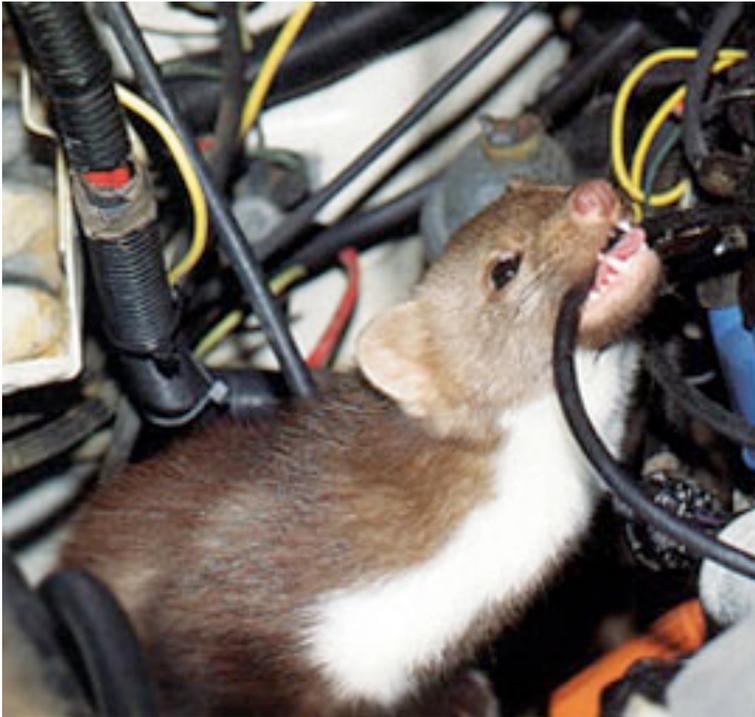
Weitere Benutzergruppen („Stakeholder“)



Weitere Benutzergruppen („Stakeholder“)



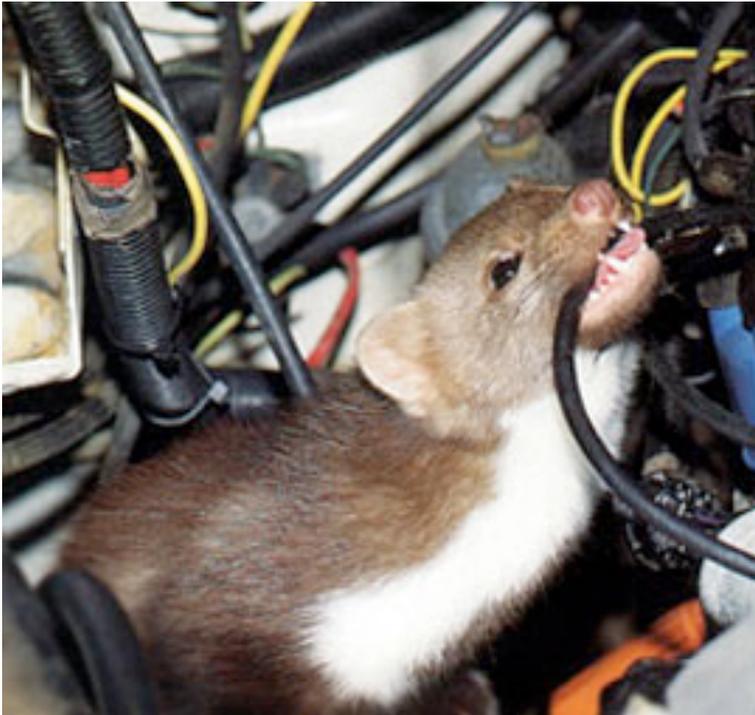
Weitere Benutzergruppen („Stakeholder“)



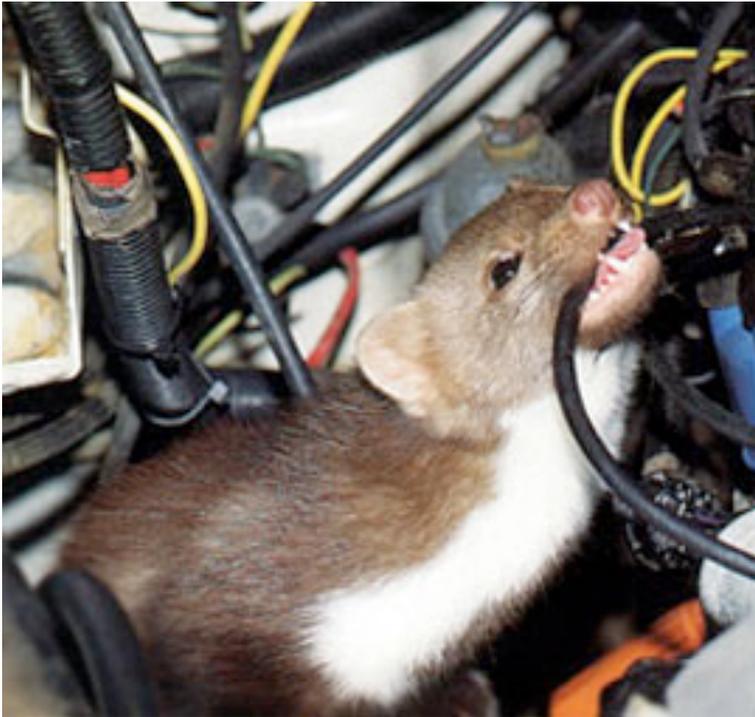
Weitere Benutzergruppen („Stakeholder“)



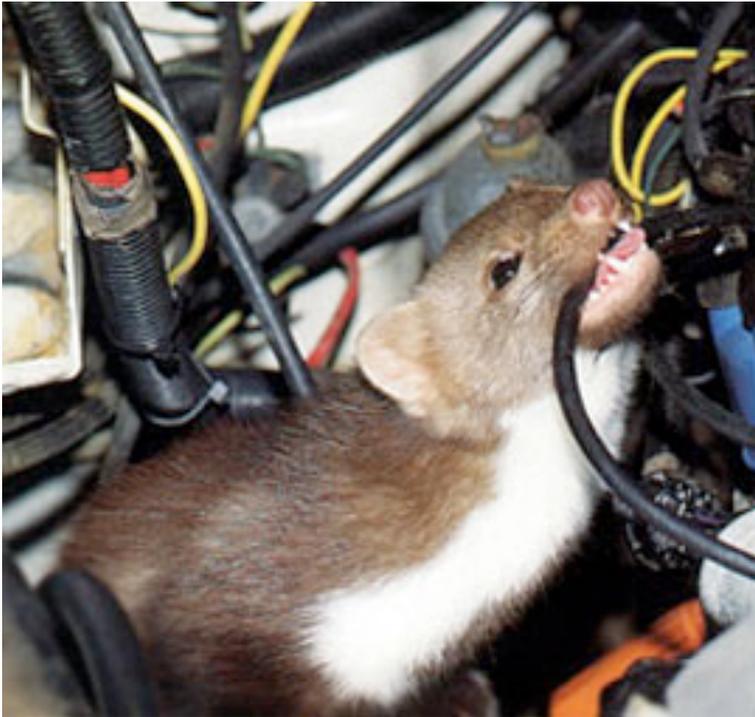
Weitere Benutzergruppen („Stakeholder“)



Weitere Benutzergruppen („Stakeholder“)



Weitere Benutzergruppen („Stakeholder“)



- Benutzer sind alle Personen, deren Wünsche Einfluss auf das Fahrzeug haben können
- Benutzergruppen eines Fahrzeugs
 - Fahrer
 - Insassen
 - Andere Verkehrsteilnehmer
 - Fahrzeuge
 - Fahrer
 - Radfahrer
 - Reiter
 - Fussgänger
 - Diebe, Vandalen
 - Marder
 - Insekten, Wild(unfälle)
 - Mitfahrende Haustiere
 - ...
 - Mitarbeiter im Service
 - Gesetzgeber
 - Gesetzliche Vorgaben werden auch als Randbedingungen bezeichnet

