

Simulation der SW-Systemzuverlässigkeit in Automatisierungssystemen auf Grundlage von SW-Komponenten

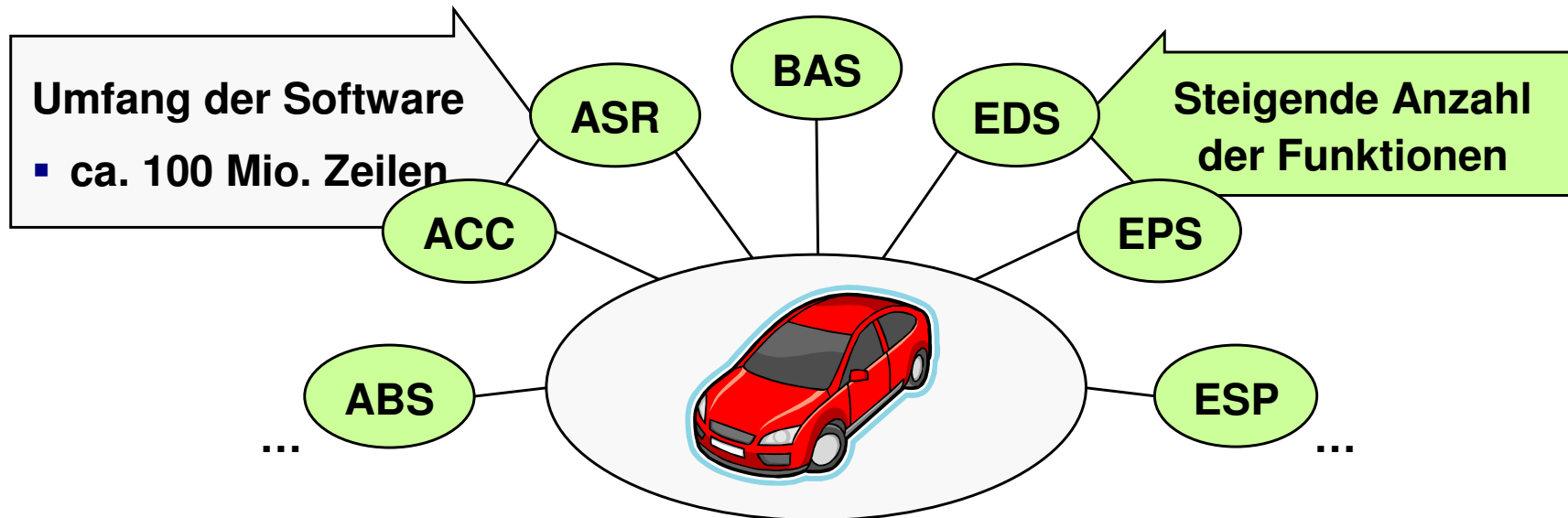
Dipl.-Ing. Michael Wedel
Prof. Dr.-Ing. Dr. h. c. Peter Göhner

AUTOMOTIVE 2008 – Sicherheit und Zuverlässigkeit für automobiler Informationstechnik
19. - 20. November 2008, Auditorium Boschzentrum, Stuttgart-Feuerbach

Einfluss der Zuverlässigkeit auf die Kaufentscheidung

Kauf eines Neuwagens: Zuverlässigkeit ist wichtigstes Kriterium

[DAT-Report 2007]



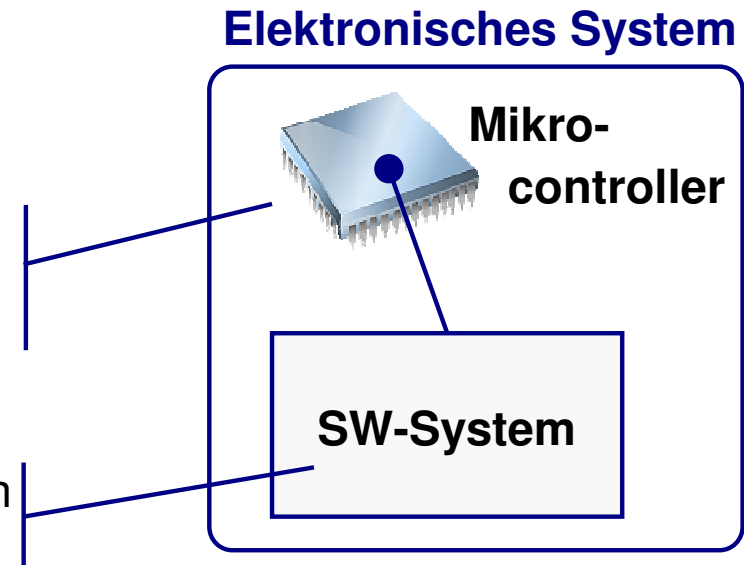
- **Spät entdeckte Fehler in der Software führen zu...**
 - **Versagen der Funktionen im Feld**
 - **Hohen Kosten und hohem Aufwand zur Fehlerbehebung**
- **Ziel: Frühzeitige Betrachtung der SW-Zuverlässigkeit**

Inhalt

- **Grundlagen der SW-Zuverlässigkeit**
- Komponentenbasierte SW-Entwicklung
- Berechnung der Zuverlässigkeit eines komponentenbasierten SW-Systems
- Frühzeitige Simulation der Zuverlässigkeit von SW-Systemen
- Zusammenfassung

Einfluss der Software auf die Funktionalität eines Kfz

- **Anzahl und Komplexität elektronischer Systeme** im Kraftfahrzeug steigt
- *Beispiele in modernen Kraftfahrzeugen:*
 - *Fahrerassistenzsysteme (ABS, ACC, ASR, ...)*
 - *Klimaanlage, Navigation und Multimedia (DVD, Radio, TV, ...)*
 - *Aktive Sicherheitssysteme*
 - *Motorsteuerung*
 - ...
- **Aufbau der elektronischen Systeme** aus den Kernbestandteilen Mikrocontroller und SW-System
- **Steigender Anteil der Software** in Kraftfahrzeugen
- Prognose für das Jahr 2010
 - Speicherplatz für Quellcode ca. 1 Gigabyte (entspricht ca. 200.000 DIN-A4-Seiten)
 - 13% Anteil am Gesamtwert eines Fahrzeugs



Definitionen zur SW-Zuverlässigkeit

- **Software Reliability Engineering** = Systematisches Vorgehen bestehend aus
 - Spezifikation der **geforderten SW-Zuverlässigkeit**
 - **Test auf Einhaltung** dieser SW-Zuverlässigkeit

- **SW-Zuverlässigkeit** = Wahrscheinlichkeit, dass das SW-System ohne Versagen funktioniert

Versagen = Reaktion des SW-Systems auf Eingaben widerspricht der Spezifikation

Wahrscheinlichkeit = Relative Häufigkeit eines Ereignisses (hier: Versagen des SW-Systems)

$$R_{SW} = 1 - \frac{\text{Häufigkeit **Versagen** des SW-Systems}}{\text{Häufigkeit **aller Ausführungen**}}$$

$$\text{z. B. } 1 - \frac{40}{200} = 0,80$$

Frage: Wie wird die Einhaltung der SW-Zuverlässigkeit getestet?

