Vorlesung Softwaretechnik I (SS 2024)

8. Entwurf

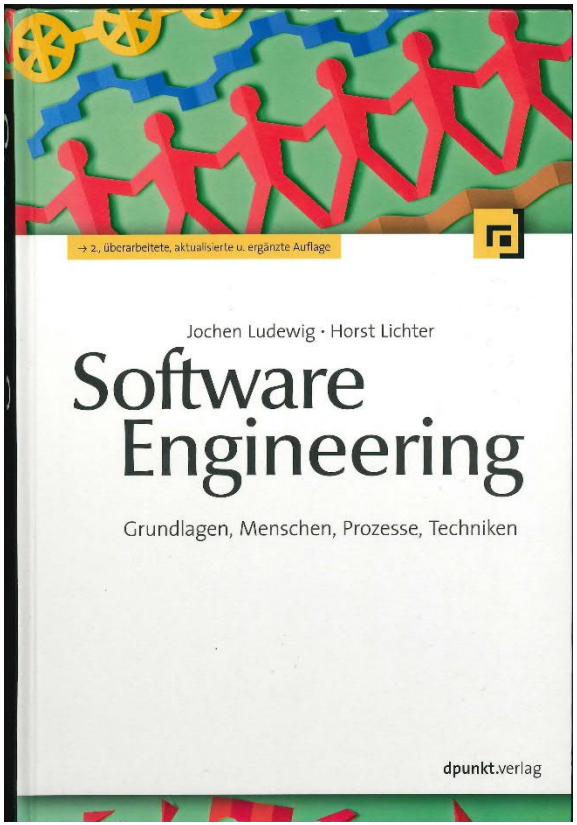
Prof. Dr. Jens Grabowski

Tel. 39 172022

grabowski@informatik.uni-goettingen.de

Der Inhalt dieses Kapitels orientiert sich eng an:

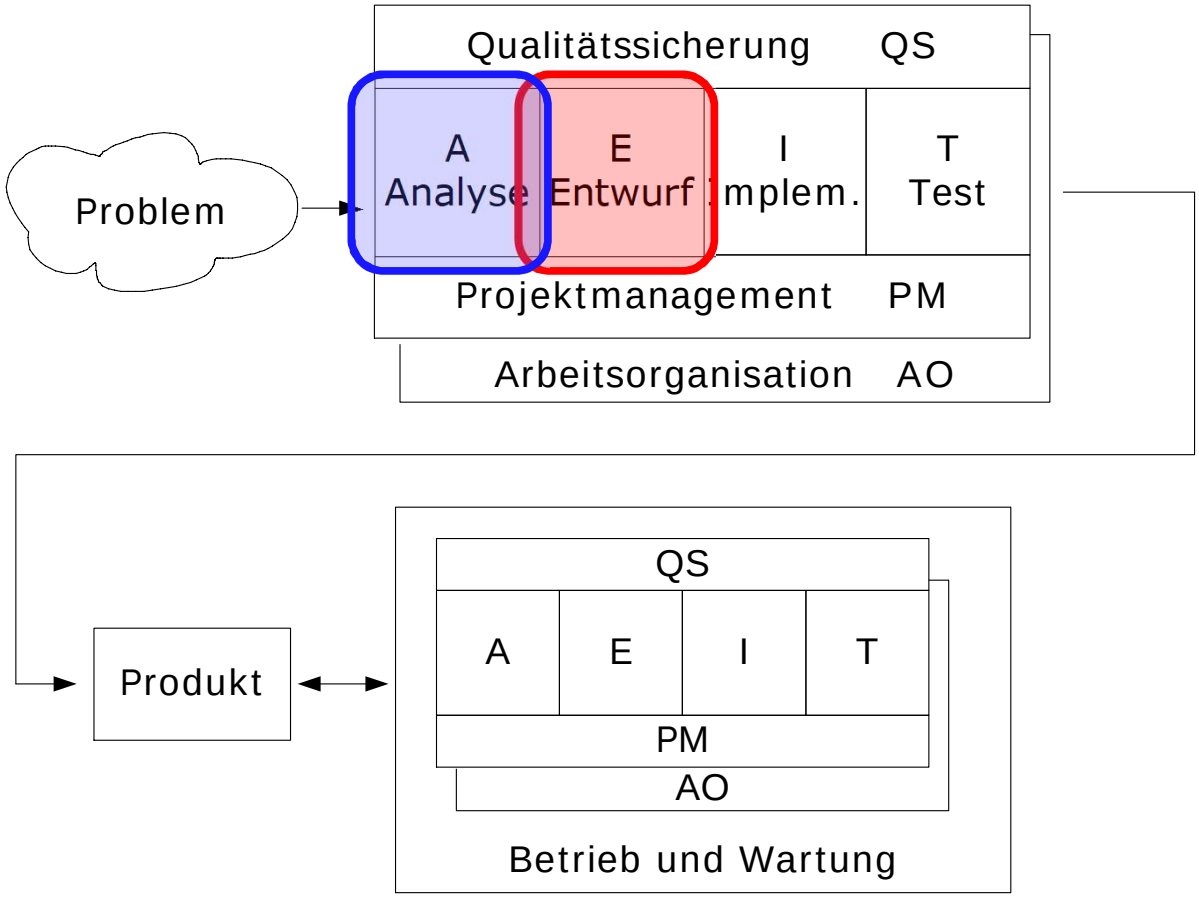
Titel: Software Engineering
Autoren: J. Ludewig, H. Lichter
Verlag: dpunkt.verlag
ISBN: 3-89864-268-2



Inhalt

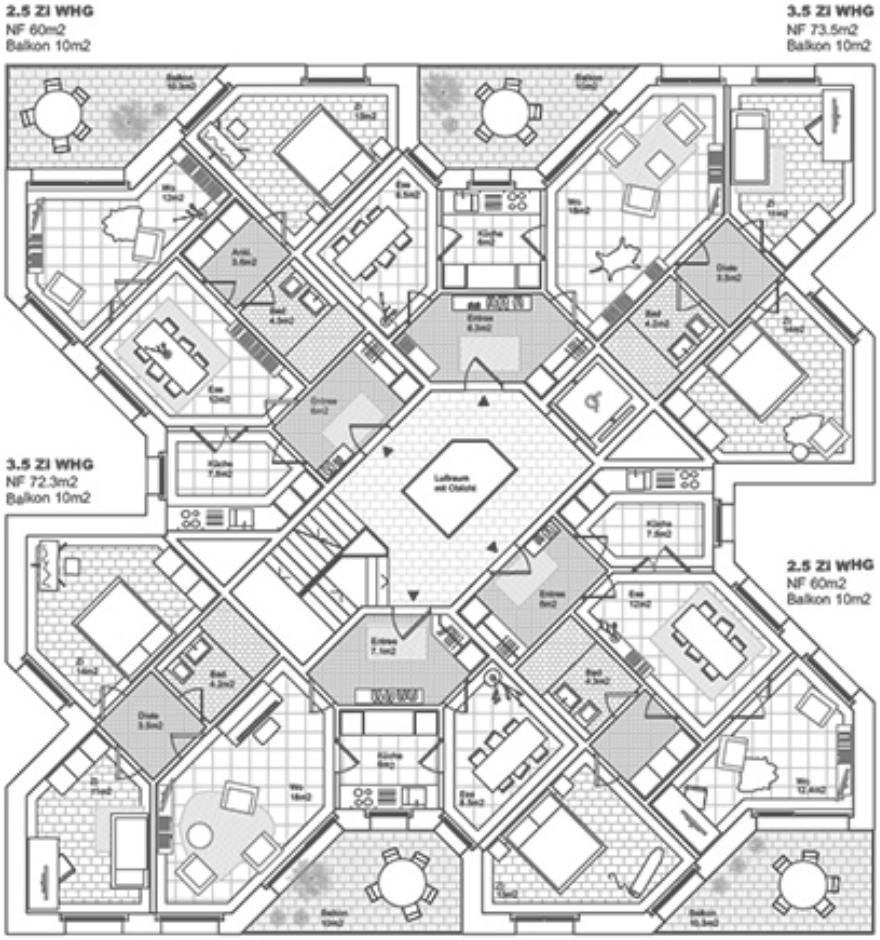
- Ziele des Architekturentwurfs
- Begriffe
- Prinzipien des Architekturentwurfs
- Architekturmuster
- Entwurfsmuster
- Qualität von Softwarearchitekturen
- Lernziele

Worum geht es in diesem Kapitel?