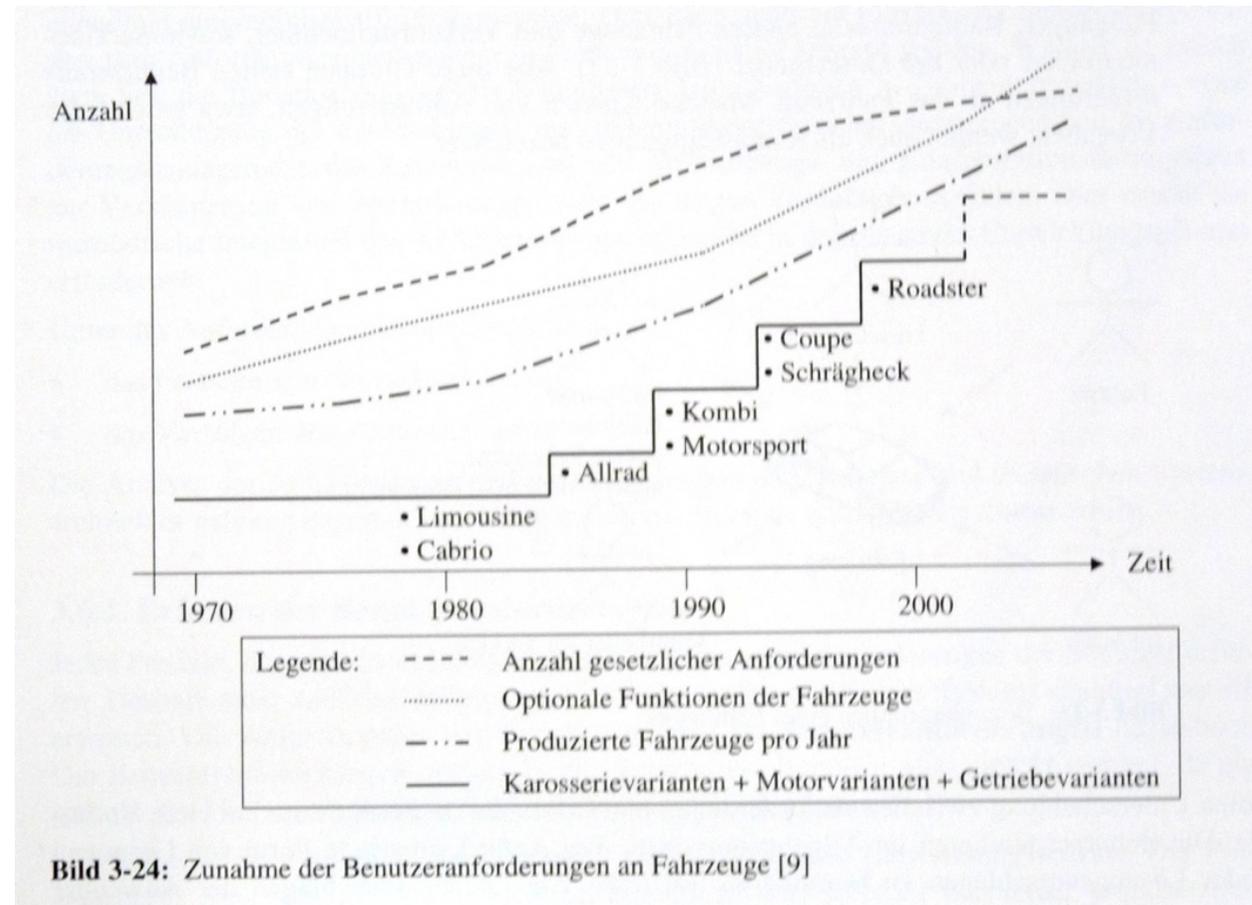
Drei Arten von Benutzeranforderungen

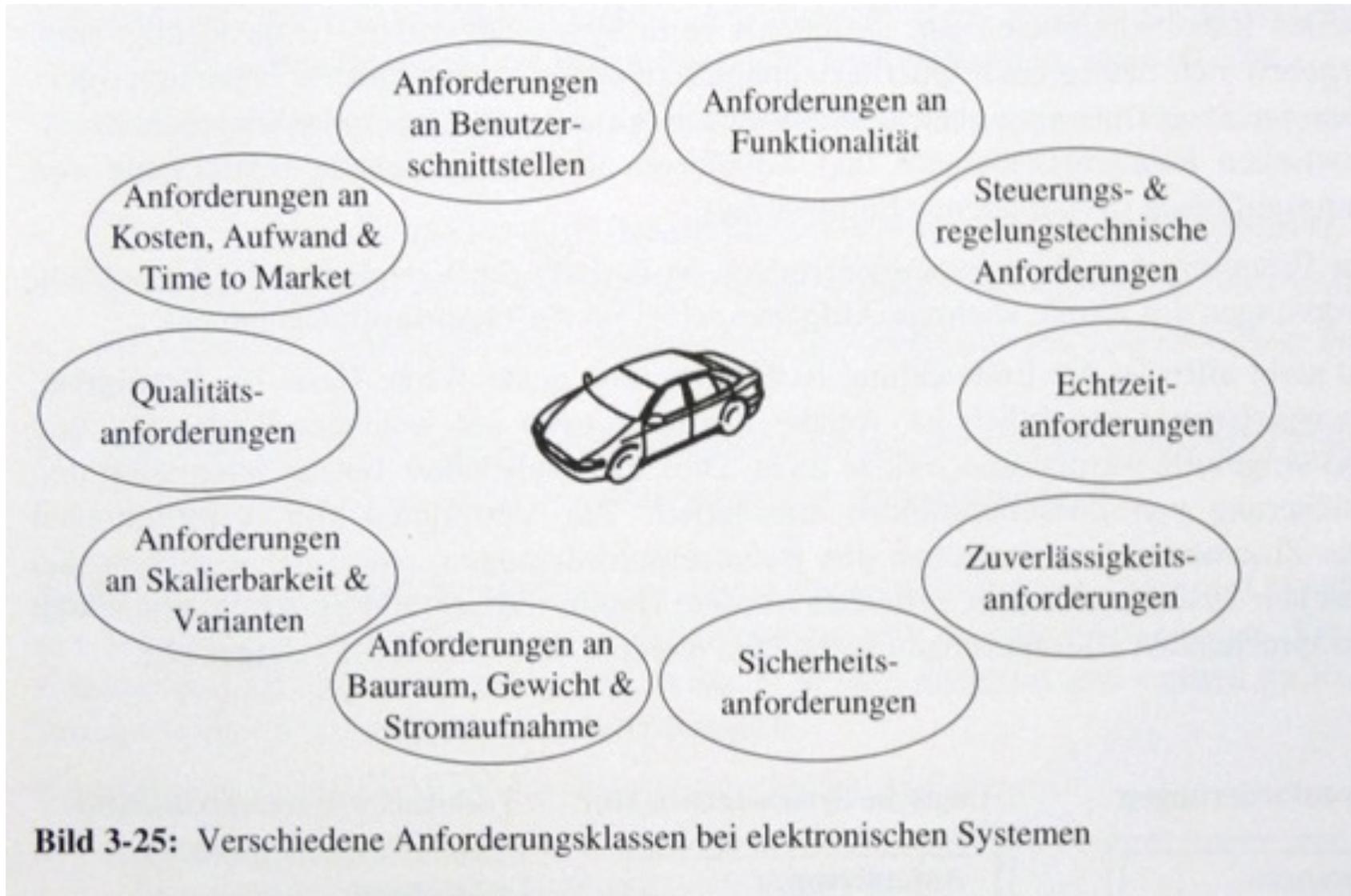
- Unterscheidung nach Zeitpunkt der Erfassung
 - Anforderungen, die zu Beginn des Projektes gestellt werden
 - Anforderungen, die im Laufe des Projektes gestellt werden
 - Änderungswünsche
 - Zusätzliche Anforderungen
 - Anforderungen, die nach Übergabe des Produktes gestellt werden
 - Neue Anforderungen
 - Fehlermeldungen
 - Verbesserungsvorschläge
- Erfassung von Benutzeranforderungen durch
 - Ableitung aus existierenden Systemen
 - Technische Entwicklungen
 - Rückmeldungen
 - Interviews
 - Workshops
 - ...

