

# Software-Engineering II

## Sommersemester 2017

---

Herzlich willkommen.

SE II - Sommersemester 2017				zeitlicher Ablauf				
		Praktika				Vorlesung		
		Do, 3.DS	Mi, 4.DS	Mi, 3.Ds	Fr, 2. Ds	Die	Do	
		15/041/01+61	15/042/01	15/042/61	15/042/01+62	VO/041	VO/042	
KW								
12	20.3.	-	-	-	1	1	1	
13	27.3.	1	1	1	2	2	2	Themenausgabe für die BGA
14	3.4.	2	2	2	3	3	3	
15	10.4.	3	3	3	-	4	4	Ostern
16	17.4.	4	4	4	4	5	5	
17	24.4.	5	5	5	5	6	6	
18	1.5.	6	6	6	6	7	7	Abgabe Teil 1 der BGA
19	8.5.	7	7	7	7	8	8	
20	15.5.	8	8	8	8	9	9	
21	22.5.	-	9	9	-	10	-	Himmelfahrt
22	29.5.	9	10	10	9	11	10	
23	5.6.	10	11	11	10	-	11	Pfingsten
24	12.6.	11	-	-	11	12	12	DIES ac.
25	19.6.	12	12	12	12	13	13	
26	26.6.	13	13	13	13	14	14	
27	3.7.	14	14	14	14	15	15	Abgabe Teil 2 der BGA
BGA: Belegarbeit								
Zeitraum für die Präsentationen zur BGA (Teilnahme aller Gruppenmitglieder erforderlich ):								
26. KW	Zusatz (freiwillig)							
27. KW								
28. KW	Prüfungsphase (Terminvereinbarung)							
29. KW								

60 %				40 %
Pflichten- heft	Produkt	Projekt- arbeit	Dokumen- tation	Präsen- tation
30 Pkt.	30 Pkt.	30 Pkt.	30 Pkt.	80 Pkt.
Gruppennote				individuelle Note

Gruppe:

Projektleiter/in

Verantwortliche(r) für:

- Analyse
- Entwurf (Grob-, Fein-)
- Implementierung
- Datenbank (wenn vorhanden)
- Test
- Dokumentation
- Qualitätssicherung

Worin bestand meine Aufgabe ?

Wie habe ich sie gelöst?

→ Welche Fachinhalte habe ich angewandt?

Welche Ergebnisse liegen vor?

**Reflexion:**

Was habe ich gelernt?

(Welche Kompetenzen habe ich erreicht/ausgebaut ?)

Was würde ich wieder so machen?