Es geht um …

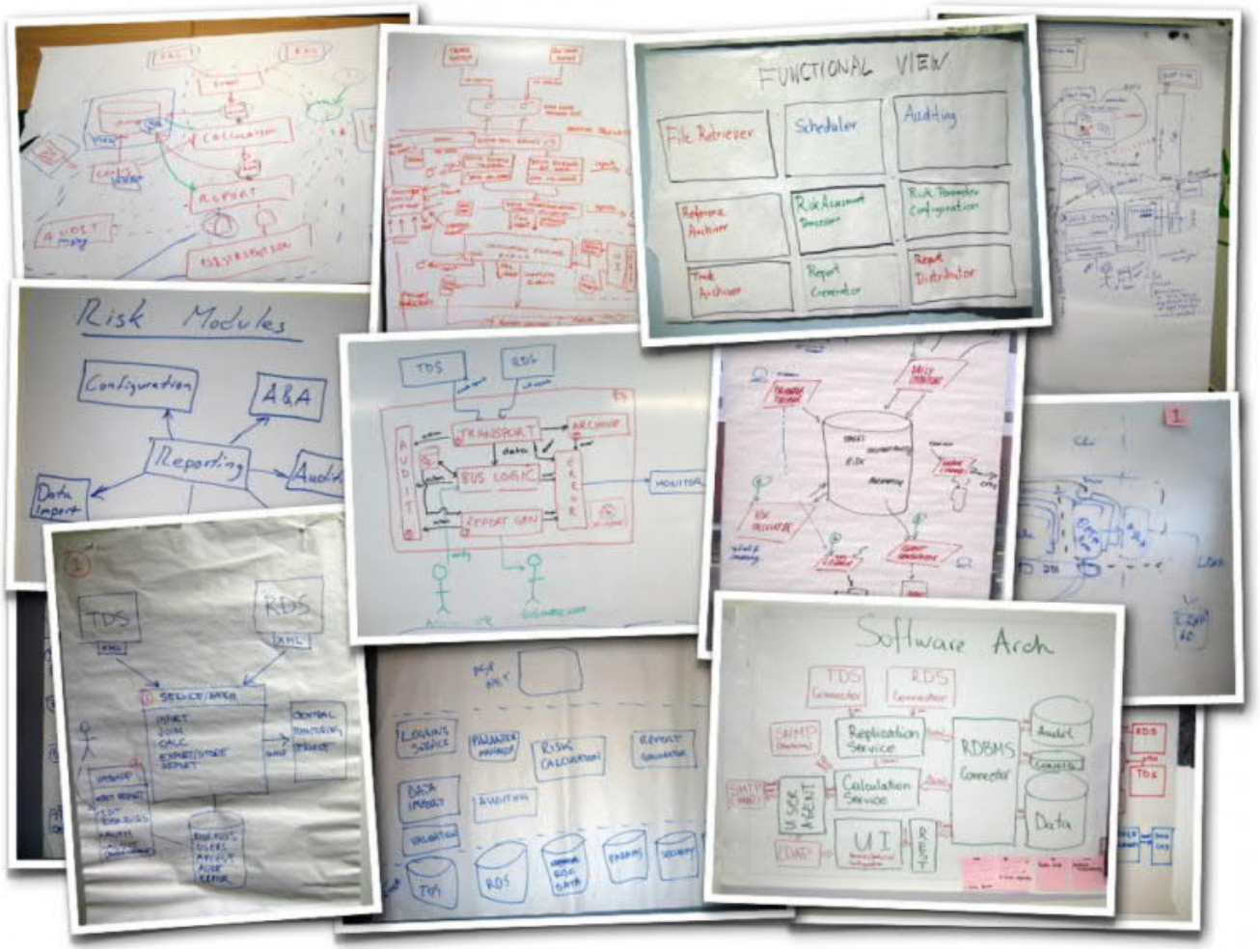
- Architektur …



Grundriss Haus D, Nord
Mst. 1:100

[www.abrahachermann.com/index.php?/contact/058frohburg/]

… um Softwarearchitektur!



[www.voxxed.com]

Ziele des Architekturentwurfs

- Gliederung des Systems in überschaubare (handhabare) Einheiten.
- Festlegung der Lösungsstruktur.
- Hierarchische Gliederung.

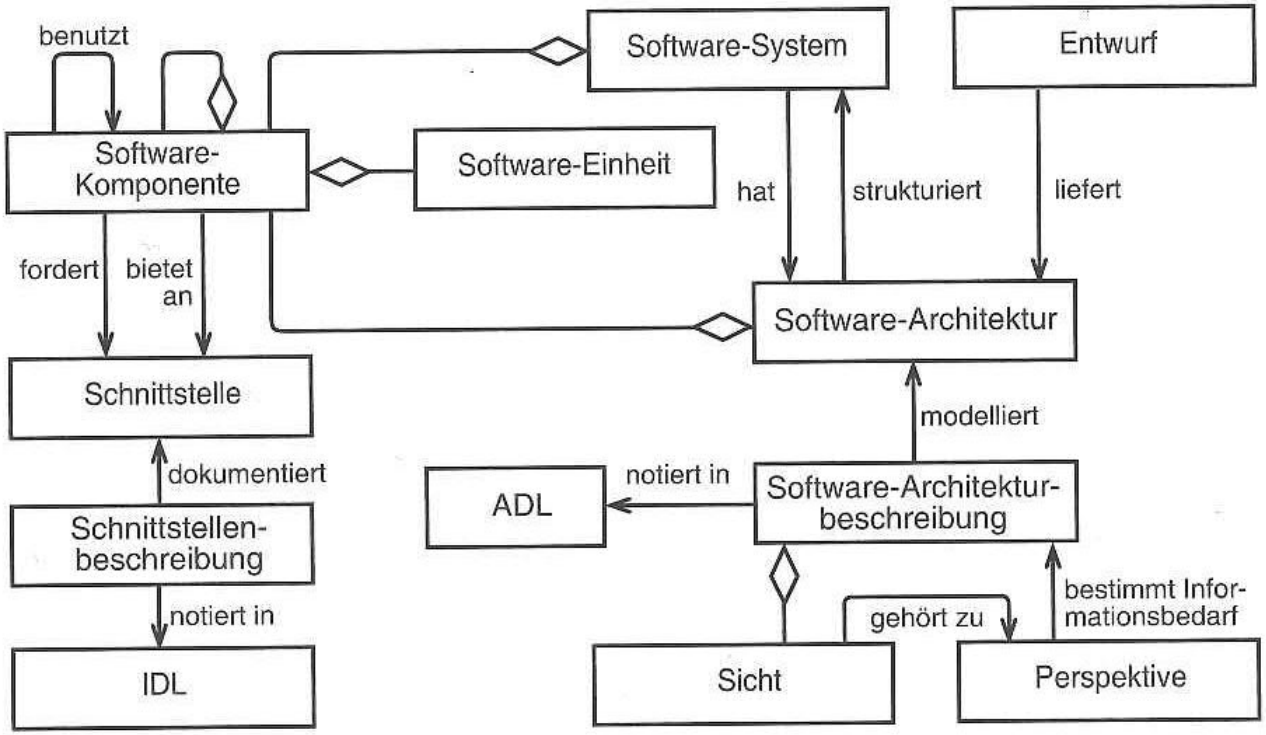
  - Meist wird zwischen Grob- und Feinentwurf unterschieden.

Grobentwurf (Gesamtstruktur des Systems)	• Architekturentwurf • Subsystem-Spezifikation • Schnittstellen-Spezifikation
Feinentwurf (Detailstruktur des Systems)	• Komponentenentwurf • Datenstrukturentwurf • Algorithmenentwurf