Anzahl der Benutzeranforderungen

■ Zunahme mit jeder Fahrzeuggeneration

- Gestiegene Anzahl der Fahrzeugfunktionen
- Zunahme der Fahrzeugvarianten
- Gestiegene Kundenerwartungen
- Gestiegene Gesetzliche Anforderungen





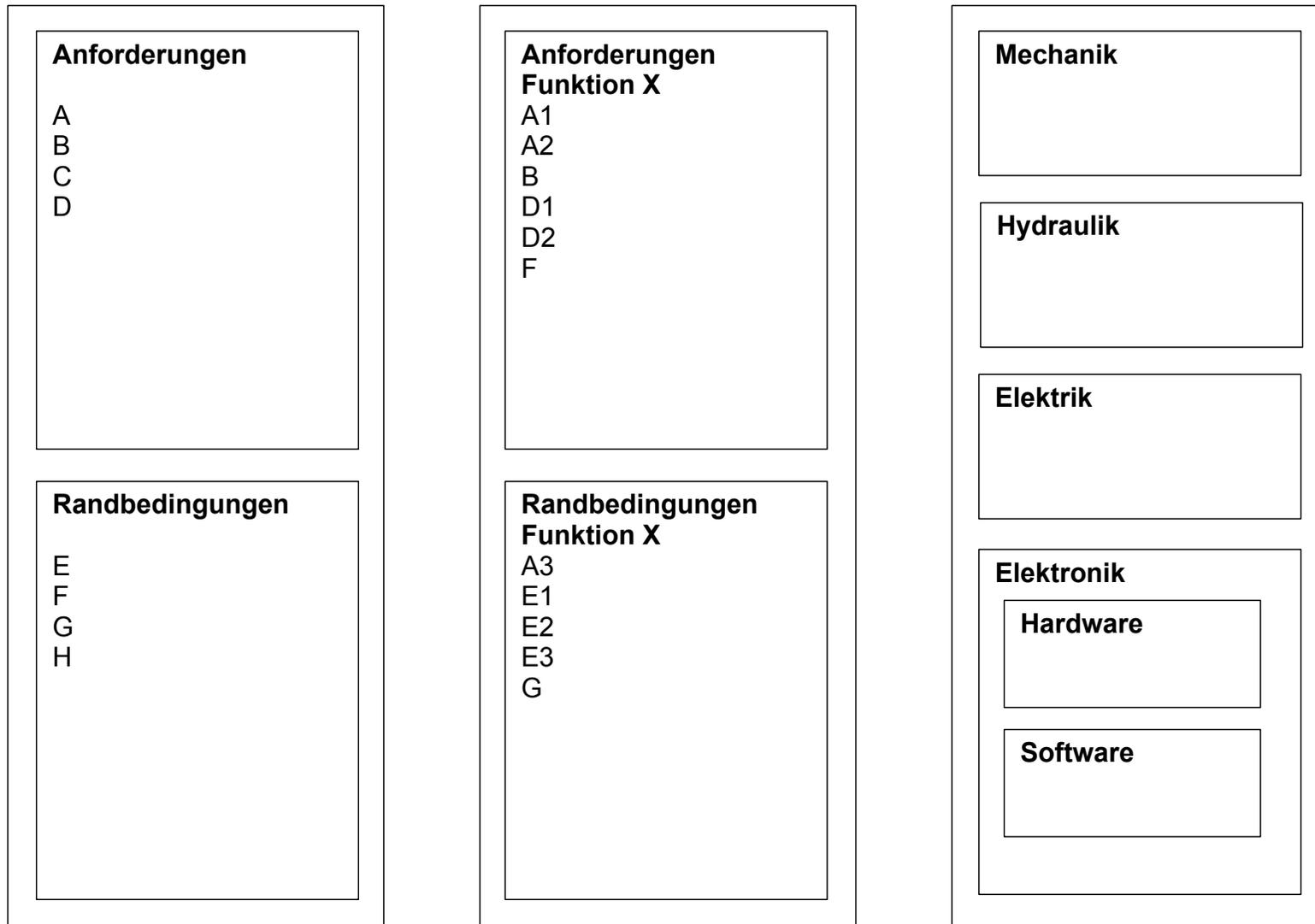
6. SW-Entwicklung / 2. Unterstützungsprozesse

Unterstützungsprozesse für die Embedded Software Entwicklung

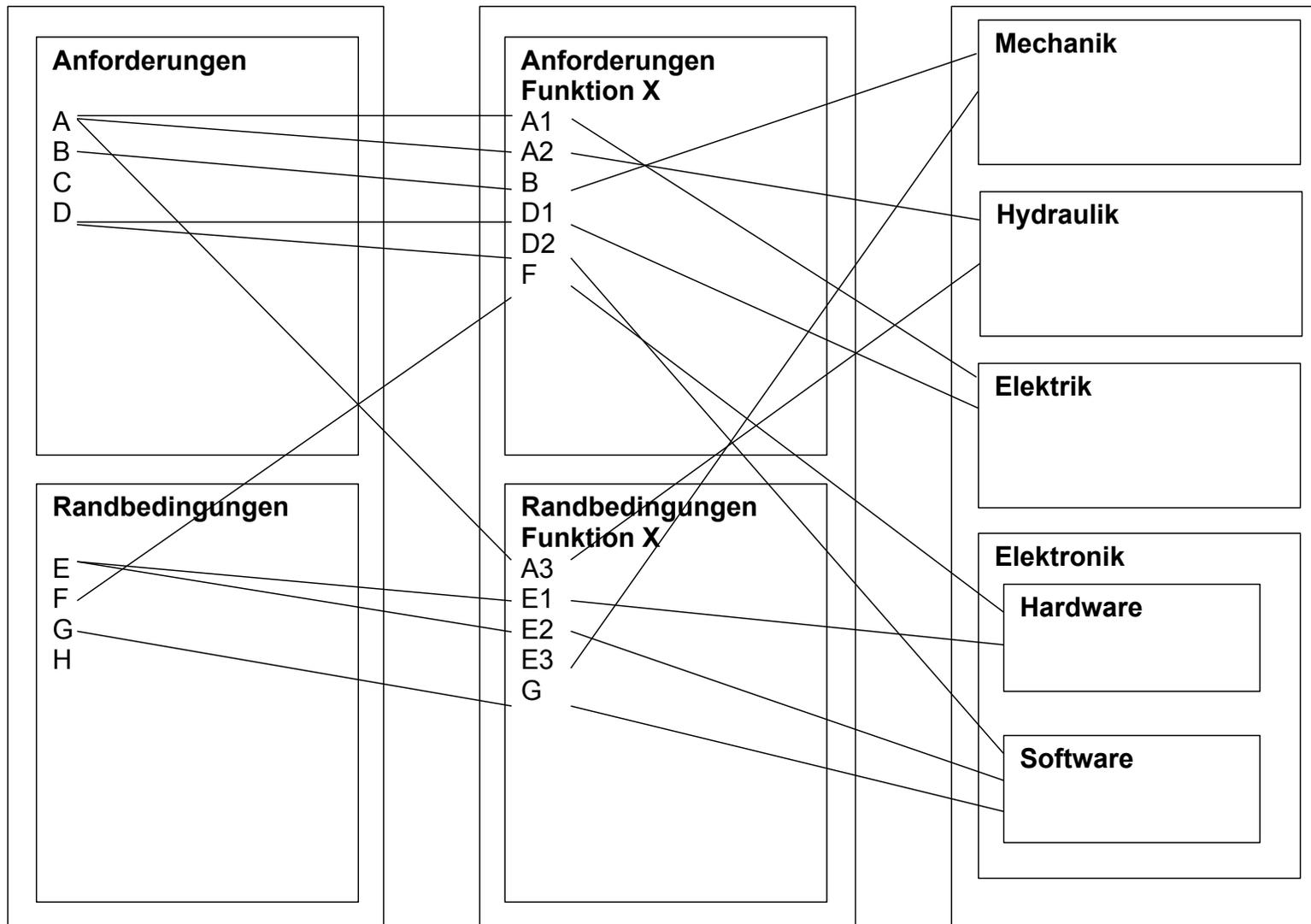


1. Vorgehensmodelle und Standards
2. Konfigurationsmanagement
3. Projektmanagement
4. Lieferantenmanagement
- 5. Anforderungsmanagement**
 1. Erfassen von Benutzeranforderungen
 - 2. Verfolgen von Anforderungen**
6. Qualitätssicherung