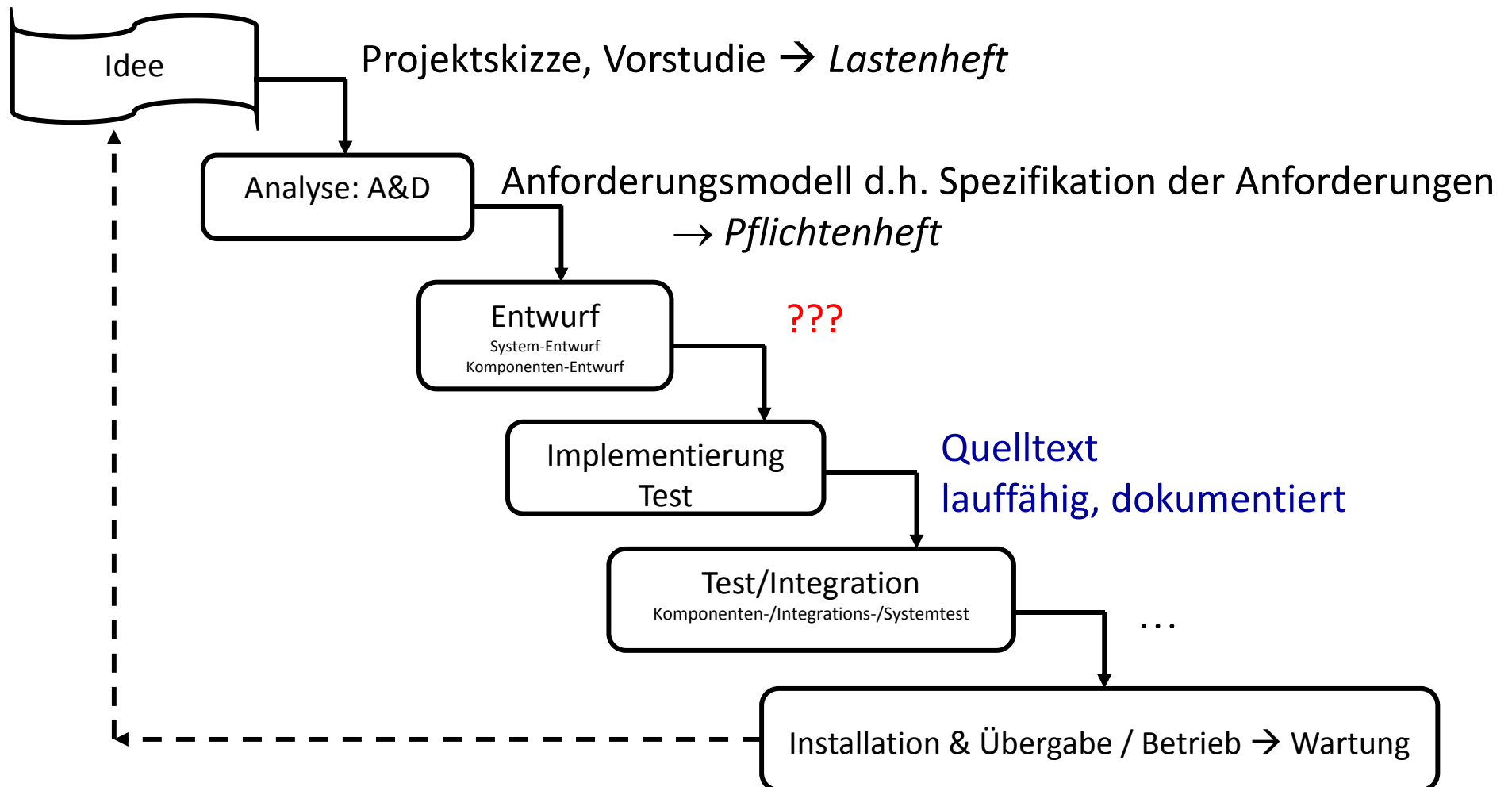
Was würde ich anders machen?

→ Worin liegen meine Stärken?

[illegible]

Wozu lädt das Modul Software Engineering ein?

# SW-Lebenszyklus



...

Klausur:

Was ist ein System?

Es dient einem Zweck, ist in sich abgeschlossen,  
hat Kontakt zu seiner Umwelt über Schnittstellen,

besteht aus Komponenten, die Beziehungen zueinander haben.

Die Anordnung dieser Komponenten entspricht der Struktur des Systems.

Diese statische Struktur ist Basis für das dynamische Verhalten des Systems.

Was ist das Besondere an einem SW-System?

- hoch komplex
- immateriell
- keine natürliche Lokalität
- Werkstoff = Sprache
  - Die Sprache erfordert zwar die Strukturierung im Kleinen  
aber nicht die Strukturierung im Großen.

→ Diese Strukturierung im Großen ist Aufgabe der Entwerfer.

„Wer ein komplexes System schafft,  
legt Strukturen fest, die lange Zeit gültig bleiben.