Inhalt

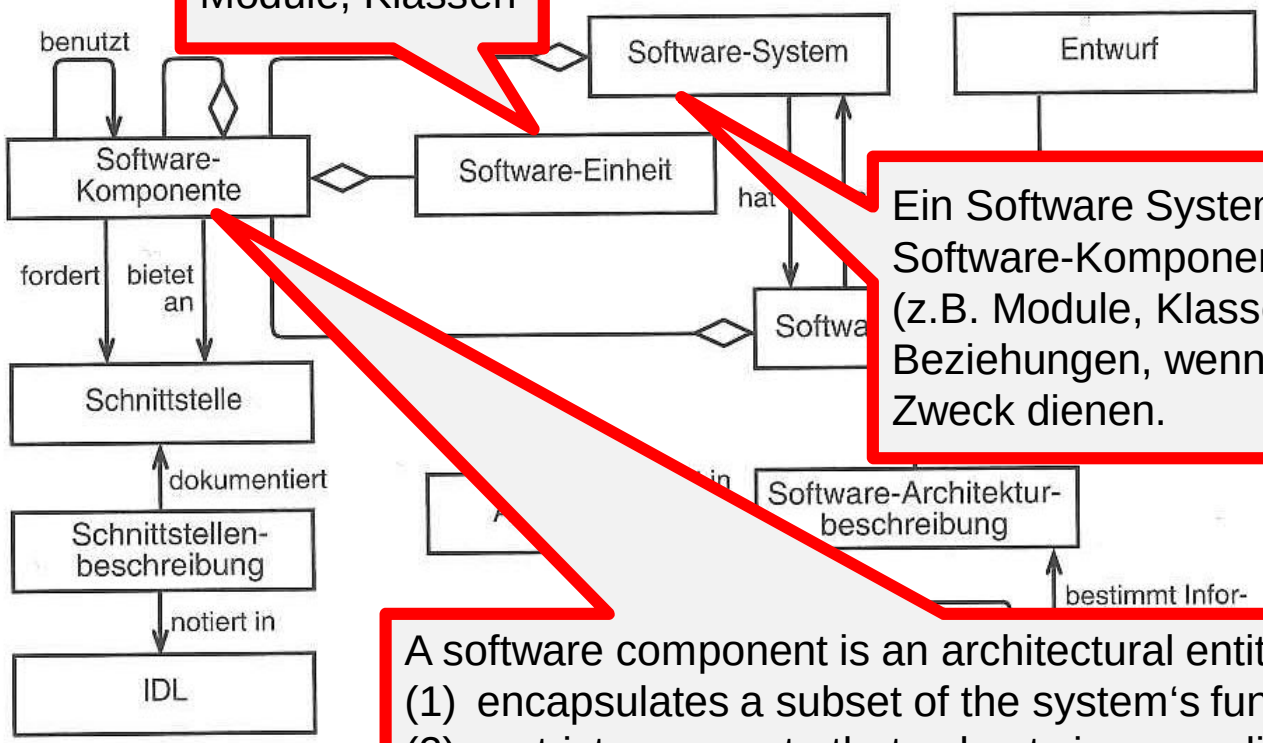
- Ziele des Architekturentwurfs
- **Begriffe**
- Prinzipien des Architekturentwurfs
- Architekturmuster
- Entwurfsmuster
- Qualität von Softwarearchitekturen
- Lernziele

Begriffe



Begriffe

Module, Klassen

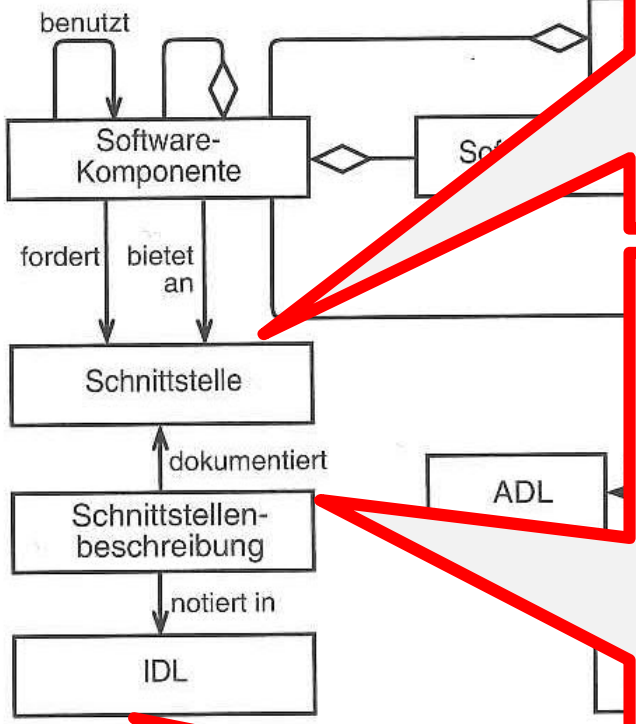


Ein Software System ist eine Menge von Software-Komponenten und -Einheiten (z.B. Module, Klassen) und ihren Beziehungen, wenn Sie einem bestimmten Zweck dienen. (Ludewig, Lichter, 2012)

A software component is an architectural entity that
(1) encapsulates a subset of the system‘s functionality and/or data,
(2) restricts access to that subset via an explicitly defined interface, and
(3) has explicitly defined dependencies on its required execution context.

(Taylor, Medvidovic, Dashofy, 2010)

Begriffe



Schnittstelle – Die Grenze zwischen zwei kommunizierenden Komponenten. Die Schnittstelle einer Komponente stellt die Leistungen der Komponente für ihre Umgebung zur Verfügung und/oder fordert Leistungen, die sie aus der Umgebung benötigt.

(Ludewig, Lichter, 2012)

Enthält Angaben zu:

- *Syntax und Art der Kommunikation* (z.B. Name, Reihenfolge, Typ von Parametern).
- *Funktionale Merkmale* (z.B. angebotene/benötigte Dienste, Vor- und Nachbedingungen, Verhalten im Fehlerfall).
- *Allgemeine Qualitätsangaben* (z.B. Antwortzeiten, Genauigkeit von Resultaten).

Z.B.:

- CORBA IDL
- WSDL
- JAVA Interfaces

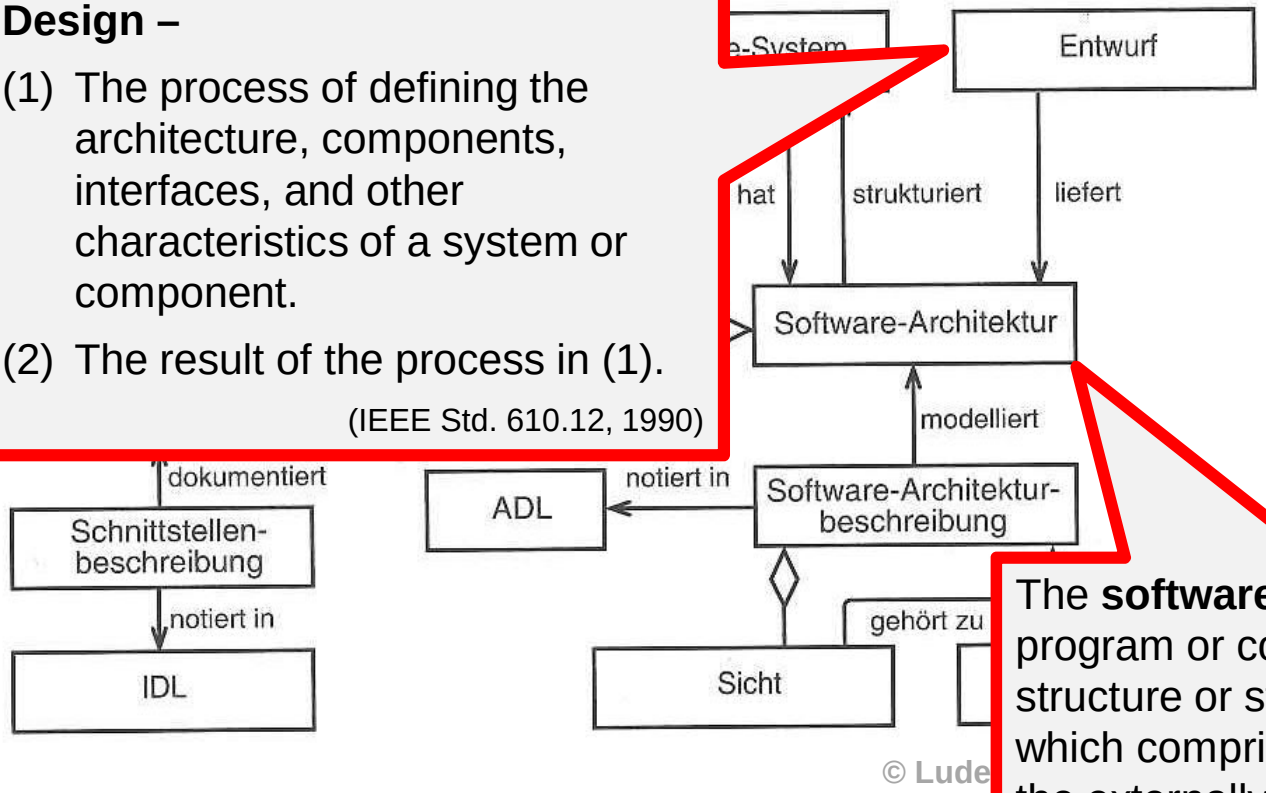
Begriffe

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).

(IEEE Std. 610.12, 1990)



The **software architecture** of a program or computing system is the structure or structures of the system which comprise software elements, the externally visible properties of those elements, and the relationships among them.