Test auf Einhaltung der geforderten SW-Zuverlässigkeit

▪ Tätigkeiten beim Test

- Aus späteren Betriebsbedingungen Eingaben für Test ableiten → Testfälle
- Reaktion des SW-Systems auf Testfälle speichern → Testergebnisse
- Testergebnisse mit Spezifikation vergleichen → Empirische Daten über Versagen
- Empirische Daten in Definition der SW-Zuverlässigkeit einsetzen → R_{SW}

▪ Vorbedingungen für den Test

- Technische Umgebung und Elektronik zur Ausführung vorhanden
- SW-System entwickelt und Teile integriert

→ Probleme:

- **Vorbedingungen erst spät im Entwicklungsprozess erfüllt**
- **Abgleich mit geforderter SW-Zuverlässigkeit nicht frühzeitig möglich**

→ **Idee: Verwendung von empirischen Daten aus der mehrfachen Verwendung von SW-Komponenten zur frühzeitigen Berechnung der SW-Zuverlässigkeit**

Inhalt

- Zuverlässigkeit von SW-Systemen
- **Komponentenbasierte SW-Entwicklung**
- Berechnung der Zuverlässigkeit komponentenbasierter SW-Systeme
- Frühzeitige Simulation der Zuverlässigkeit von SW-Systemen
- Zusammenfassung

Trend zur systematischen Mehrfachverwendung von Software

▪ Ziele der Mehrfachverwendung

- Senkung von Entwicklungskosten
- Steigerung der Qualität
- Reduktion der Entwicklungstiefe: Konzentration auf Kerngeschäft und Auslagerung von Entwicklungsaktivitäten an Zulieferer

▪ Zunehmende Standardisierung

z. B. AUTOSAR, EASIS

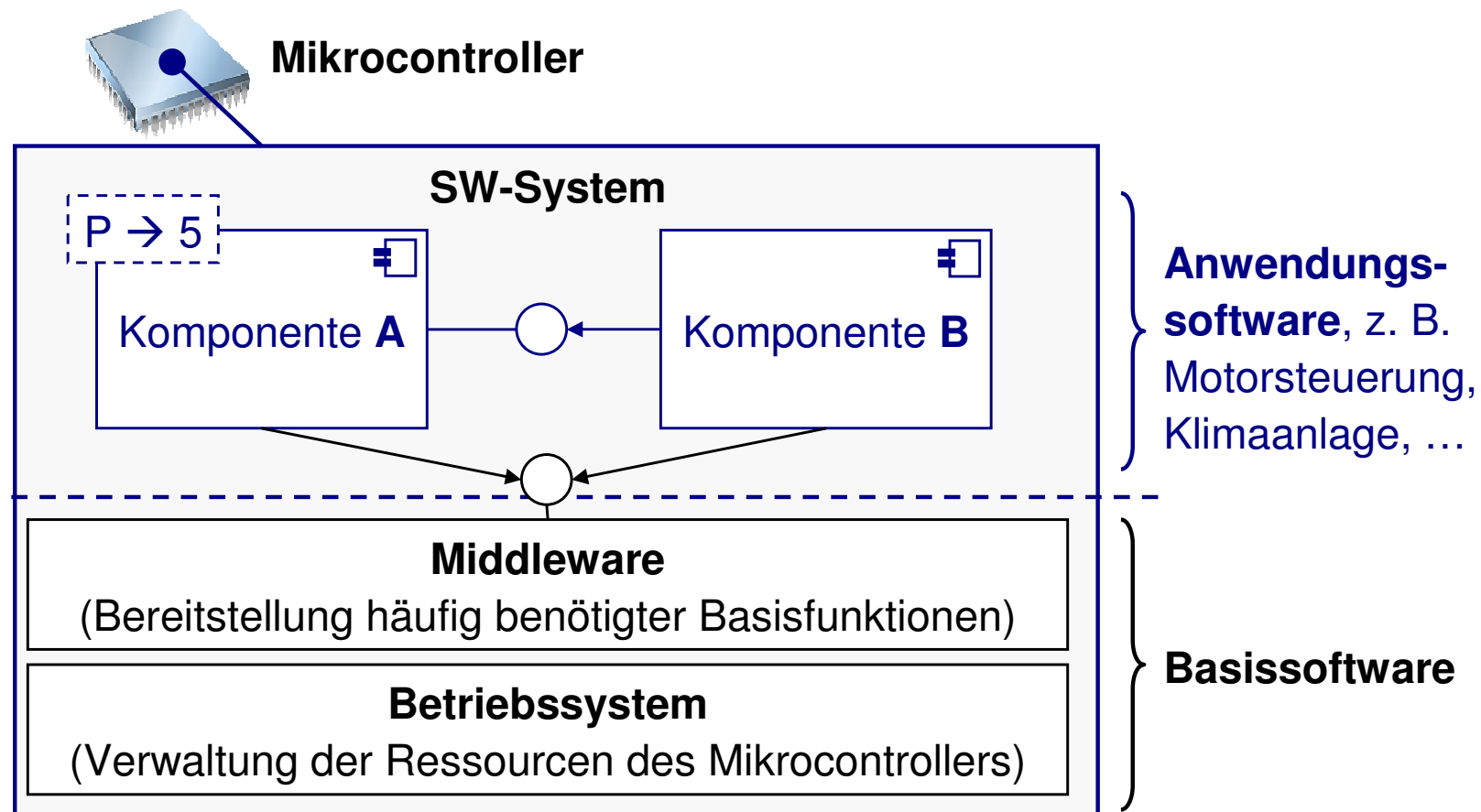
- Einheitliche Komponentenmodelle und Schnittstellendefinition erleichtern Entwicklung mehrfach verwendbarer Software
- Vorgabe grundlegender Architekturen ermöglichen Austausch von Software über Zulieferer und Hersteller hinweg

➔ **Mehrfachverwendung von Software, die häufig benötigte Funktionen bereitstellt**

Frage: Wie wird Software aus mehrfach verwendbaren Teilen aufgebaut?