Darum gilt **das Entwerfen** auch in der Software-Entwicklung als  
**die zentrale Tätigkeit.**

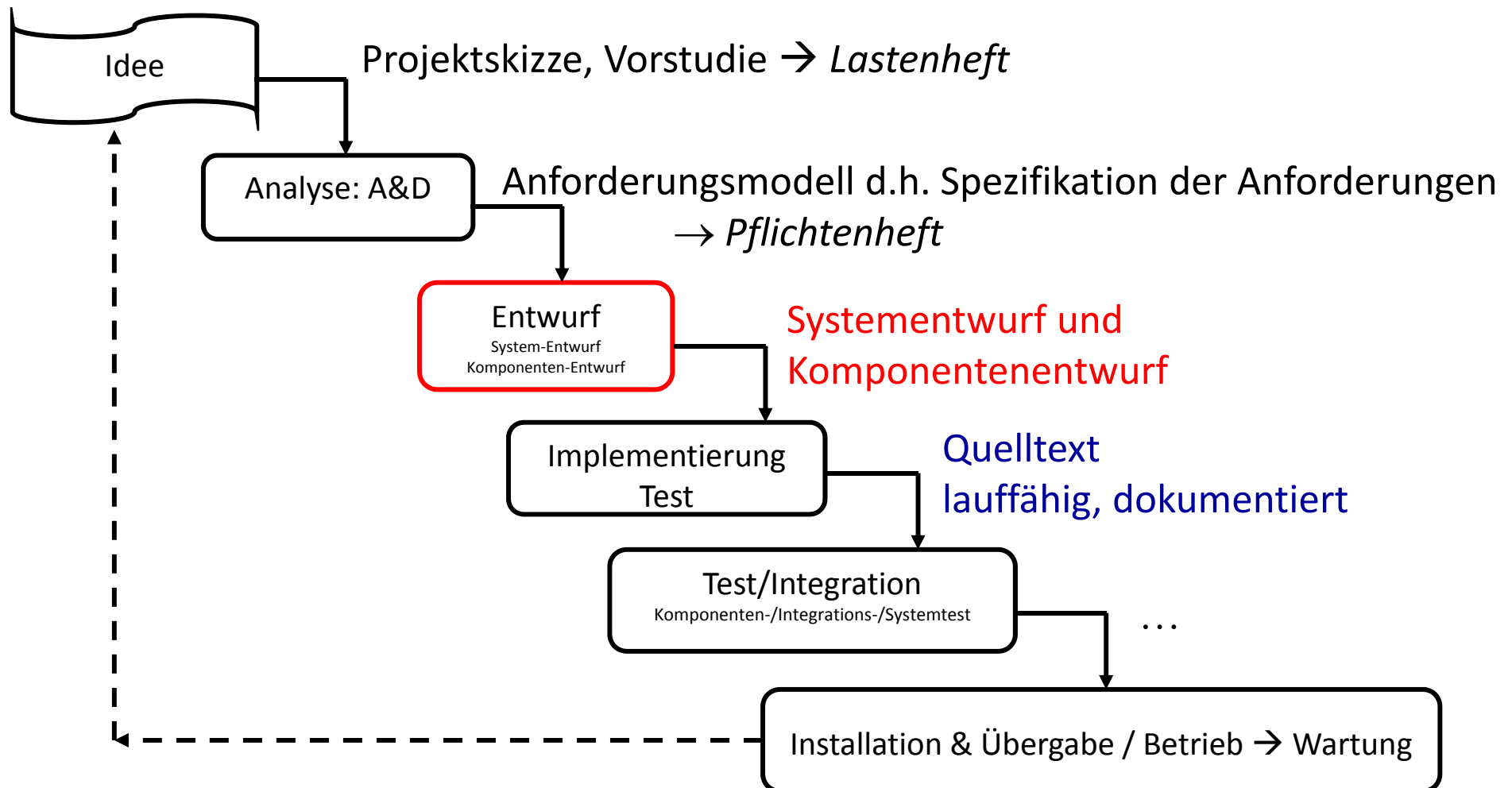
Fehler im Entwurf bleiben uns oft über Jahrzehnte erhalten.“

Ludewig/Lichter „Software Engineering“, 3. Auflage 2013, S. 399

Welche Strukturen? Welche Komponenten? Welche Beziehungen?  
Wie lässt sich die hohe Komplexität beherrschen?