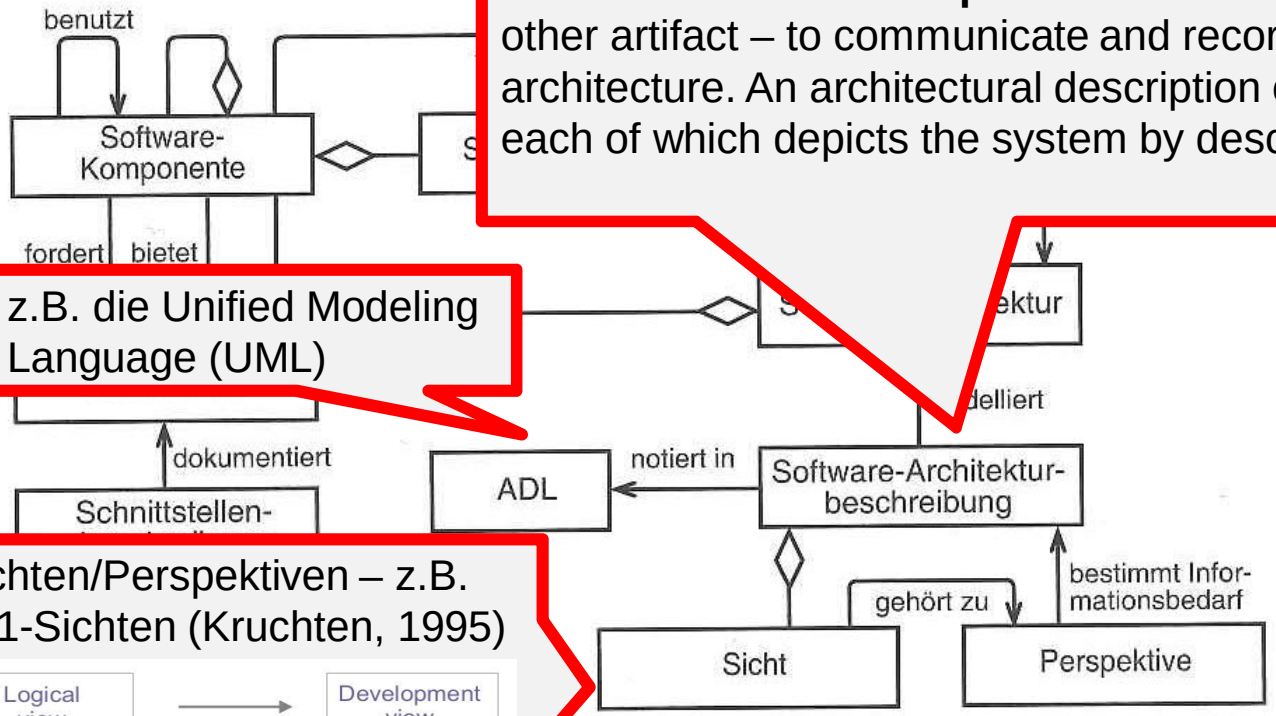
(Bass, Clements, Kazman, 2003)

Begriffe

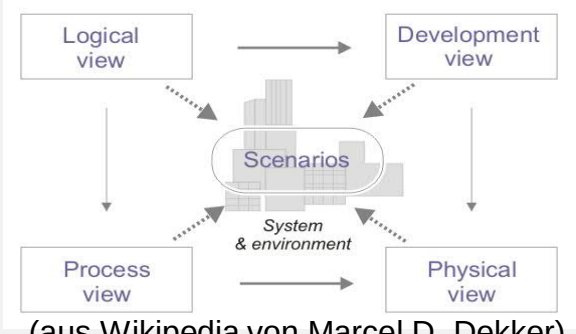
An **architectural description** is a model – document, product or other artifact – to communicate and record a system‘s architecture. An architectural description conveys a set of views each of which depicts the system by describing domain concerns.

(Ellis et al., 1996)

z.B. die Unified Modeling Language (UML)



Sichten/Perspektiven – z.B. 4+1-Sichten (Kruchten, 1995)

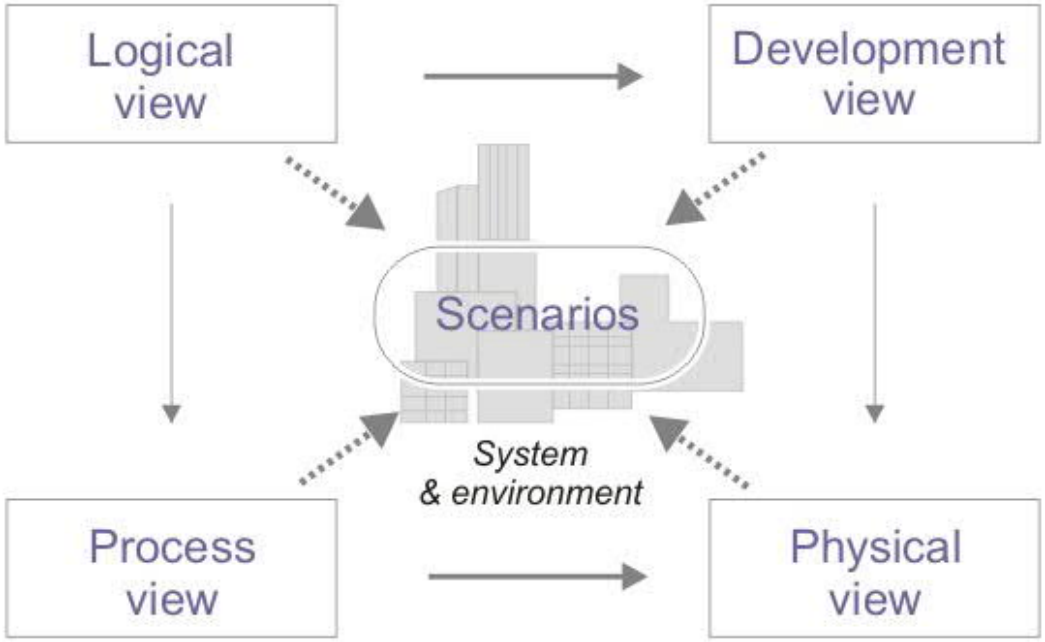


(aus Wikipedia von Marcel D. Dekker)

Begriffe

Benutzersicht:
1. Einbettung des Systems in die Umgebung.
2. Systemgrenzen, Schnittstellen.

Entwicklersicht:
1. Zentrale Komponenten der Architektur,
2. ihre Schnittstellen und
3. Beziehungen.

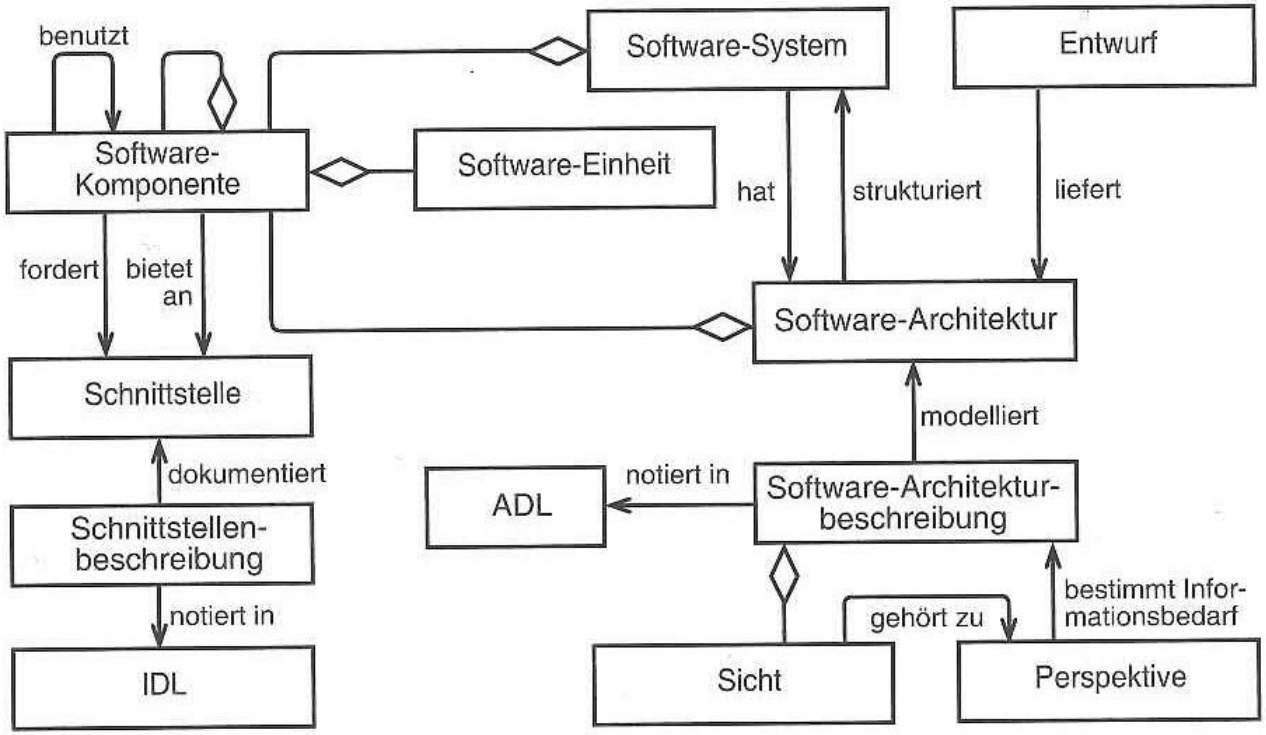


(aus Wikipedia von Marcel D. Dekker)

Integratorensicht:
1. Kontrollfluss,
2. zeitliche Abhängigkeiten.