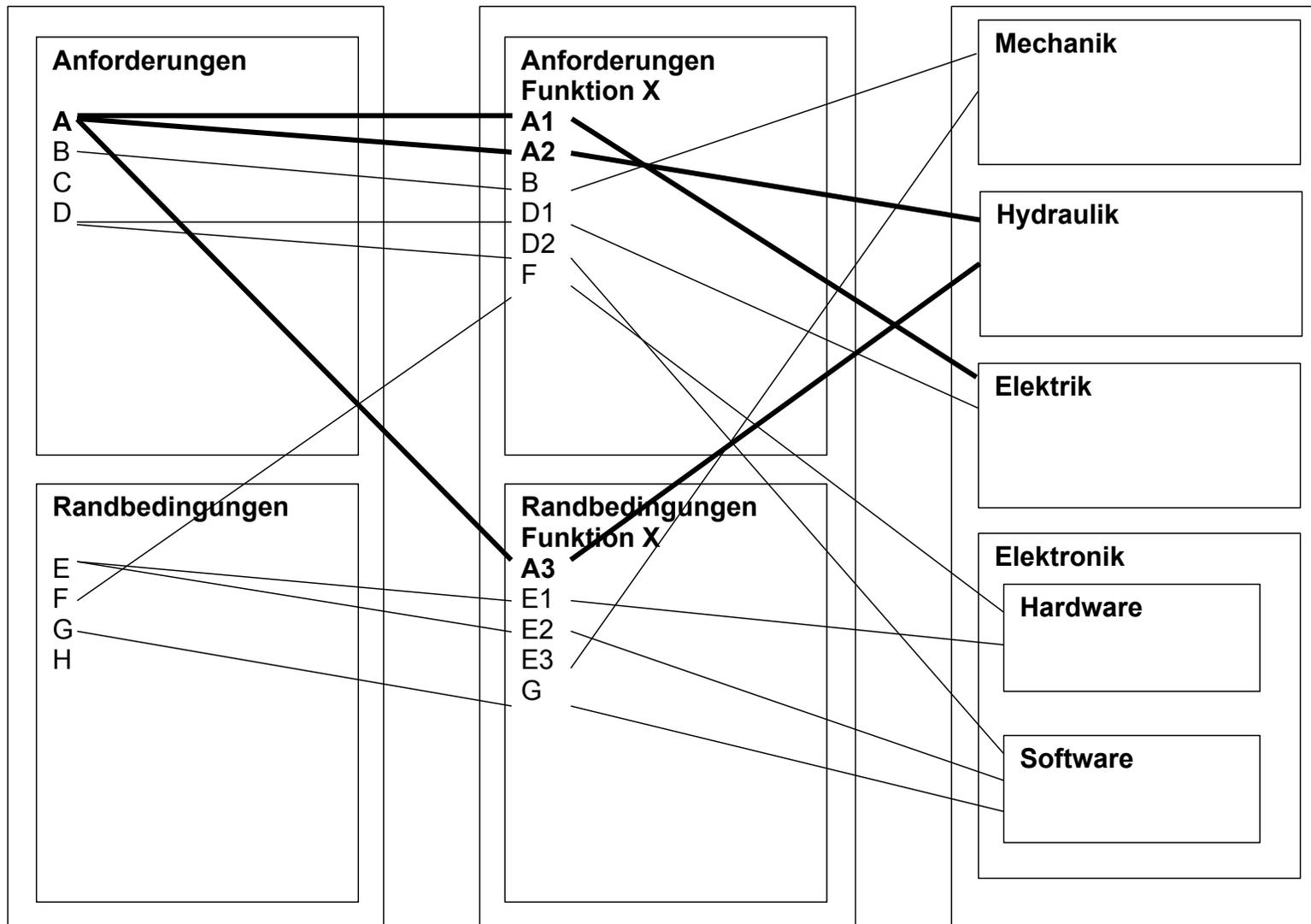
Benutzeranforderungen, logische und technische Systemarchitektur (Nach Schäuffele, Zurawka)



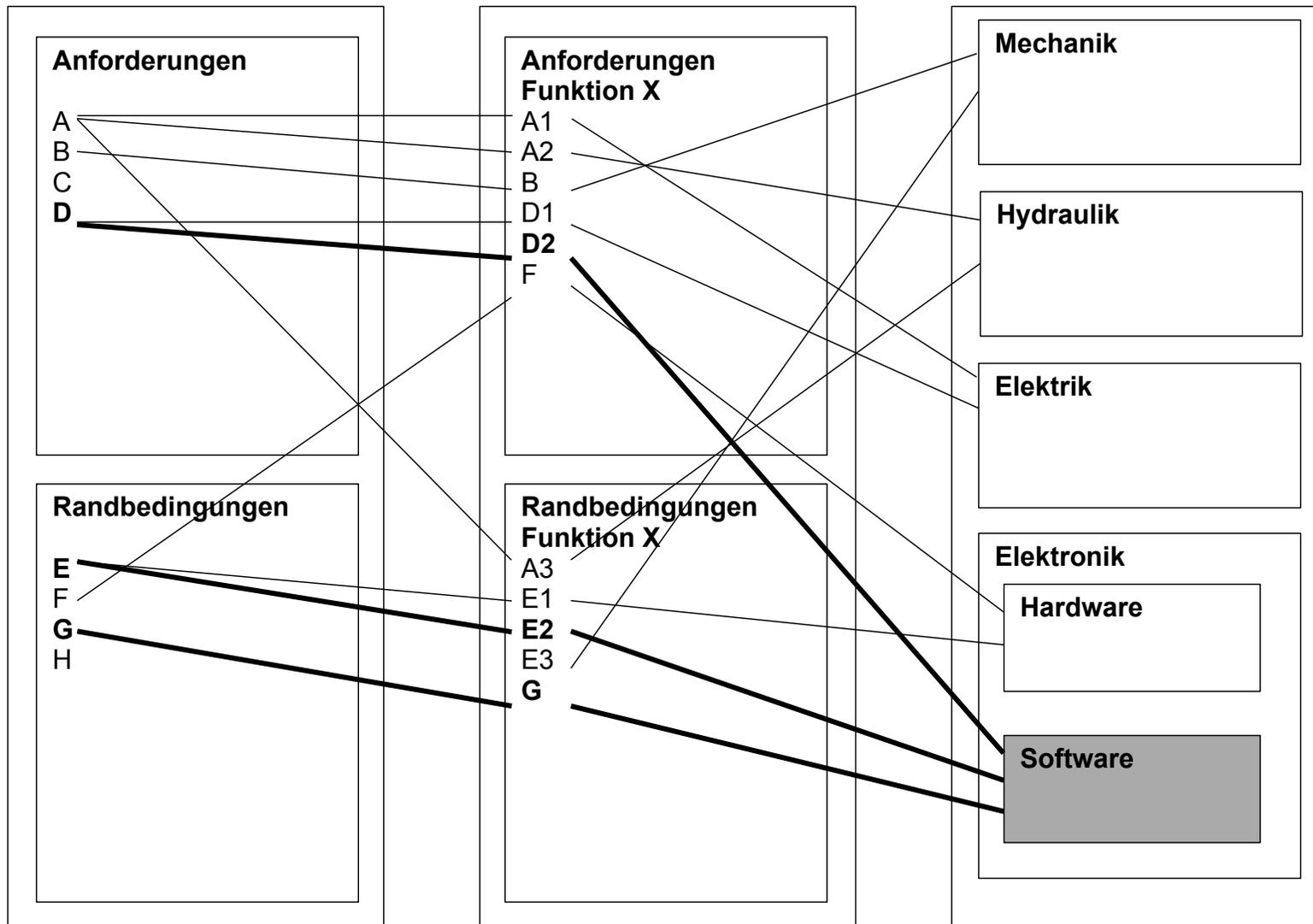
Verfolgen von Anforderungen: Welche Anforderungen werden wie umgesetzt?