# SW-Lebenszyklus

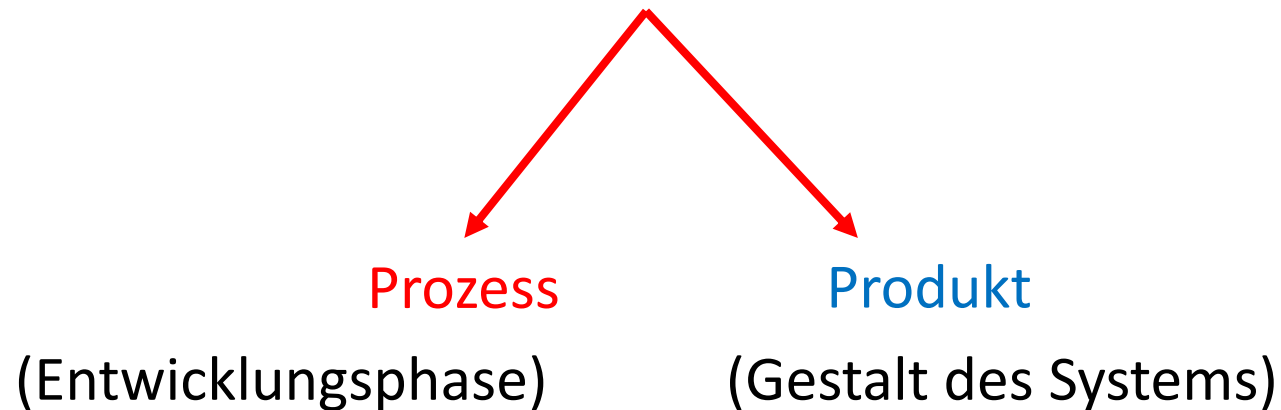


...

IEEE Standard 610.12 (1990)

Design –

- (1) **The process** of defining the architecture, components, interfaces, and other characteristics of a system or component
  - (2) **The result** of the process in (1).
- 




## Software-Architektur :

## Struktur

IEEE-Standard: 1471 (2000):

Architecture – The **fundamental** organization of a system embodied in this components, their relationships to each other and to environment, and the principles guiding its design and evolution.



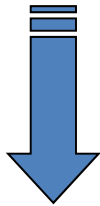
## Software-Architektur :

(Geleitwort von Ernst Denert

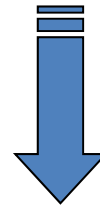
im Buch „Moderne SW-Architektur“ von Johannes Siedersleben)

„Das ist die Königsdisziplin des Software Engineering.