Verteilungssicht:
Abbildung der Komponenten auf Infrastruktur- und Hardware-Einheiten.

Begriffe – Fragen?



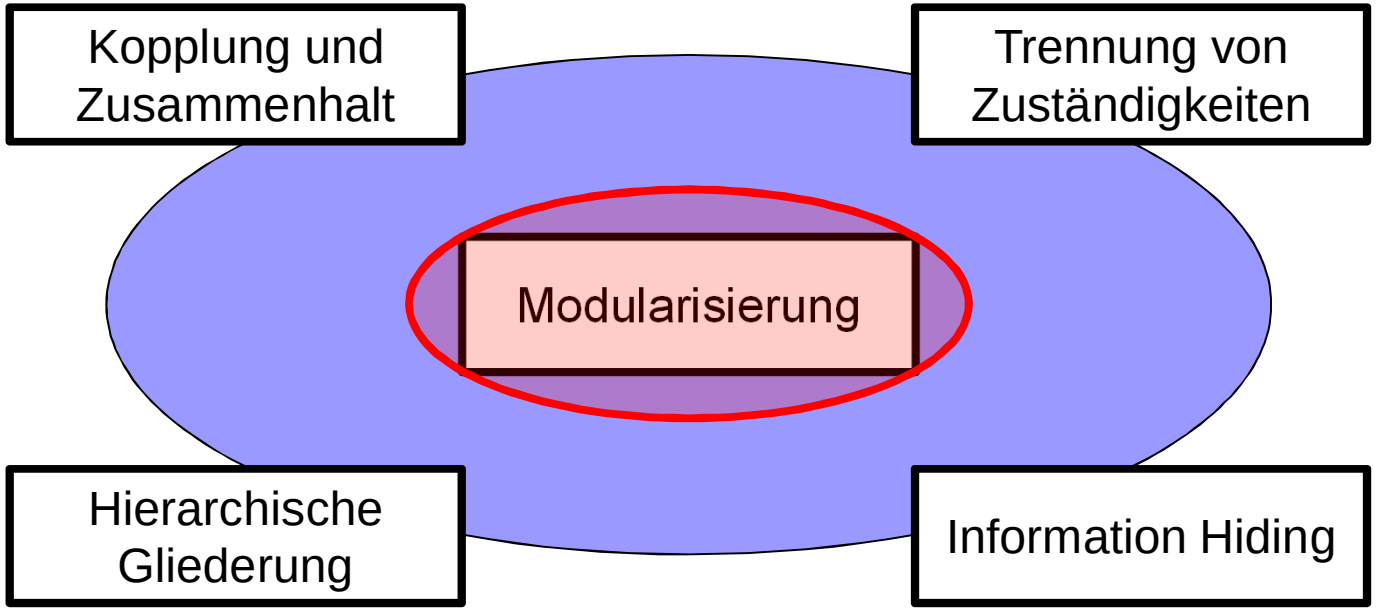
Inhalt

- Ziele des Architekturentwurfs
- Begriffe
- **Prinzipien des Architekturentwurfs**
- Architekturmuster
- Entwurfsmuster
- Qualität von Softwarearchitekturen
- Lernziele

Prinzipien des Architekturentwurfs

Wie sollen die Komponenten miteinander vernetzt sein?

Welche Eigenschaften müssen zusammengefasst werden?



Wie sollen die Komponenten verfeinert und strukturiert werden?

Was muss versteckt, was an der Schnittstelle angeboten werden?

Prinzipien des Architekturentwurfs - Modularisierung

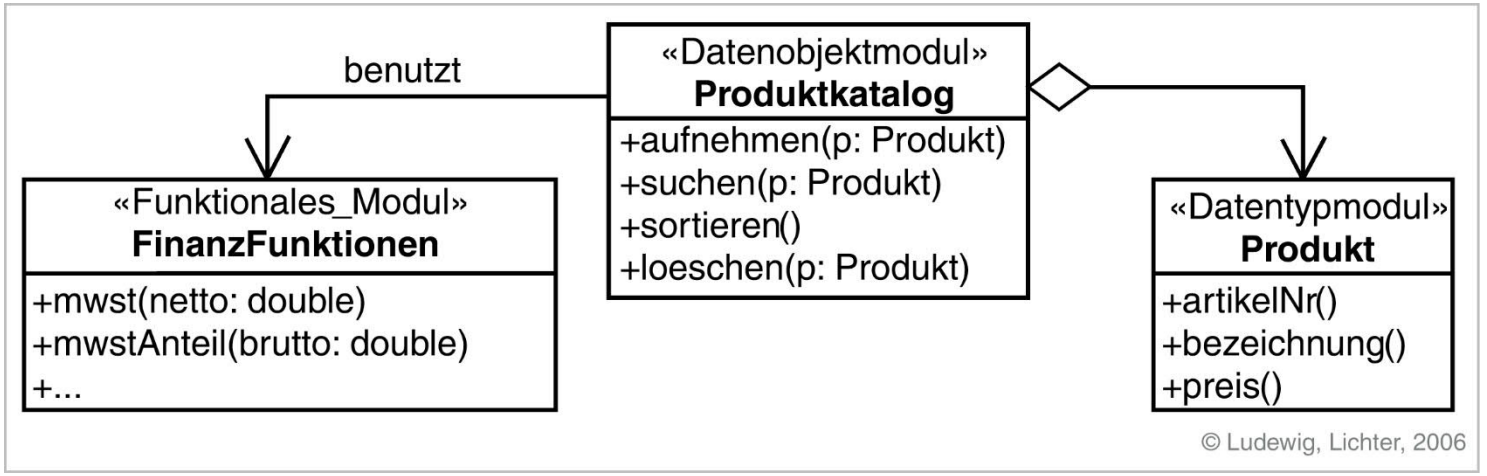
- Aufteilung des Systems in seine Komponenten.

- Definitionen
  - **Modular decomposition**
    - The process of breaking a system into components to facilitate design and development; an element of modular programming.
      - (IEEE Std. 610.12, 1990)
  - **Modularity**
    - The degree to which a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components.
      - (IEEE Std. 610.12, 1990)

Prinzipien des Architekturentwurfs - Modularisierung

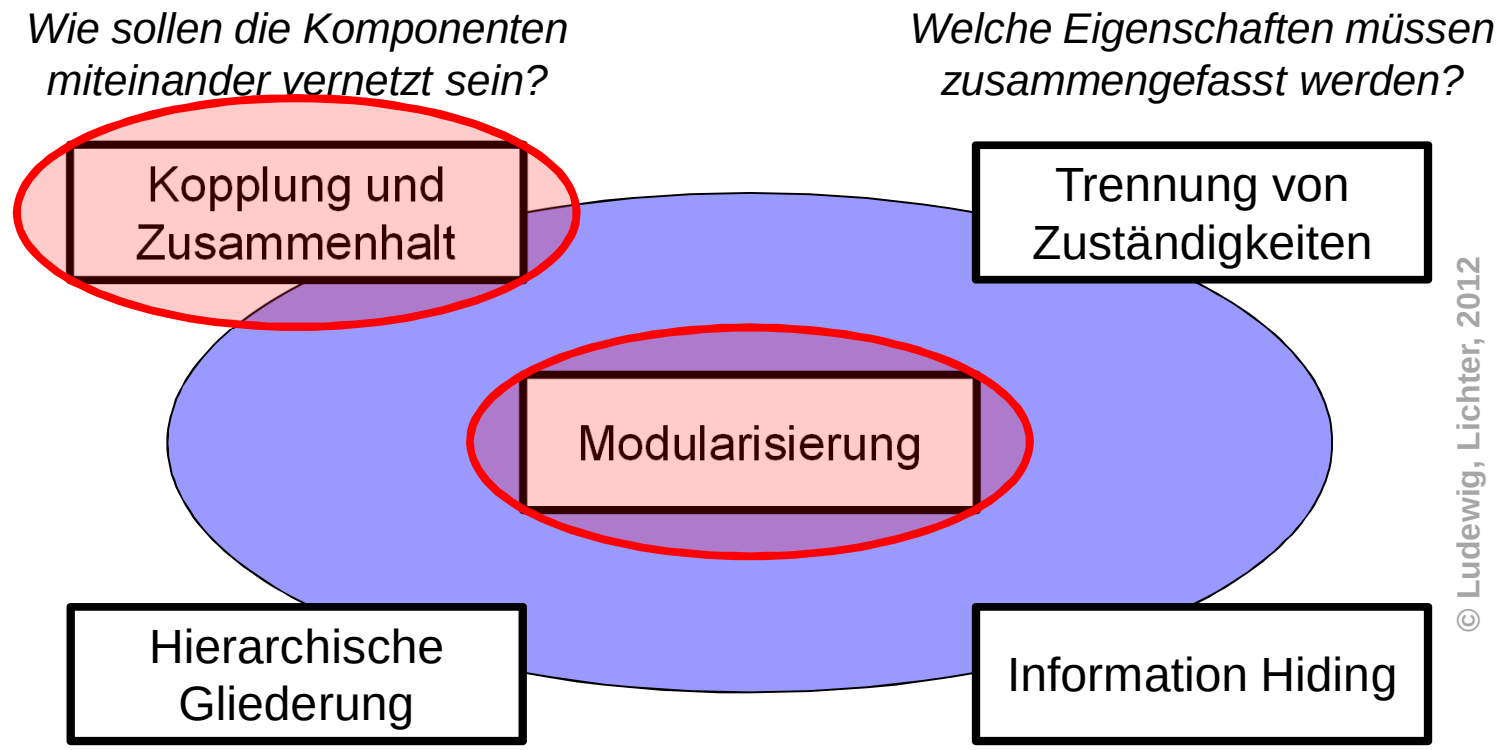
- Ziele eines modularen Entwurfs (Parnas, 1972):
  - Die Struktur jedes Moduls soll einfach und leicht verständlich sein.
  - Die Implementierung eines Moduls soll austauschbar sein; Informationen über die Implementierung der anderen Module sind dazu nicht erforderlich. Die anderen Module sind vom Austausch nicht betroffen.
  - Die Module sollen so entworfen sein, dass wahrscheinliche Änderungen ohne Modifikation der Modulschnittstellen durchgeführt werden können.
  - Große Änderungen sollen sich durch eine Reihe kleiner Änderungen realisieren lassen. Solange die Schnittstellen der Module nicht verändert sind, soll es möglich sein, alte und neue Modulversionen miteinander zu testen.