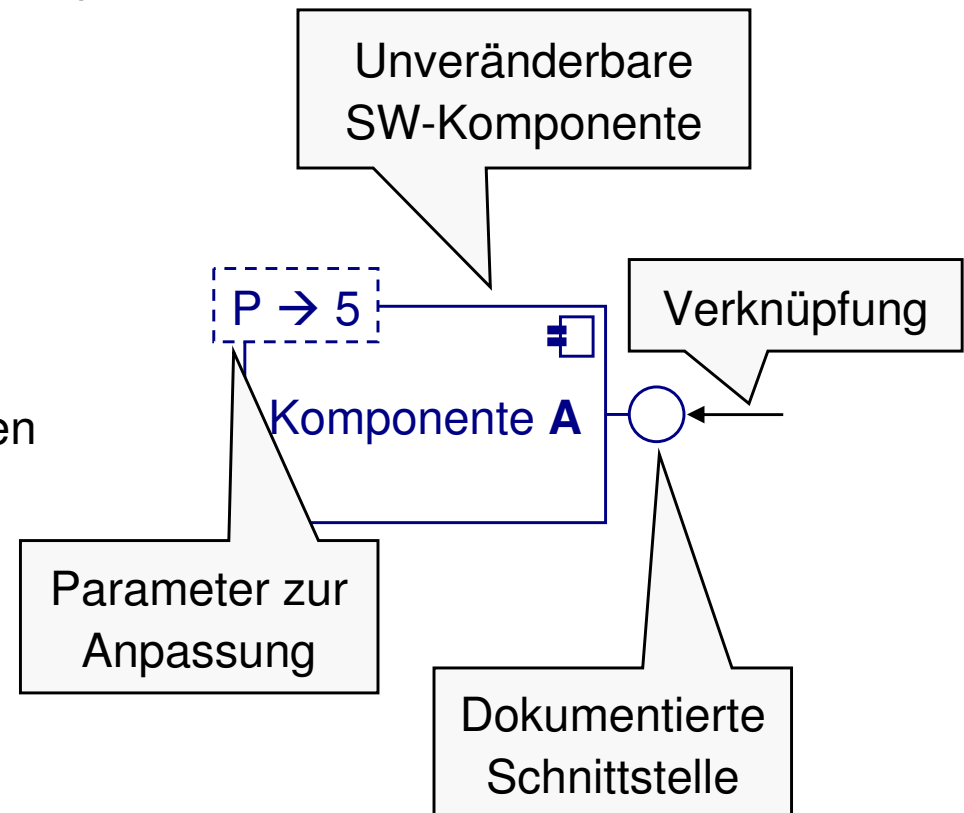
Konstruktion des SW-Systems durch Mehrfachverwendung

- **Schematische SW-Architektur** in einem elektronischen System
 - **Basissoftware:** stellt über **mehrere SW-Systeme** hinweg häufig benötigte Dienste über Schnittstellen als **Grundlage für die Anwendung** bereit
 - **Anwendungssoftware:** realisiert Funktionen des elektronischen Systems durch Kombination **mehrfach verwendbarer SW-Komponenten**



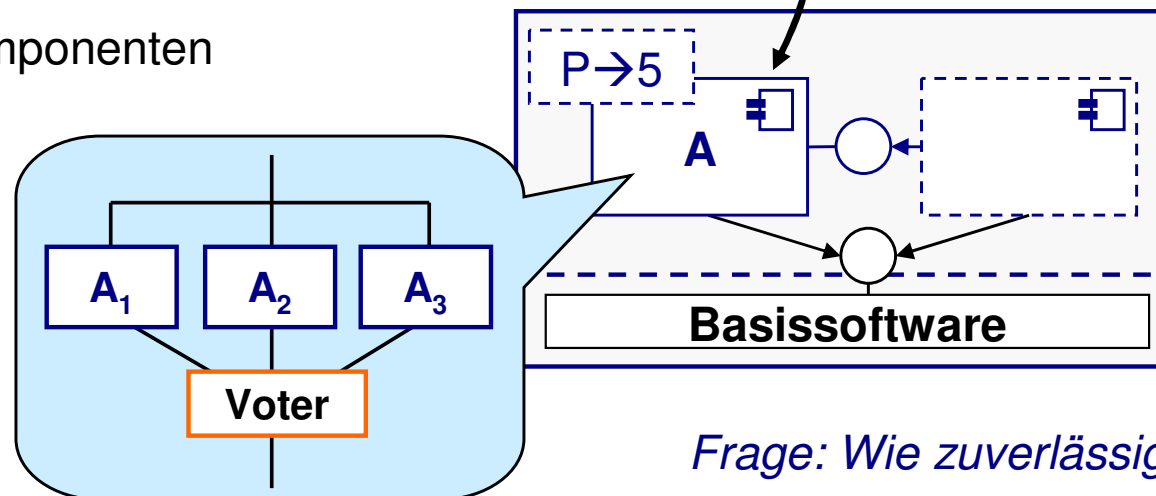
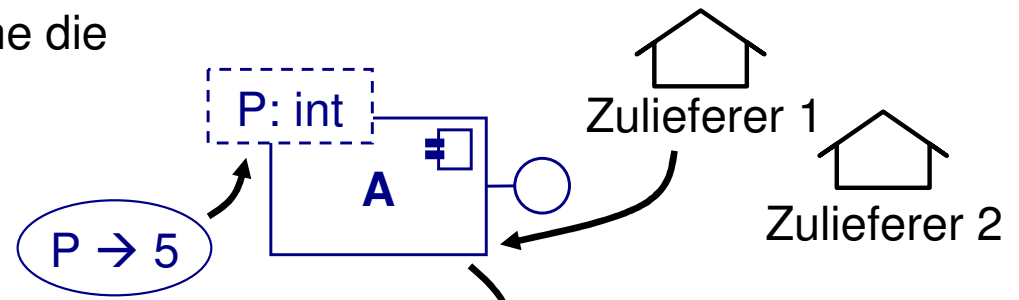
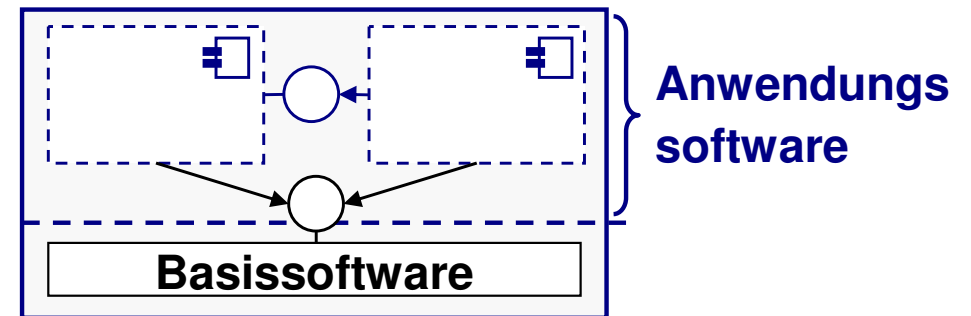
Definition der Merkmale einer SW-Komponente