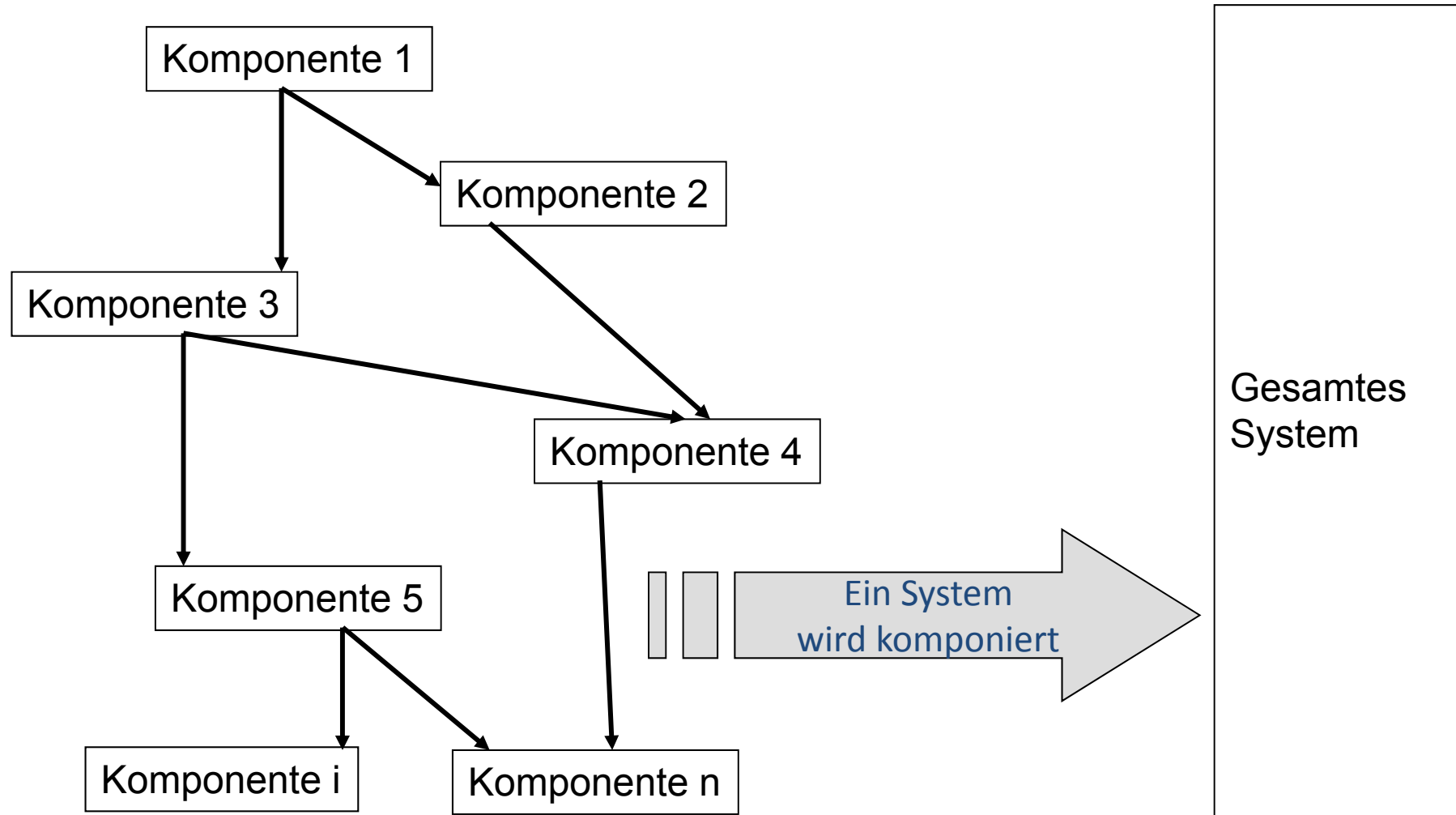
Sie befasst sich mit der Gestaltung von Software-Systemen, mit deren **Struktur im Kleinen und im Großen.**“