Prinzipien des Architekturentwurfs - Modularisierung



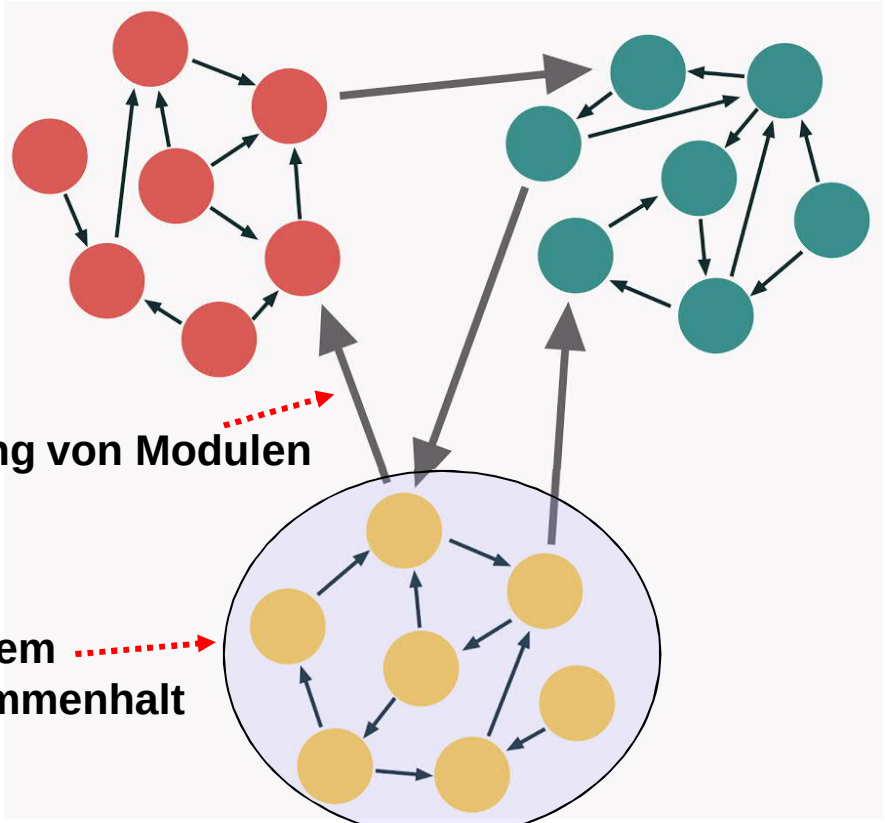
- Mögliche Modularten, z.B. nach (Nagl, 1990):
  - **Funktionale Module**
    - Gruppieren Berechnungen, kein Gedächtnis.
  - **Datenobjektmodule**
    - Realisieren das Konzept der Datenkapselung.
    - Schnittstelle bilden Operationen zur Manipulation der gekapselten Daten.
  - **Datentypmodule**
    - Implementieren benutzerdefinierte (abstrakte) Datentypen.

Prinzipien des Architekturentwurfs



Prinzipien des Architekturentwurfs - Kopplung und Zusammenhalt

- Ziele:
  - Geringe inter-modulare Kopplung
  - Hoher intra-modularer Zusammenhalt



[http://enterprisecraftsmanship.com]

Prinzipien des Architekturentwurfs - Kopplung und Zusammenhalt

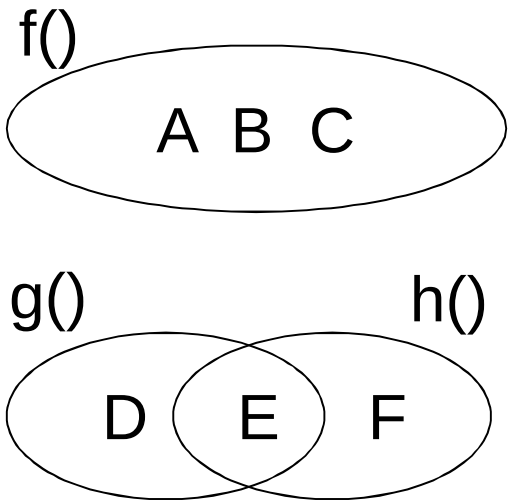
- Kohäsion (cohesion)
  - Maß für die Stärke des inneren Zusammenhang (innerhalb) eines Moduls.
  - Je höher die Kohäsion desto besser die Modularisierung.
- Kopplung (coupling)
  - Ein Maß für den Zusammenhang zwischen zwei Komponenten.
  - Je geringer die wechselseitige Kopplung zwischen den Komponenten, desto besser die Modularisierung.

Prinzipien des Architekturentwurfs - Kopplung und Zusammenhalt

- Metriken (Chidamber & Kemerer, 1993)
  - CBO (Coupling Between Object Classes)
    - Eine Klasse ist mit einer anderen Klasse gekoppelt, wenn sie deren Methoden und/oder Attribute benutzt oder direkt von ihr erbt. CBO gibt die Anzahl der an eine Klasse gekoppelten Klassen an.
  - LCOM (Lack of Cohesion in Methods)
    - Die Anzahl der Paare von Methoden der Klasse ohne gemeinsame Instanzvariablen minus Anzahl der Paare von Methoden dieser Klasse mit gemeinsamen Instanzvariablen.

Prinzipien des Architekturentwurfs - Kopplung und Zusammenhalt

- LCOM Beispiel:



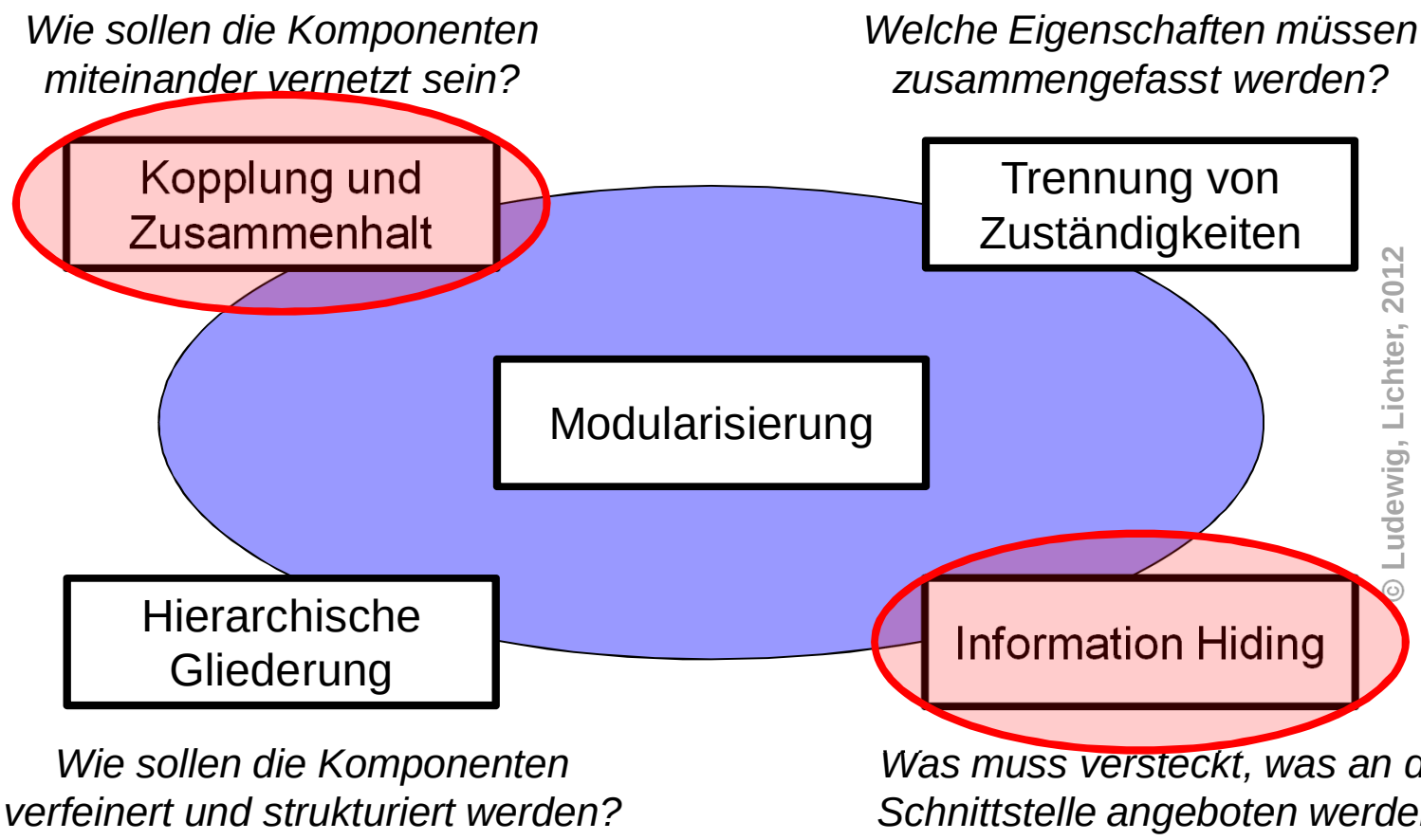

```
class X {
  int A, B, C, D, E, F;
  void f() { … uses A, B, C … }
  void g() { … uses D, E … }
  void h() { … uses E, F … }
}
```

Paarbildung

(f, g), (f, h)	keine gemeinsamen Variablen
(g, h)	gemeinsame Variable

LCOM = 2 – 1 = 1

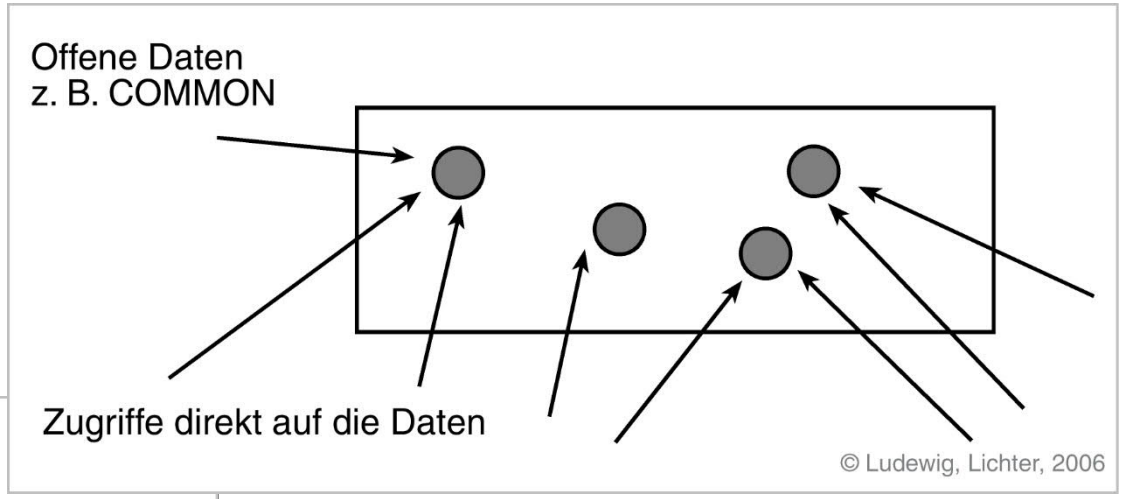
Prinzipien des Architekturentwurfs

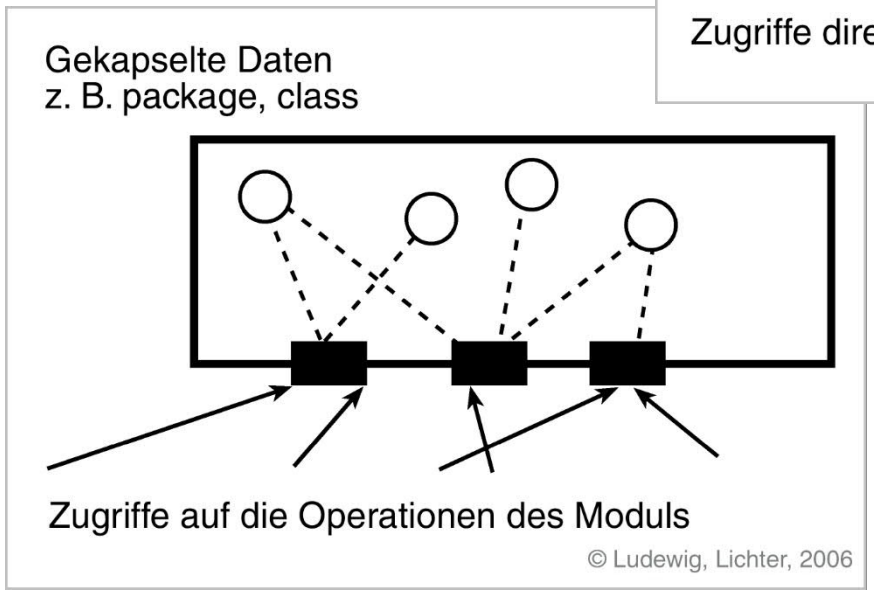


Prinzipien des Architekturentwurfs - Information Hiding

- Def. Information hiding
  - A software development technique in which each module‘s interfaces reveal as little as possible about the module‘s inner workings, and other modules are prevented from using information about the module that is not in the module‘s interface specification.
    - (IEEE Std. 610.12, 1990)