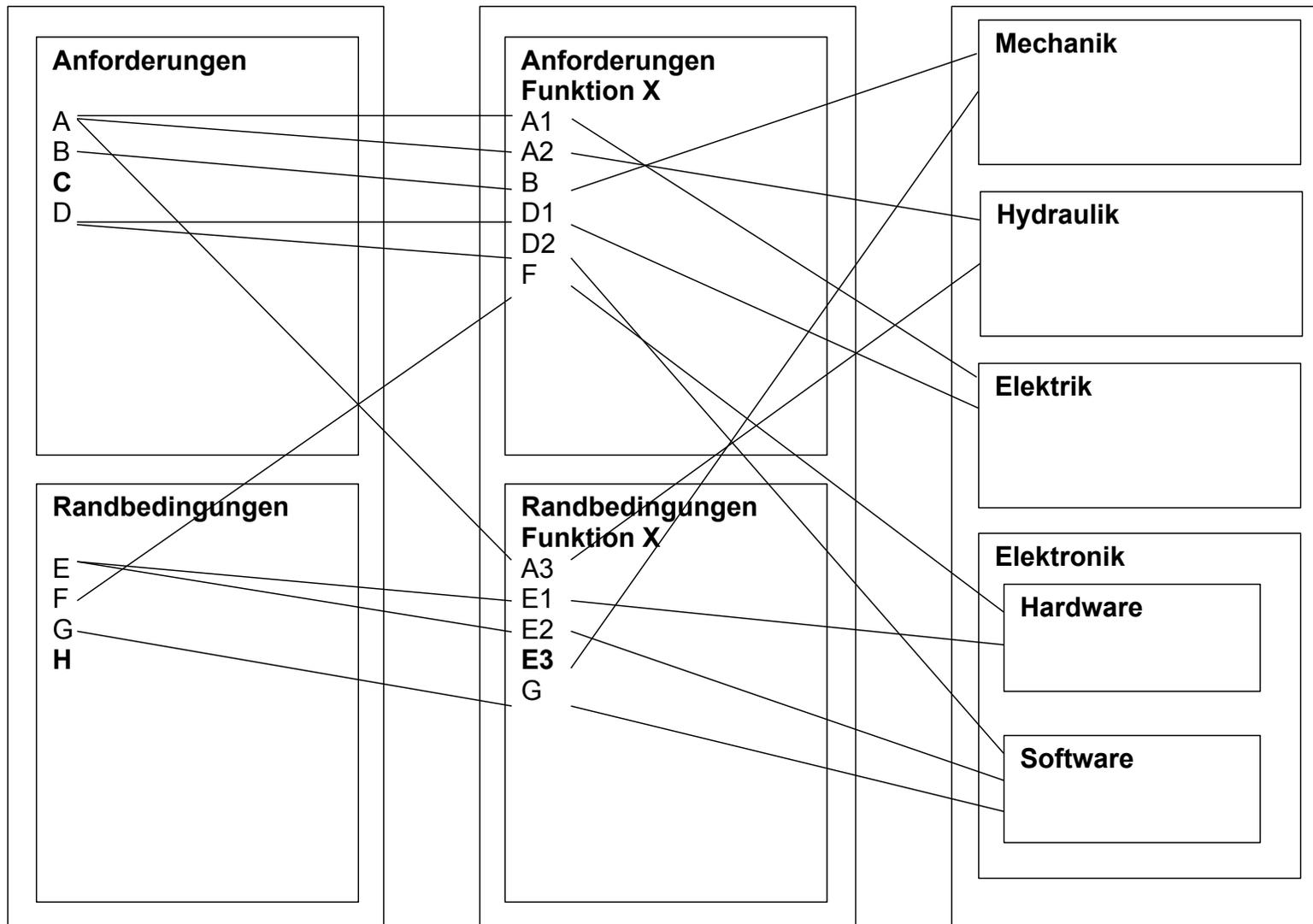
Verfolgen von Anforderungen: Welche Anforderungen werden wie umgesetzt?



Verfolgen von Anforderungen: Welche Anforderungen werden wie umgesetzt?



Verfolgen von Anforderungen: Welche Anforderungen werden wie umgesetzt?



6. SW-Entwicklung / 2. Unterstützungsprozesse

Unterstützungsprozesse für die Embedded Software Entwicklung



1. Vorgehensmodelle und Standards
2. Konfigurationsmanagement
3. Projektmanagement
4. Lieferantenmanagement
5. Anforderungsmanagement

6. Qualitätssicherung

1. Integrations- und Testschritte

2. Maßnahmen zur Qualitätssicherung von Software

■ Qualitätssicherung

- Alle Maßnahmen, die sicherstellen, dass das Produkt die geforderten Anforderungen erfüllt.

■ Richtlinien zur Qualitätssicherung: Vorbeugende Maßnahmen

- Einsatz von erfahrenen und geschulten Mitarbeitern
- Geeigneter Entwicklungsprozess mit definierten Testschritten
- Richtlinien, Maßnahmen und Standards zur Unterstützung des Prozesses
- Werkzeugumgebung zur Unterstützung des Prozesses
- Automatisierung manueller und fehlerträchtiger Arbeitsschritte

■ Maßnahmen zur Qualitätssicherung: Finden von Fehlern

- Nach jedem Schritt des Entwicklungsprozesses
- Ins V-Modell integriert, siehe „Kernprozess“

■ Software-Produkte

- Spezifikationsfehler
- Implementierungsfehler
- Ergebnis von zahlreichen Untersuchungen:
 - In den meisten Projekten überwiegen die Spezifikationsfehler

- Validation: Prüfen auf Spezifikationsfehler
- Validation ist der Prozess zur Beurteilung eines Systems oder einer Komponente mit dem Ziel festzustellen, ob der Einsatzzweck oder die Benutzererwartungen erfüllt werden. Funktionsvalidation ist demnach die Prüfung, ob die Spezifikation die Benutzeranforderungen erfüllt, ob überhaupt die Benutzerakzeptanz durch eine Funktion erreicht wird.
- Verifikation: Prüfen auf Implementierungsfehler
- Verifikation ist der Prozess zur Beurteilung eines Systems oder einer Komponente mit dem Ziel festzustellen, ob die Resultate einer gegebenen Entwicklungsphase den Vorgaben für diese Phase entsprechen. Software-Verifikation ist demnach die Prüfung, ob eine Implementierung der für den betreffenden Entwicklungsschritt vorgegebenen Spezifikation genügt.

■ Verifikation:

- „Does the system things right?“ / „Erfüllt das System seine Aufgabe richtig?“
- Prüfung z.B. gegen Technische Anforderungen
- Beispiel: Nalyse des Zeitverhaltens durch Untersuchung der Worst Case Execution Time (WCET)

■ Validation:

- „Does the system the right things?“ / „Erfüllt das System die richtige Aufgabe?“
- Prüfung z.B. gegen Benutzeranforderungen
- Beispiel: Simulation der Benutzeroberfläche

■ Ähnlich:

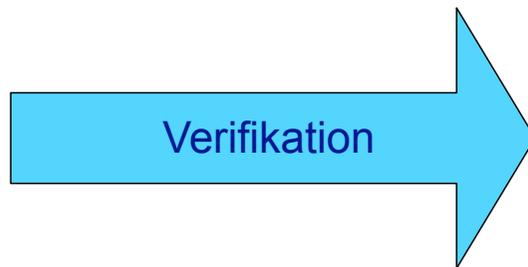
- Effektiv: Die richtigen Dinge tun.
- Effizient: Die Dinge richtig tun.