- **Unveränderbarkeit durch Dritte**
 - Mehrfache Verwendung in verschiedenen SW-Systemen ohne Änderungen am Inneren der SW-Komponente
- **Funktionale Geschlossenheit**
 - Implementierung im Inneren verborgen
 - Bereitstellung der Funktionen über dokumentierte Schnittstellen
- **Strukturelle Unabhängigkeit**
 - Keine Abhängigkeit von einer bestimmten anderen SW-Komponente
- **Verknüpfbarkeit**
 - Verknüpfung mit anderen SW-Komponenten über Schnittstellen
- **Anpassbarkeit**
 - Anpassung des Verhaltens und des Funktionsumfangs an die jeweiligen Anforderungen der Anwendung



Komponentenbasierte Entwicklung der Anwendungssoftware

- **Definition der SW-Architektur**
 - Schnittstellen und Verknüpfungen zur Realisierung der SW-Systemfunktionen
- **Auswahl** von SW-Komponenten, welche die erforderliche Funktionalität bereitstellen
- **Anpassung** an die Systemumgebung, z. B. durch Einstellung von Parametern
- **Verknüpfen** der SW-Komponenten
- **Fehlertolerante Maßnahmen** zur Steigerung der Zuverlässigkeit, z. B. diversitäre SW-Komponenten



Frage: Wie zuverlässig ist die entwickelte Anwendungssoftware?

Inhalt

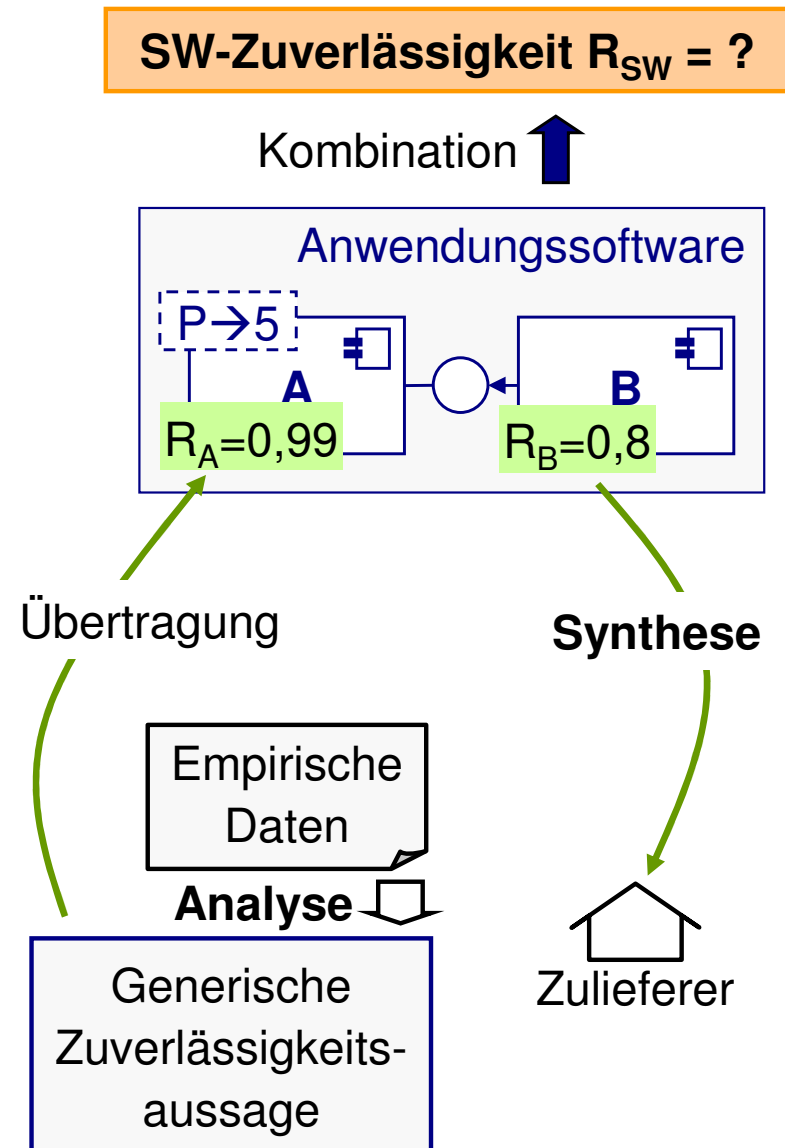
- Zuverlässigkeit von SW-Systemen
- Komponentenbasierte SW-Entwicklung
- **Berechnung der Zuverlässigkeit komponentenbasierter SW-Systeme**
- Frühzeitige Simulation der Zuverlässigkeit von SW-Systemen
- Zusammenfassung

Lösungsansatz zur Berechnung der SW-Zuverlässigkeit in frühen Entwicklungsphasen

- **Grundlegende Schritte der Berechnung**
 - Ermittlung der Zuverlässigkeiten der einzelnen SW-Komponenten
 - Kombination zur SW-Zuverlässigkeit aufgrund der SW-Architektur

Zuverlässigkeiten der einzelnen SW-Komponenten

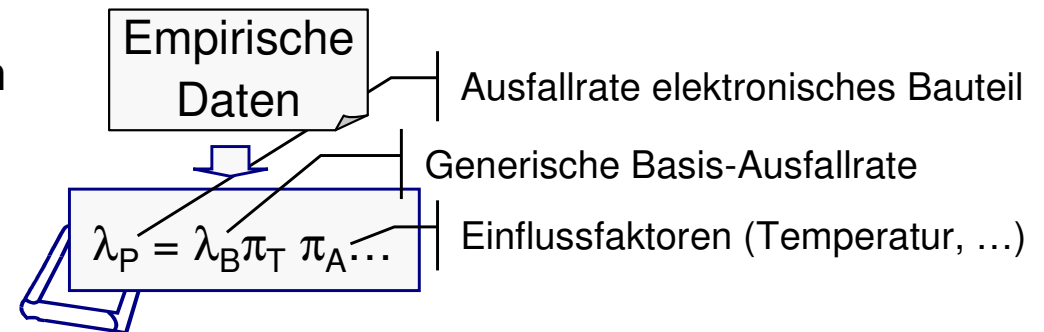
- Angabe einer **generischen Zuverlässigkeitsaussage** durch den Zulieferer
 - **Analyse** empirischer Daten aus dem bisherigen Test und Betrieb
 - **Übertragung** auf die neue Software
- Vorgabe einer **geforderten Zuverlässigkeit** für eine SW-Komponente an den Zulieferer
 - **Synthese**, so dass die Zuverlässigkeit der Anwendungssoftware erreicht wird