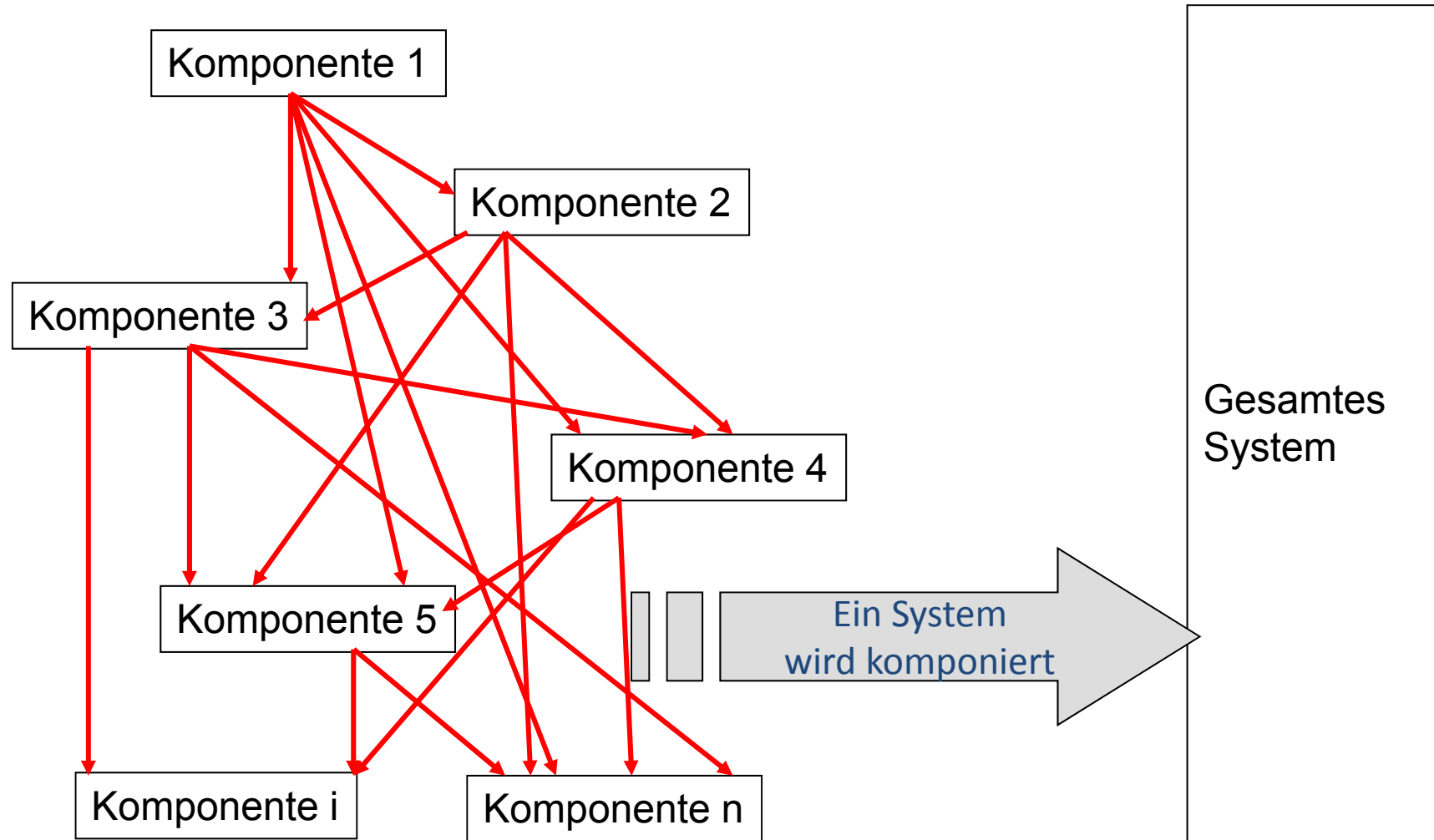


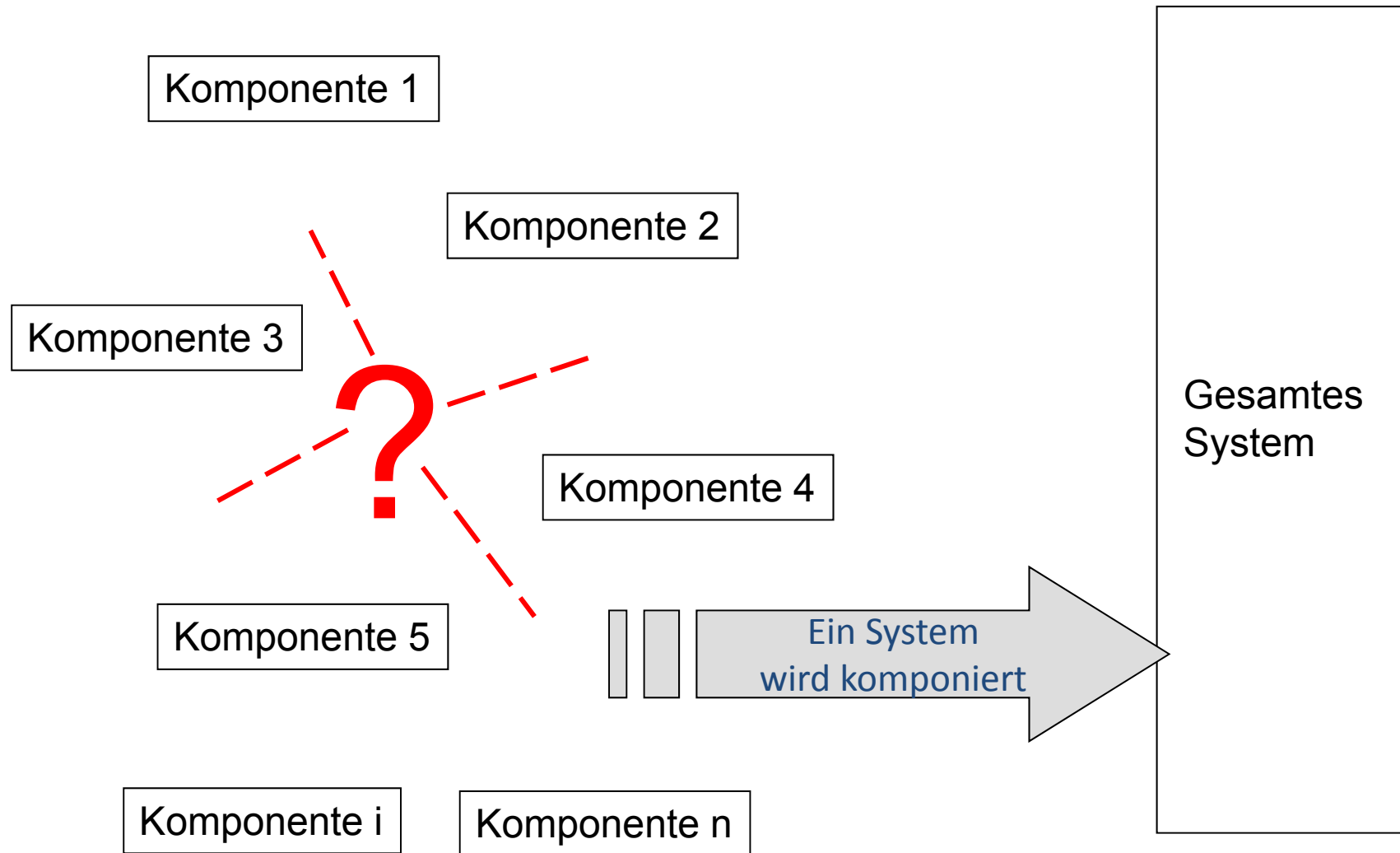
Feinentwurf



Grobentwurf







## Wie lässt sich die hohe Komplexität beherrschen?

KOMPLEXITÄT	ist essenziell	ist akzidentell
probleminhärent	Komplexität der Fachdomäne	Missverständnisse über die Fachdomäne