- Das V-Modell unterscheidet vier verschiedene Testschritte:
 - Beim Komponententest wird eine Komponente gegen die Spezifikation der Komponente getestet.
 - Beim Integrationstest wird das System gegen die Spezifikation der technischen Systemarchitektur getestet.
 - Beim Systemtest wird das System gegen die Spezifikation der logischen Systemarchitektur getestet.
 - Beim Akzeptanztest wird das System gegen die Benutzeranforderungen getestet.
- Validation und Verifikation



- Das V-Modell unterscheidet vier verschiedene Testschritte:
 - Beim Komponententest wird eine Komponente gegen die Spezifikation der Komponente getestet.
 - Beim Integrationstest wird das System gegen die Spezifikation der technischen Systemarchitektur getestet.
 - Beim Systemtest wird das System gegen die Spezifikation der logischen Systemarchitektur getestet.
 - Beim Akzeptanztest wird das System gegen die Benutzeranforderungen getestet.
- Validation und Verifikation

- Das V-Modell unterscheidet vier verschiedene Testschritte:
 - Beim Komponententest wird eine Komponente gegen die Spezifikation der Komponente getestet.
 - Beim Integrationstest wird das System gegen die Spezifikation der technischen Systemarchitektur getestet.
 - Beim Systemtest wird das System gegen die Spezifikation der logischen Systemarchitektur getestet.
 - Beim Akzeptanztest wird das System gegen die Benutzeranforderungen getestet.
- Validation und Verifikation



- Das V-Modell unterscheidet vier verschiedene Testschritte:
 - Beim Komponententest wird eine Komponente gegen die Spezifikation der Komponente getestet.
 - Beim Integrationstest wird das System gegen die Spezifikation der technischen Systemarchitektur getestet.
 - Beim Systemtest wird das System gegen die Spezifikation der logischen Systemarchitektur getestet.
 - Beim Akzeptanztest wird das System gegen die Benutzeranforderungen getestet.
- Validation und Verifikation

- Das V-Modell unterscheidet vier verschiedene Testschritte:
 - Beim Komponententest wird eine Komponente gegen die Spezifikation der Komponente getestet.
 - Beim Integrationstest wird das System gegen die Spezifikation der technischen Systemarchitektur getestet.
 - Beim Systemtest wird das System gegen die Spezifikation der logischen Systemarchitektur getestet.
 - Beim Akzeptanztest wird das System gegen die Benutzeranforderungen getestet.
- Validation und Verifikation



- Das V-Modell unterscheidet vier verschiedene Testschritte:
 - Beim Komponententest wird eine Komponente gegen die Spezifikation der Komponente getestet.
 - Beim Integrationstest wird das System gegen die Spezifikation der technischen Systemarchitektur getestet.
 - Beim Systemtest wird das System gegen die Spezifikation der logischen Systemarchitektur getestet.
 - Beim Akzeptanztest wird das System gegen die Benutzeranforderungen getestet.
- Validation und Verifikation

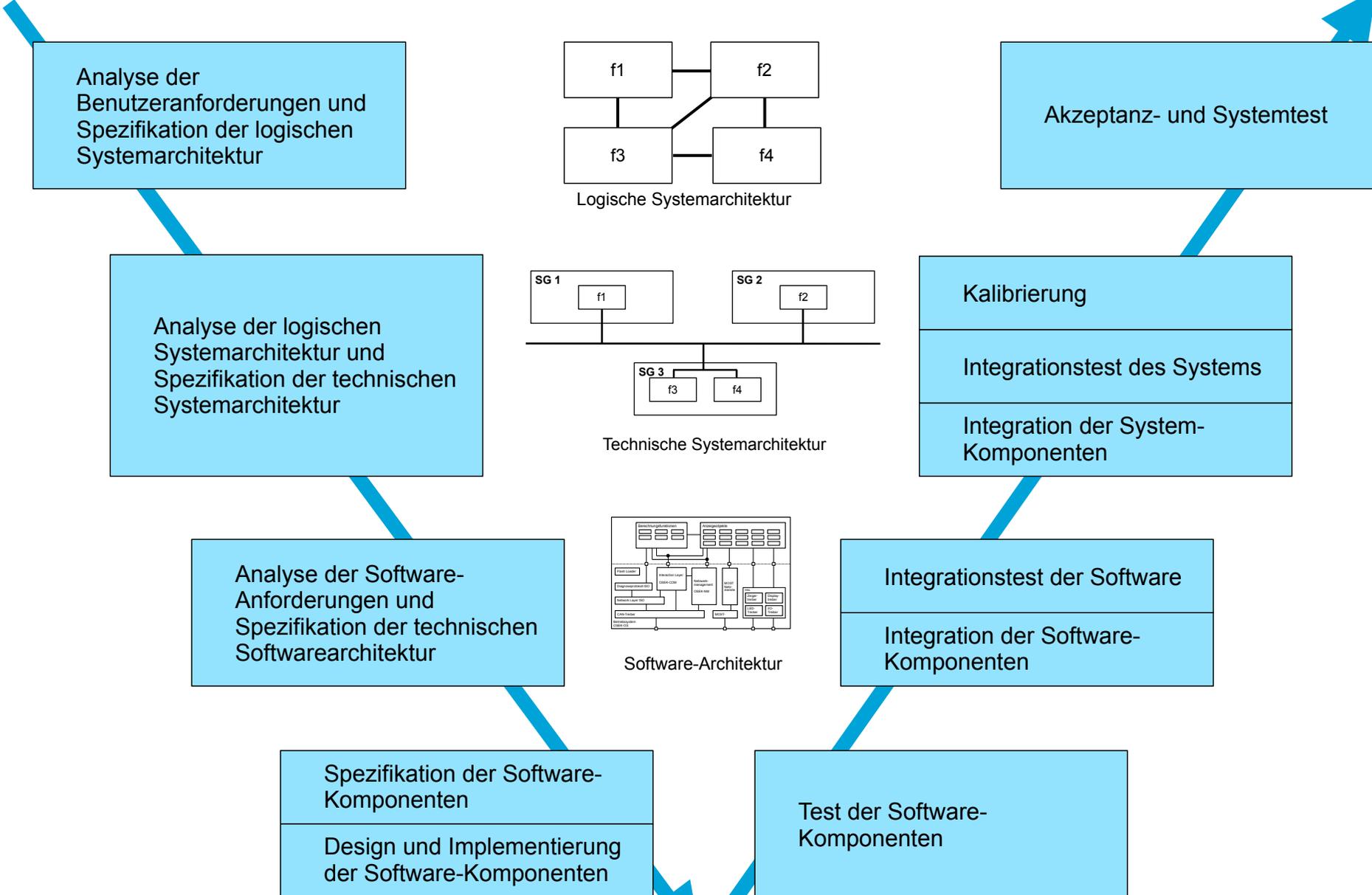
- Das V-Modell unterscheidet vier verschiedene Testschritte:
 - Beim Komponententest wird eine Komponente gegen die Spezifikation der Komponente getestet.
 - Beim Integrationstest wird das System gegen die Spezifikation der technischen Systemarchitektur getestet.
 - Beim Systemtest wird das System gegen die Spezifikation der logischen Systemarchitektur getestet.
 - Beim Akzeptanztest wird das System gegen die Benutzeranforderungen getestet.
- Validation und Verifikation