Angabe generischer Zuverlässigkeitsaussagen durch den Zulieferer

- **Analogie zu elektronischen Bauteilen**

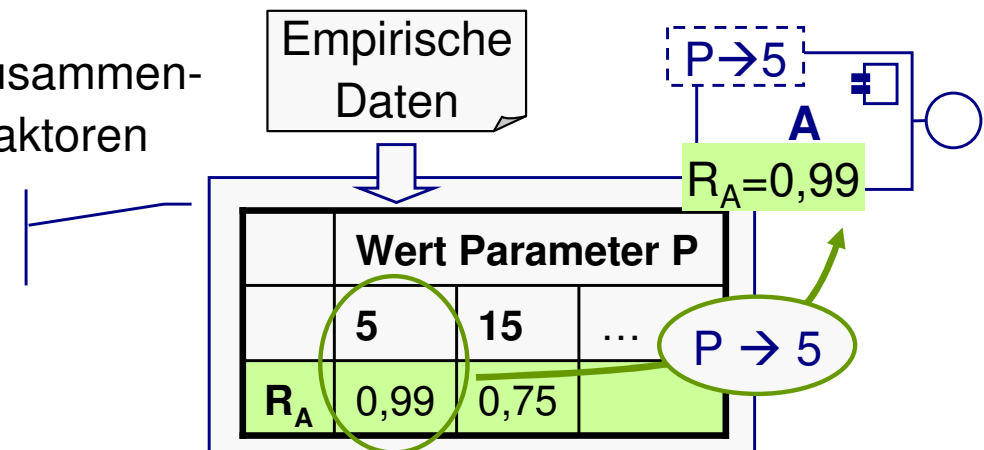
- Modellierung der Zuverlässigkeit in Abhängigkeit physikalischer **Einflussfaktoren**



- Einflussfaktoren für **SW-Komponenten** ergeben sich aus den definierten Merkmalen
 - Anpassbarkeit: **Parameter** beeinflussen das Verhalten der SW-Komponente, z. B. *An-/Abschalten einzelner Funktionen*
 - Verknüpfbarkeit: **Unterschiedliche Aufrufe** der Funktionen an den Schnittstellen

- **Generische Zuverlässigkeitsaussage** = Zusammenhang zwischen Zuverlässigkeit und Einflussfaktoren

- Einfache Zusammenhänge durch Tabellenangaben
- Komplexe Zusammenhänge mittels mathematischer Regressionsmodelle

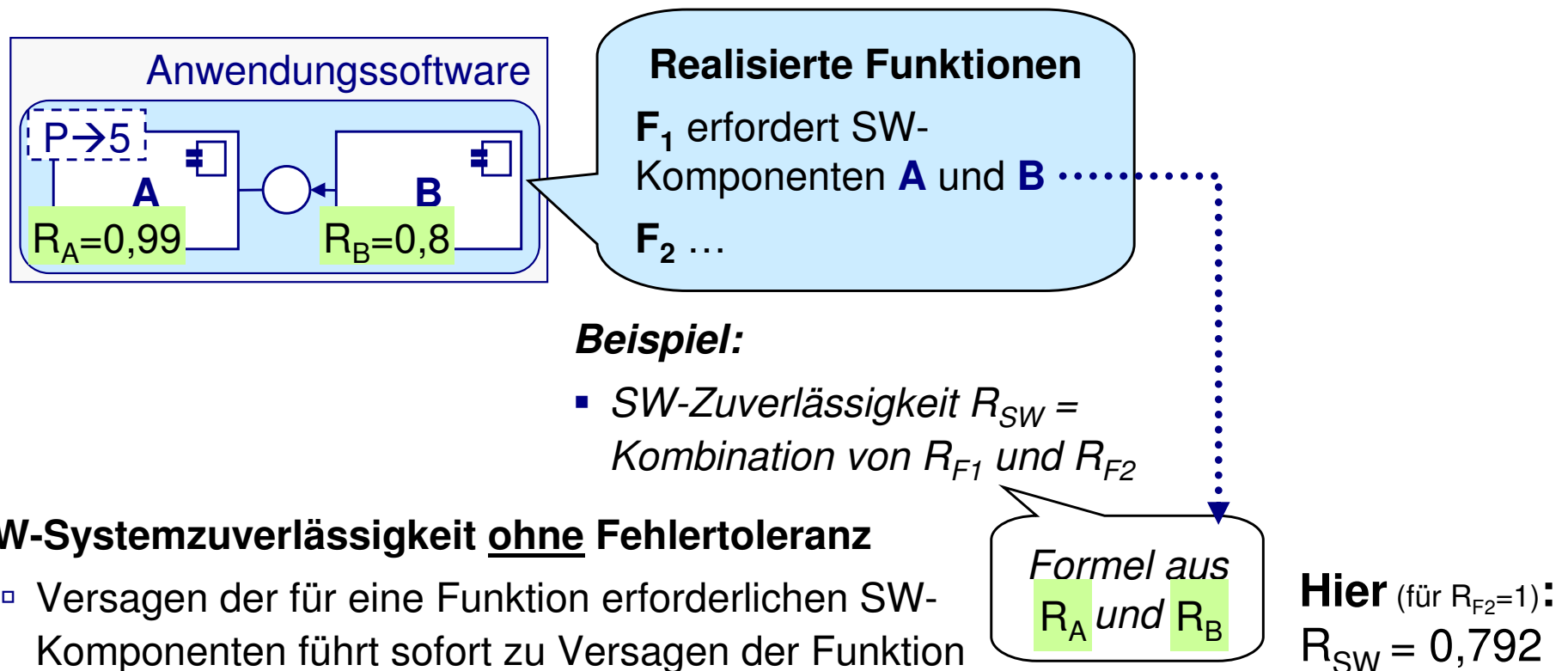


Kombination aufgrund der SW-Architektur (1)

- **SW-Zuverlässigkeit** errechnet sich aus der Zuverlässigkeit **der realisierten Funktionen**

- Analyse, welche SW-Komponenten jeweils zu einer Funktion beitragen
- Kombination der einzelnen Zuverlässigkeiten in der Reihenfolge