Lösungsabhängig	Komplexität der Technologie und der Architektur	Missverständnisse über die Technologie und überflüssige Lösungsanteile

essenziell: natürlich, der Aufgabenstellung innewohnend

akzidentell: künstlich hinzugefügt, nicht zur Aufgabenstellung bzw. zum Lösungsansatz gehörend  
d.h. unnötig



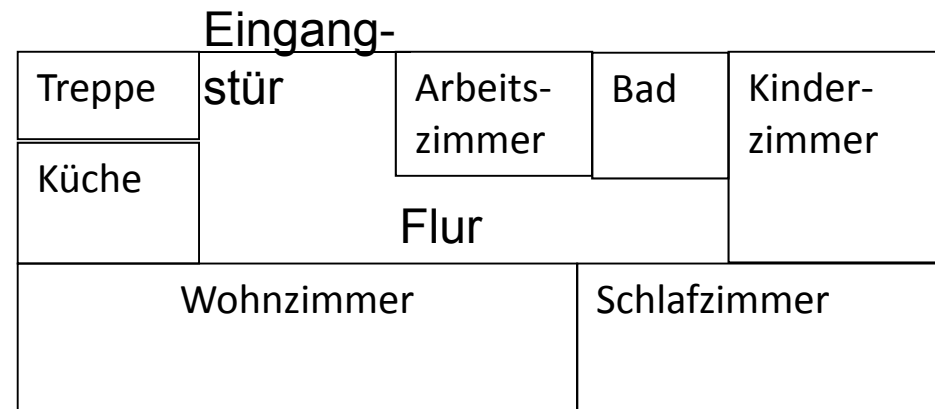
## Worin liegt also das Risiko beim SW-Entwurf?