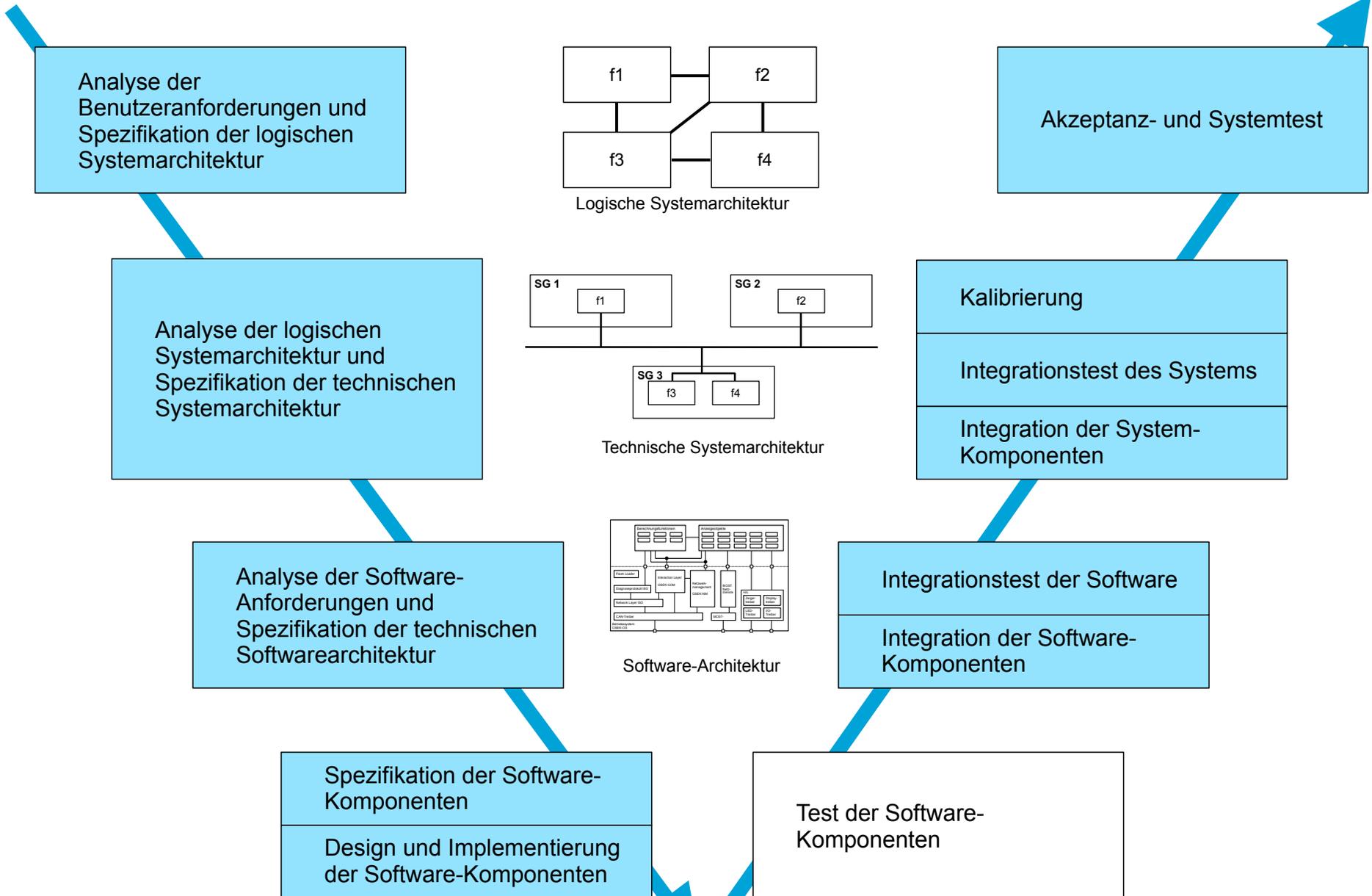
- Das V-Modell unterscheidet vier verschiedene Testschritte:
 - Beim Komponententest wird eine Komponente gegen die Spezifikation der Komponente getestet.
 - Beim Integrationstest wird das System gegen die Spezifikation der technischen Systemarchitektur getestet.
 - Beim Systemtest wird das System gegen die Spezifikation der logischen Systemarchitektur getestet.
 - Beim Akzeptanztest wird das System gegen die Benutzeranforderungen getestet.
- Validation und Verifikation

- Das V-Modell unterscheidet vier verschiedene Testschritte:
 - Beim Komponententest wird eine Komponente gegen die Spezifikation der Komponente getestet.
 - Beim Integrationstest wird das System gegen die Spezifikation der technischen Systemarchitektur getestet.
 - Beim Systemtest wird das System gegen die Spezifikation der logischen Systemarchitektur getestet.
 - Beim Akzeptanztest wird das System gegen die Benutzeranforderungen getestet.
- Validation und Verifikation

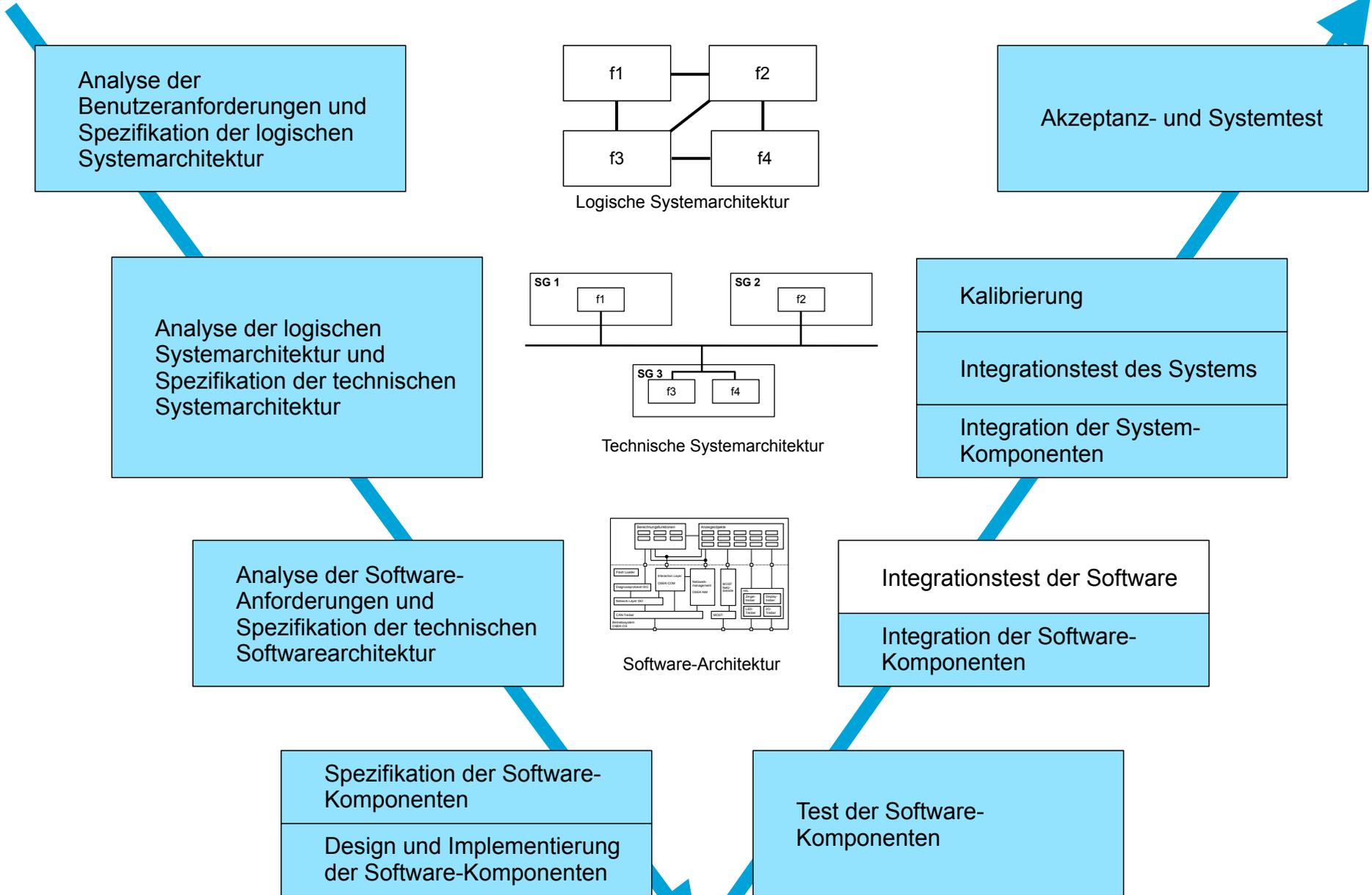
Integrations- und Testschritte (Nach Schäuuffele, Zurawka)