SW-Komponenten → Realisierte Funktionen → Anwendungssoftware

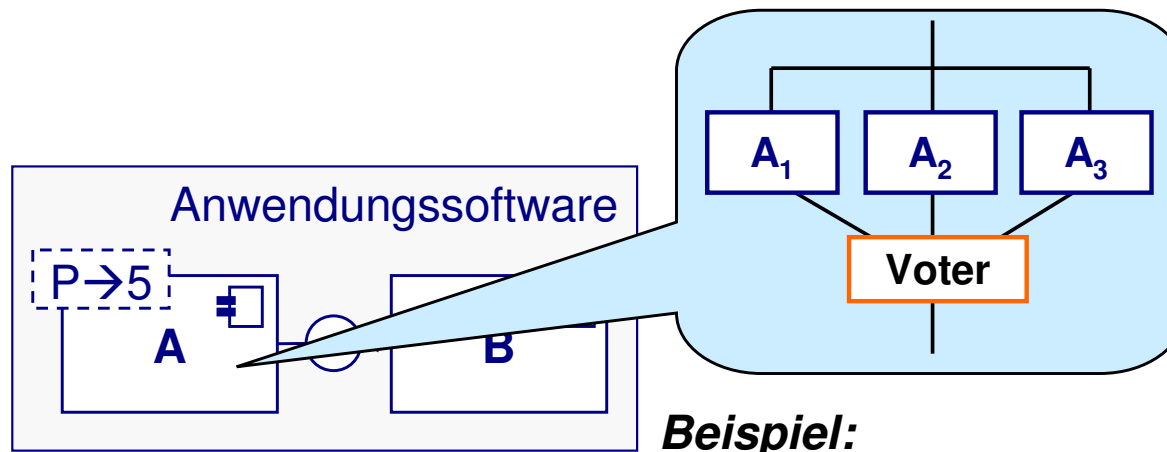


- **SW-Systemzuverlässigkeit ohne Fehlertoleranz**

- Versagen der für eine Funktion erforderlichen SW-Komponenten führt sofort zu Versagen der Funktion

Kombination aufgrund der SW-Architektur (2)

- Zusätzlich Modellierung und Analyse von **fehlertoleranten Maßnahmen**



Beispiel:

- ⊙ Fehlermaskierung von SW-Komponente **A** mit Mehrheitsentscheid (2-aus-3-Voting)

- **SW-Systemzuverlässigkeit mit Fehlertoleranz**

- Anpassung der Zuverlässigkeit der SW-Komponenten entsprechend der realisierten fehlertoleranten Maßnahmen

Angepasste Formel für A

$$R_{A,Red} = \sum_{i=m}^n \binom{n}{i} R_A^i (1 - R_A)^{n-i}$$

Hier: $R_{SW,Red} = 0,887$

Frage: Wie kann dies zum frühzeitigen Abgleich mit der geforderten SW-Zuverlässigkeit genutzt werden?

Inhalt

- Zuverlässigkeit von SW-Systemen
- Komponentenbasierte SW-Entwicklung
- Berechnung der Zuverlässigkeit komponentenbasierter SW-Systeme
- **Frühzeitige Simulation der Zuverlässigkeit von SW-Systemen**
- Zusammenfassung

Lösungsansatz zur Simulation der SW-Zuverlässigkeit

- **Ziel:** Entwicklung einer Anwendungssoftware mit der geforderten SW-Zuverlässigkeit (möglichst geringe Abweichung nach oben oder unten)
- **Vorgehen zur frühzeitigen Simulation**
 - Entwurf verschiedener komponentenbasierter Systemszenarien
 - Berechnung der SW-Zuverlässigkeit für die verschiedenen Szenarien
 - Abgleich mit der geforderten SW-Zuverlässigkeit

Entwurf von Szenarien durch unterschiedliche

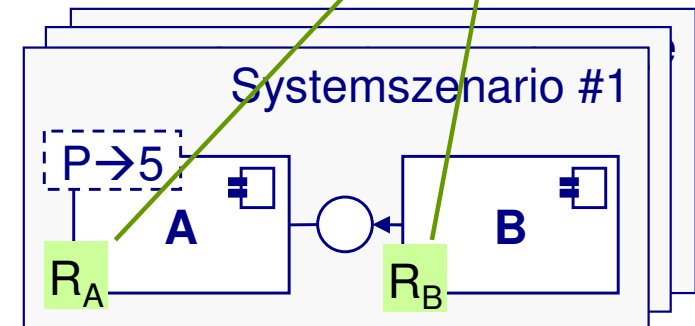
- Definition der SW-Architektur
- Auswahl der SW-Komponenten
- Anpassung und Verknüpfung der SW-Komponenten
- Fehlertolerante Maßnahmen