---

- unnötige Erhöhung der natürlich gegebenen Komplexität
  - durch Missverständnisse der Aufgabenstellung (in der Analyse)
  - durch unnötige Lösungsanteile
  - durch technologische, technische Fehler
- Übertragung der Komplexität des Systems  
in die Komplexität der Schnittstellen zwischen Komponenten



## (Bau-)Architektur



Bernd Brügge, Allen H. Dutoit,  
Objektorientierte SW-Technik

Ein Vergleich (nach „Objektorientierte SW-Technik“ Bernd Brügge, Allen H. Dutoit)

	Gebäude	SW-System
<i>Komponenten</i>	Räume	Subsysteme, Module, Pakete, Klassen
<i>Schnittstelle</i>	Türen, Fenster, Schornstein, ....	Dienste (Export, Import)
<i>funktionale Anforderungen</i>	Wohnhaus, wohnen, schlafen, ...	essenzielle Fktn., AWF anmelden, suchen, ...
<i>nicht-funktionale Anforderungen</i>	Farbe der Wände, ...	Antwortzeiten, Speicherbedarf, ....
<i>nicht-kostspielige Änderungen (relativ)</i>	Inneneinrichtung,	Optimierung vorhandener Funktionalitäten, ...
<i>kostspielige Änderungen (relativ)</i>	Wände, Fenster, Türen versetzen	Schnittstellen ändern



Nach der Analysephase sind also folgende Fragen bei der Software-Entwicklung zu klären:

- Worin besteht das Ziel der Entwurfsphase?
- Welches Risiko existiert beim „Komponieren“ eines SW-Systems?
- Was eigentlich ist eine Komponente bzw. eine Schnittstelle im Kontext der SW-Entwicklung?
- Welche Entwurfsprinzipien gibt es?
  - Worin liegen ihre Stärken?
  - Welche Konsequenzen ergeben sich aus ihrer Anwendung?
  - Was bedeutet „Vererbung“ aus der Perspektive des SW-Entwurfes?
- Welche grundlegenden SW-Architekturen gibt es?
- Welche grundlegenden SW-Entwurfsmuster (Design Pattern) gibt es?

Heute sind erste Fragen bereits beantwortet und die Begriffe „Entwurf“ bzw. „Architektur“ definiert:

---

### Worin besteht das Ziel der Entwurfsphase?

→ Struktur des künftigen Software-Systems festlegen.

D.h. Bestandteile (= Komponenten) und deren Beziehungen zueinander definieren  
und Beziehungen zum Kontext des Systems definieren.

### Was bedeutet „Software-Entwurf“?

„Software-Entwurf“ = Entwicklungsphase (Prozess) und gleichermaßen  
= Struktur eines Software-Systems (Produkt).

Die statische Struktur ist die Basis für den dynamischen Prozess, den das Software-System ermöglicht.

Aufgrund der Komplexität wird das Software-System auf hoher Abstraktionsebene (Grobentwurf) und auf detaillierter Ebene (Feinentwurf) beschrieben.

### Was bedeutet „Software-Architektur“?

„Software-Entwurf“ = grundlegende Struktur des Software-Systems,  
d.h. seine grundlegende Bestandteile und ihre Beziehungen zueinander  
auf hoher Abstraktionsebene  
(z.B. 3-Schichten-Architektur, Client-Server-Architektur, ...)

### Welches Risiko existiert beim „Komponieren“ eines SW-Systems?

Übertragung der Komplexität des Systems in die Komplexität der Schnittstellen zwischen den Komponenten

→ U.U. entstehen starre Software-Systeme, die sich nur schwer erweitern oder verändern lassen („Monolith“).