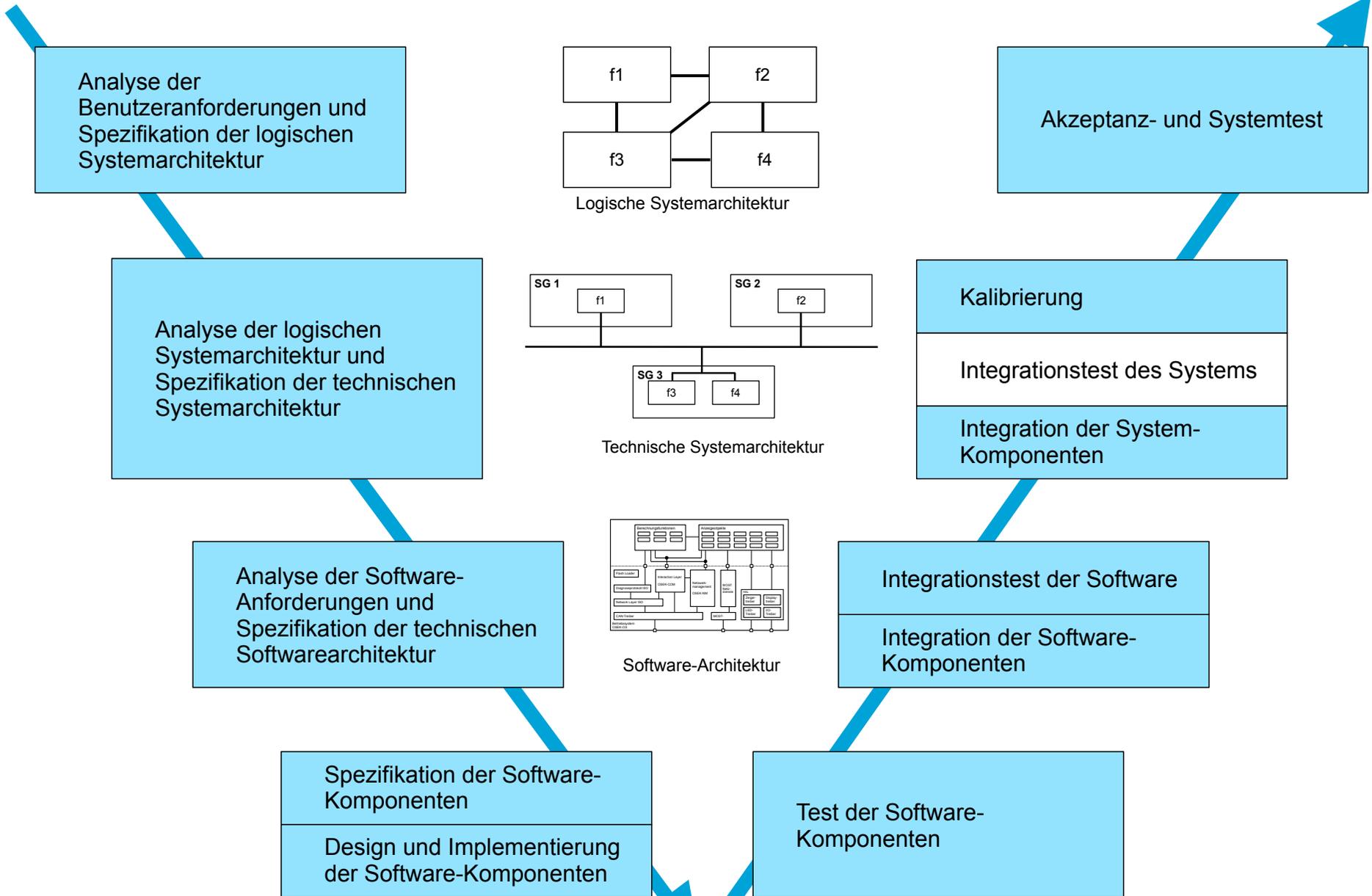
Integrations- und Testschritte (Nach Schäuuffele, Zurawka)



Integrations- und Testschritte (Nach Schäuuffele, Zurawka)



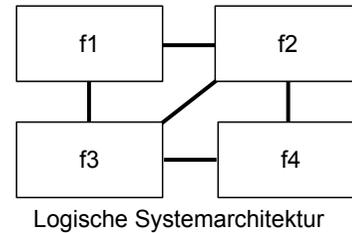
Integrations- und Testschritte (Nach Schäuuffele, Zurawka)



Integrations- und Testschritte (Nach Schäuuffele, Zurawka)

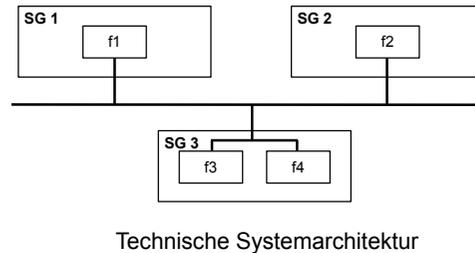


Analyse der Benutzeranforderungen und Spezifikation der logischen Systemarchitektur



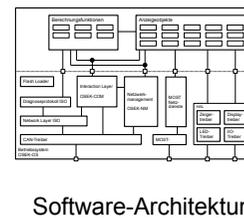
Akzeptanz- und Systemtest

Analyse der logischen Systemarchitektur und Spezifikation der technischen Systemarchitektur



Kalibrierung
Integrationstest des Systems
Integration der System-Komponenten

Analyse der Software-Anforderungen und Spezifikation der technischen Softwarearchitektur



Integrationstest der Software
Integration der Software-Komponenten

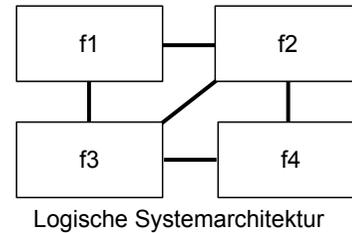
Spezifikation der Software-Komponenten
Design und Implementierung der Software-Komponenten

Test der Software-Komponenten

Integrations- und Testschritte (Nach Schäuuffele, Zurawka)

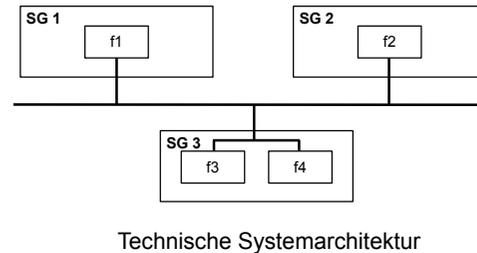


Analyse der Benutzeranforderungen und Spezifikation der logischen Systemarchitektur



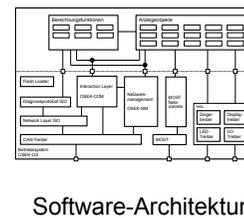
Akzeptanz- und Systemtest

Analyse der logischen Systemarchitektur und Spezifikation der technischen Systemarchitektur



Kalibrierung
Integrationstest des Systems
Integration der System-Komponenten

Analyse der Software-Anforderungen und Spezifikation der technischen Softwarearchitektur



Integrationstest der Software
Integration der Software-Komponenten

Spezifikation der Software-Komponenten
Design und Implementierung der Software-Komponenten

Test der Software-Komponenten

6. SW-Entwicklung / 2. Unterstützungsprozesse

Unterstützungsprozesse für die Embedded Software Entwicklung



1. Vorgehensmodelle und Standards
2. Konfigurationsmanagement
3. Projektmanagement
4. Lieferantenmanagement
5. Anforderungsmanagement

6. Qualitätssicherung

1. Integrations- und Testschritte

2. Maßnahmen zur Qualitätssicherung von Software

Motivation



Tests können die Anwesenheit von Fehlern zeigen, nie aber deren Abwesenheit.

Edsger W. Dijkstra

Ein fehlender Programmzweig ... wird auch durch Austesten aller vorhandenen Programmzweige nicht gefunden.

Bernhard Hohlfeld

Software wird hauptsächlich getestet

- Testverfahren können nie alle Systemzustände erreichen
 - Wurde ausreichend getestet?
- Software muss für Tests vorhanden und ausführbar sein
 - Ist die Spezifikation korrekt?

Interne Qualitätseigenschaften werden selten geprüft

- Ist die Software zuverlässig, wartbar, änderbar etc. ?
- Ist die Software effizient?
- Wie viel Speicher/Zeit verbraucht das System maximal?

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, ...

Unterstützungsprozesse aus Sicht des Entwicklers (1)



Unterstützungsprozesse aus Sicht des Entwicklers (2)



Entwickler aus Sicht der Unterstützungsprozesse (1)



Entwickler aus Sicht der Unterstützungsprozesse (2)