Geforderte Zuverlässigkeit

Abgleich \updownarrow

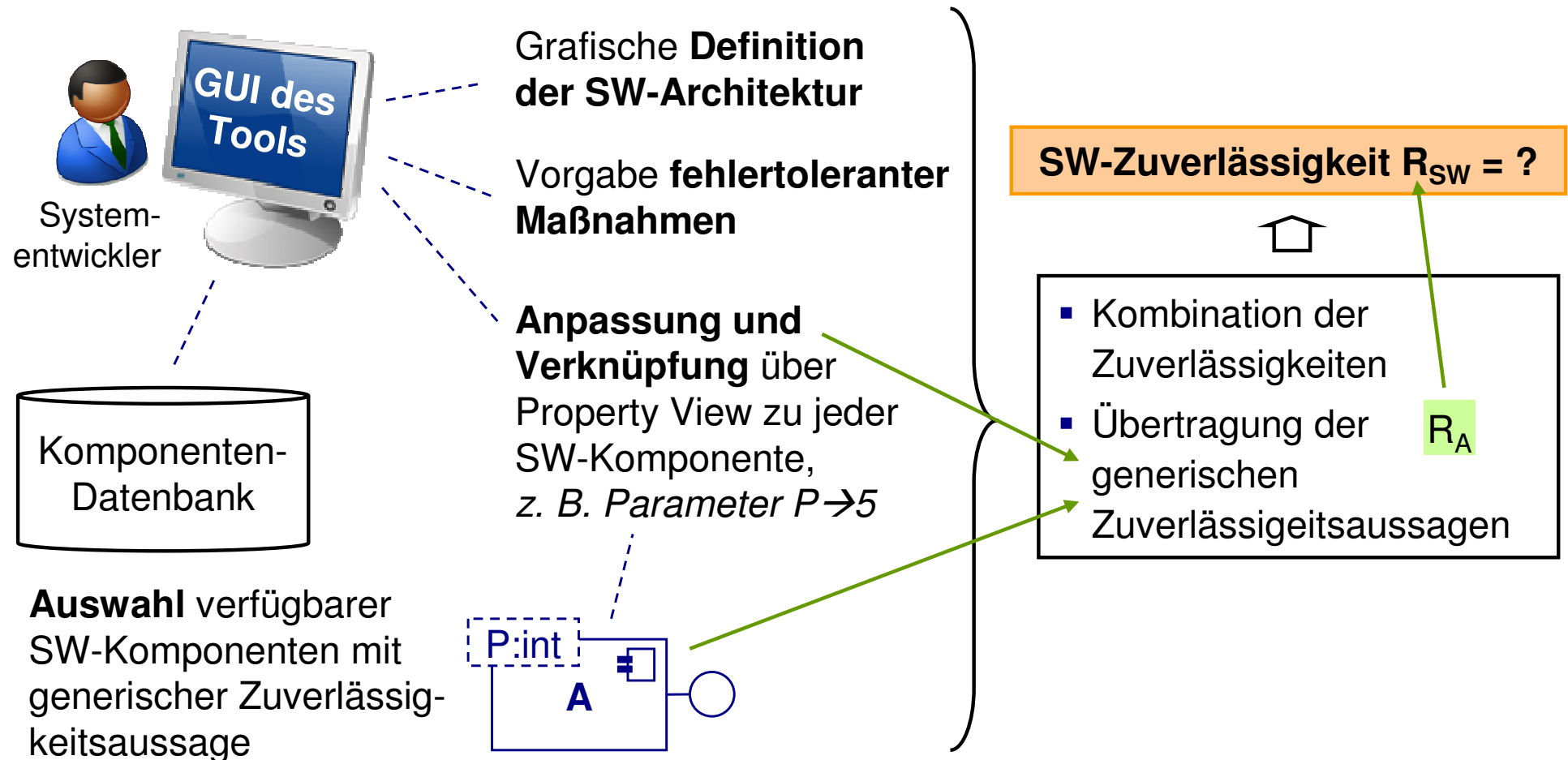
SW-Zuverlässigkeit $R_{SW} = ?$

Berechnung: Kombination der Zuverlässigkeiten \uparrow



Tool-Unterstützung zur Simulation verschiedener Szenarien

- Entwurf von Szenarien und Berechnung der jeweiligen Zuverlässigkeit durch den Systementwickler



- Realisiert als **Plugin** für die erweiterbare **Eclipse-Plattform**

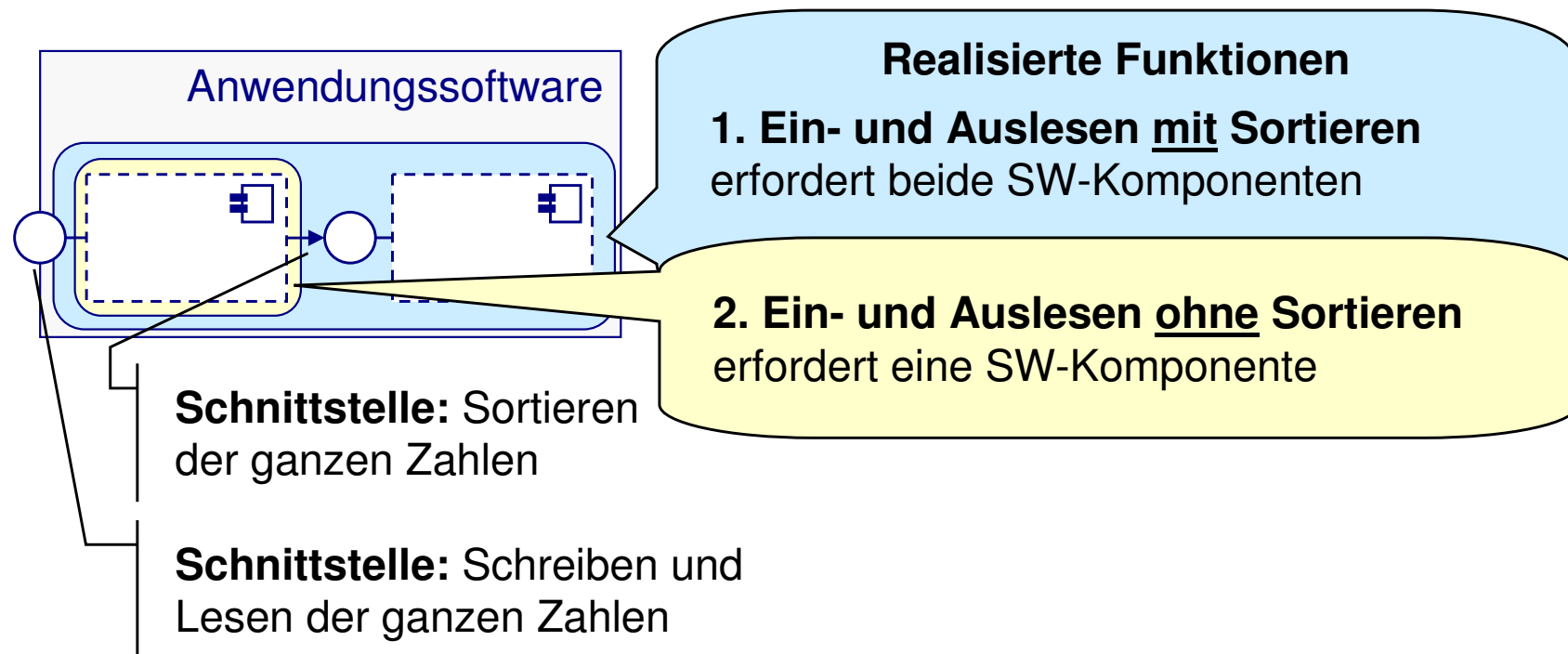
Beispiel für die Simulation anhand einer einfachen Anwendung

▪ Geforderte Funktionen:

- Schreiben von ganzen Zahlen in eine Datei
- Auslesen der Zahlen aus einer Datei
- Sortieren der ausgelesenen Zahlen vor Ausgabe in einzelnen Fällen

▪ Geforderte SW-Zuverlässigkeit $R_{SW} = 0,9$

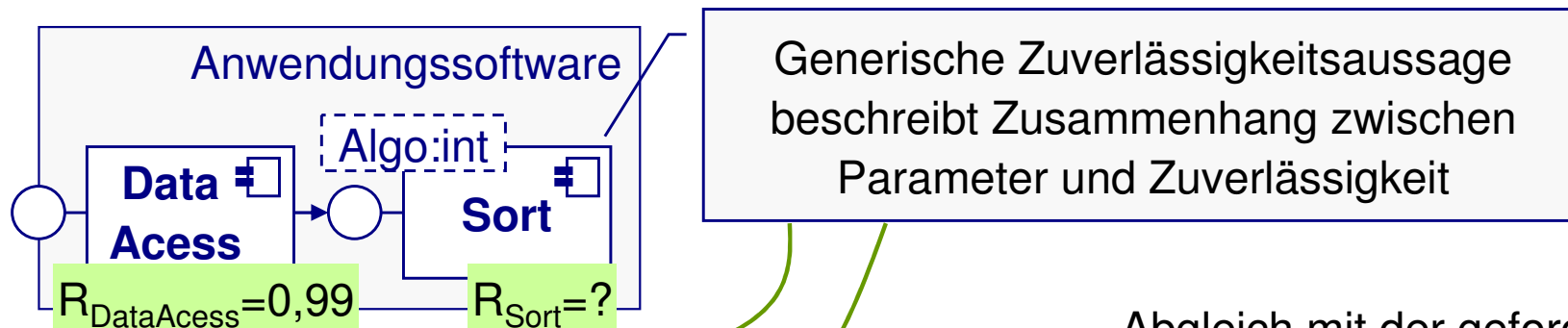
▪ Definition der grundlegenden SW-Architektur (Schnittstellen & Verknüpfungen)



Simulation der ausgewählten SW-Komponenten mit unterschiedlichen Parametereinstellungen

▪ Auswahl der SW-Komponenten

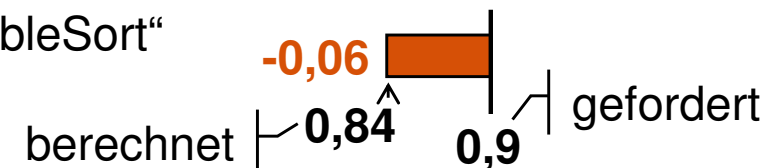
- **DataAccess:** Funktionen zum Schreiben und Lesen in eine Datei
- **Sort:** Drei Sortier-Funktionen mit unterschiedlichen Algorithmen, Auswahl des Algorithmus durch Parametereinstellung



Abgleich mit der geforderten SW-Zuverlässigkeit

▪ 1. Szenario: Verwendung des Algorithmus 1 „BubbleSort“

- $R_{Sort} = 0,85 \rightarrow R_{SW} = 0,84$



P → 1

▪ 2. Szenario: Verwendung des Algorithmus 2 „ShakerSort“

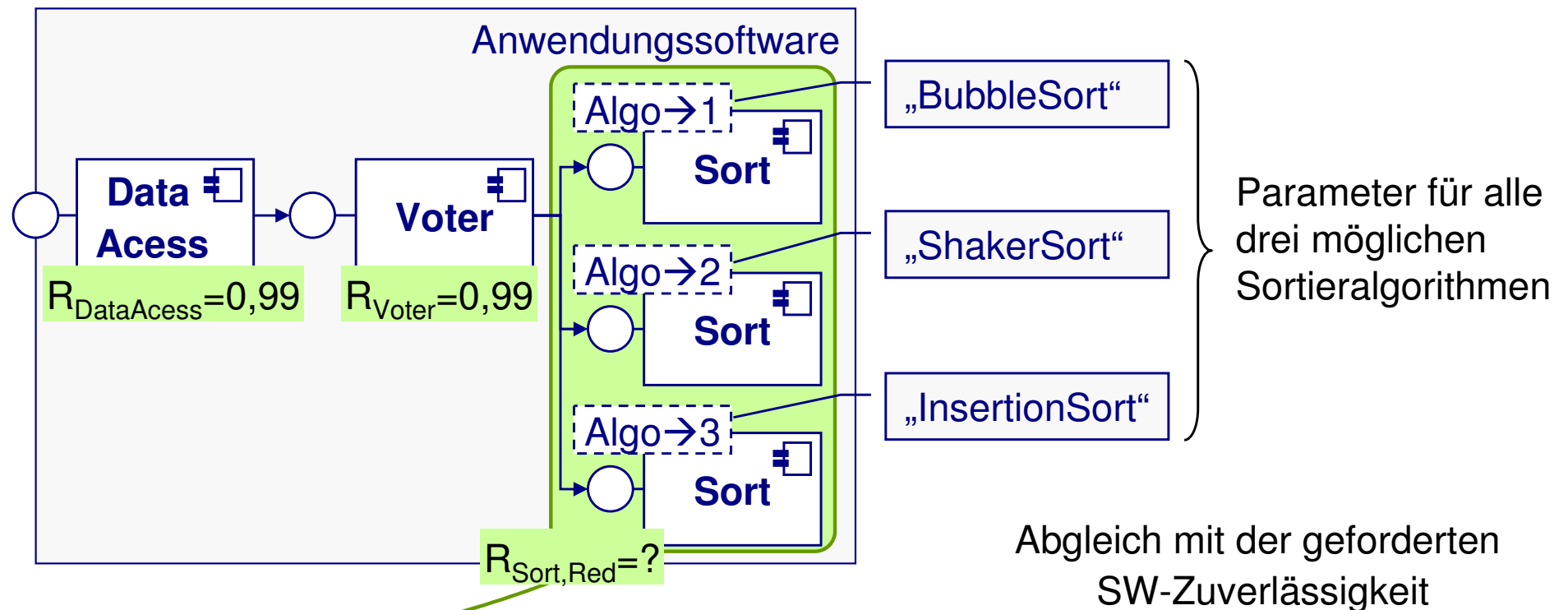
- $R_{Sort} = 0,88 \rightarrow R_{SW} = 0,87$



P → 2

Simulation fehlertoleranter Maßnahmen

- **Fehlermaskierung:** Mehrheitsentscheid unter mehreren möglichen Ergebnissen



- **3. Szenario:** Parallele Ausführung aller Sortieralgorithmen

- $R_{Sort,Red} = 0,95 \rightarrow R_{SW} = 0,93$



→ Auswahl eines Szenarios durch Abgleich mit der geforderten SW-Zuverlässigkeit

Inhalt

- Zuverlässigkeit von SW-Systemen
- Komponentenbasierte SW-Entwicklung
- Berechnung der Zuverlässigkeit komponentenbasierter SW-Systeme
- Frühzeitige Simulation der Zuverlässigkeit von SW-Systemen
- **Zusammenfassung**

Zusammenfassung

- **Frühzeitige Berechnung der SW-Zuverlässigkeit** auf Grundlage **mehrfach verwendbarer SW-Komponenten**
- Ingenieur kann die SW-Zuverlässigkeit durch **verschiedene Faktoren bei der komponentenbasierten Entwicklung** beeinflussen
- **Simulation** der verschiedenen Szenarien ermöglicht **frühzeitigen Abgleich** der **berechneten** mit der **geforderten SW-Zuverlässigkeit**

Ausblick

- **Evaluierung der Simulationsergebnisse:** Verwendung von SW-Komponenten aus großen Open-Source-Projekten zum Vergleich

Danksagung

Deutsche
Forschungsgemeinschaft



- Unterstützung im Rahmen der DFG-Forschergruppe 460 „System-Zuverlässigkeit“

Vielen Dank für Ihr Interesse

www.ias.uni-stuttgart.de

michael.wedel@ias.uni-stuttgart.de

peter.goehner@ias.uni-stuttgart.de