Prinzipien des Architekturentwurfs - Information Hiding



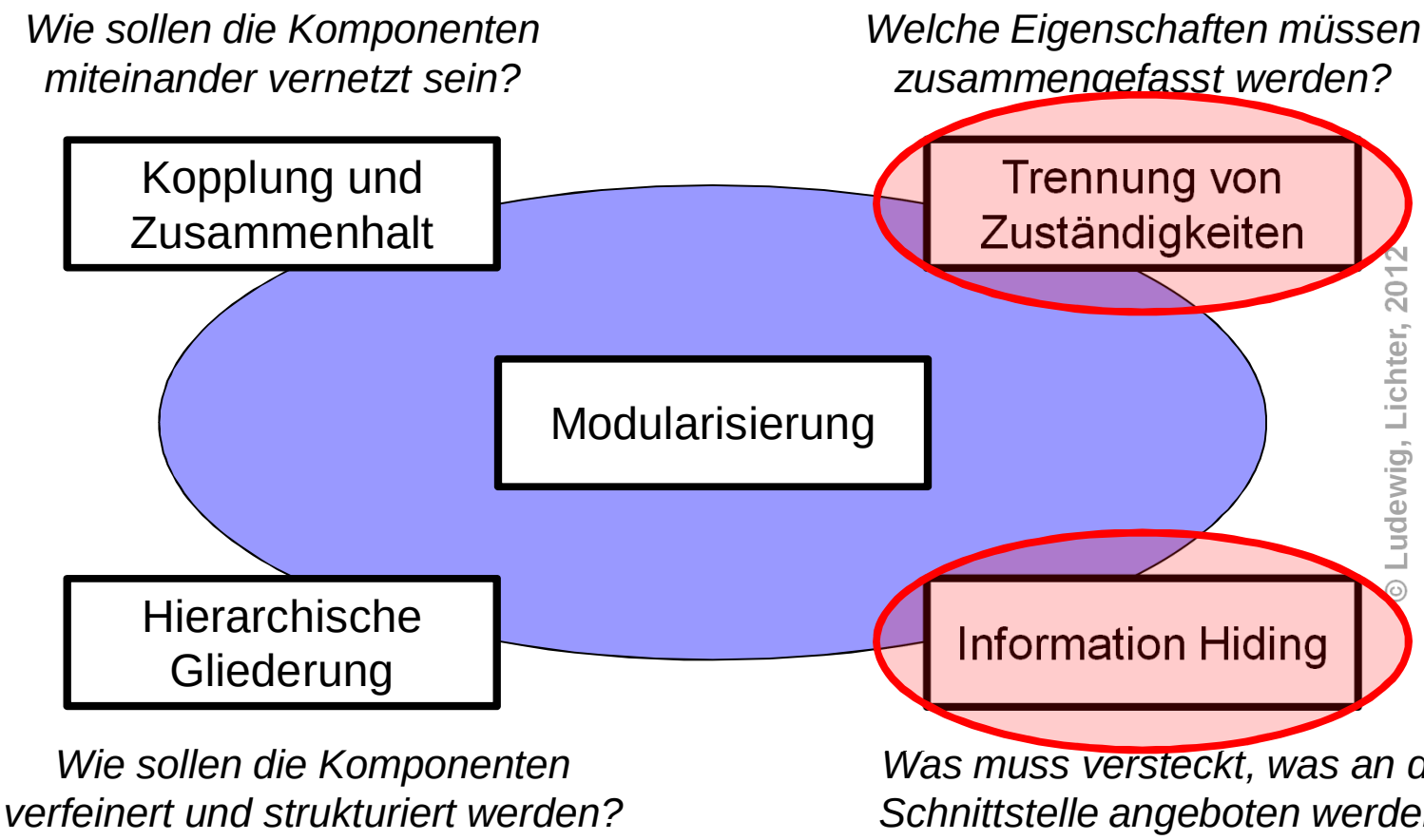


Prinzipien des Architekturentwurfs - Information Hiding

- **Vorteile**
  - Lokalität des Programms verbessert
    - Gefahr unbeabsichtigter Fehlmanipulationen gering.
    - Vereinfachte Änderbarkeit von Datenstrukturen.
  - Datenzugriffe können auf hohem Abstraktionsniveau (Funktionsaufrufe) überwacht werden.

- **Nachteile**
  - Methodenaufruf kostet Rechenzeit. Hier können optimierende Compiler helfen.
  - Konsequentes Information Hiding führt zu vielen Modulen/Klassen, wodurch die Verständlichkeit leidet.

Prinzipien des Architekturentwurfs



Prinzipien des Architekturentwurfs - Trennung von Zuständigkeiten

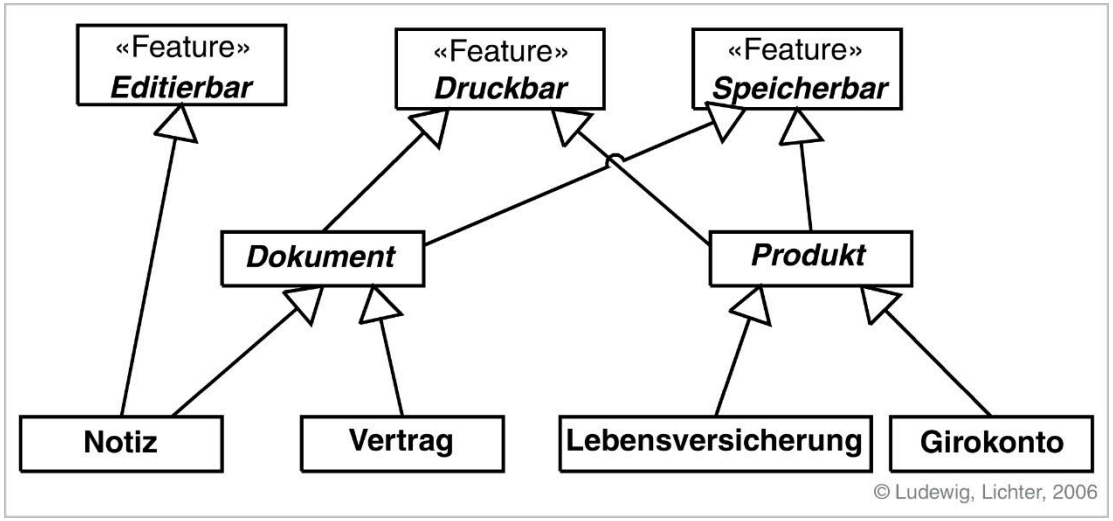
(engl. „separation of concerns“)

- Prinzip
  - Jede Komponente sollte nur für einen ganz bestimmten Aufgabenbereich zuständig sein.
  - Komponenten die gleichzeitig mehrere Aufgaben wahrnehmen, sind häufig
    - unnötig komplex,
    - schwer verständlich,
    - schwierig wart- und weiterentwickelbar und
    - schlecht wiederverwendbar.

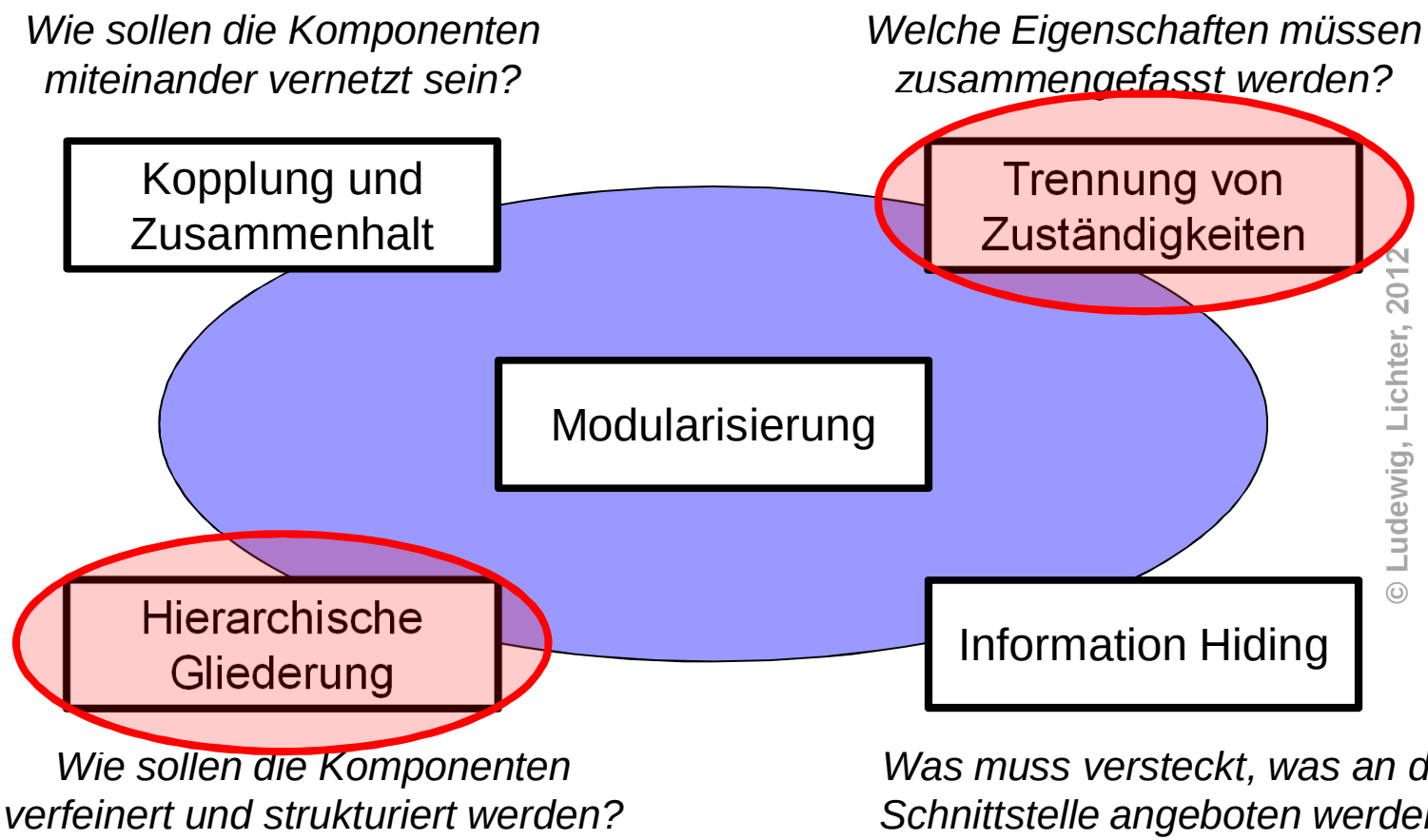
Prinzipien des Architekturentwurfs - Trennung von Zuständigkeiten

(engl. „separation of concerns“)

- Regeln zur Trennung von Zuständigkeiten
  - OO-Design: Zusammengehörige Daten und Operationen werden in einer Klasse zusammengefasst.
  - Trenne fachliche und technische Komponenten.
  - Ordne variable Funktionalitäten (auch solche, die später erweitert werden müssen) eigenen Komponenten zu.



Prinzipien des Architekturentwurfs



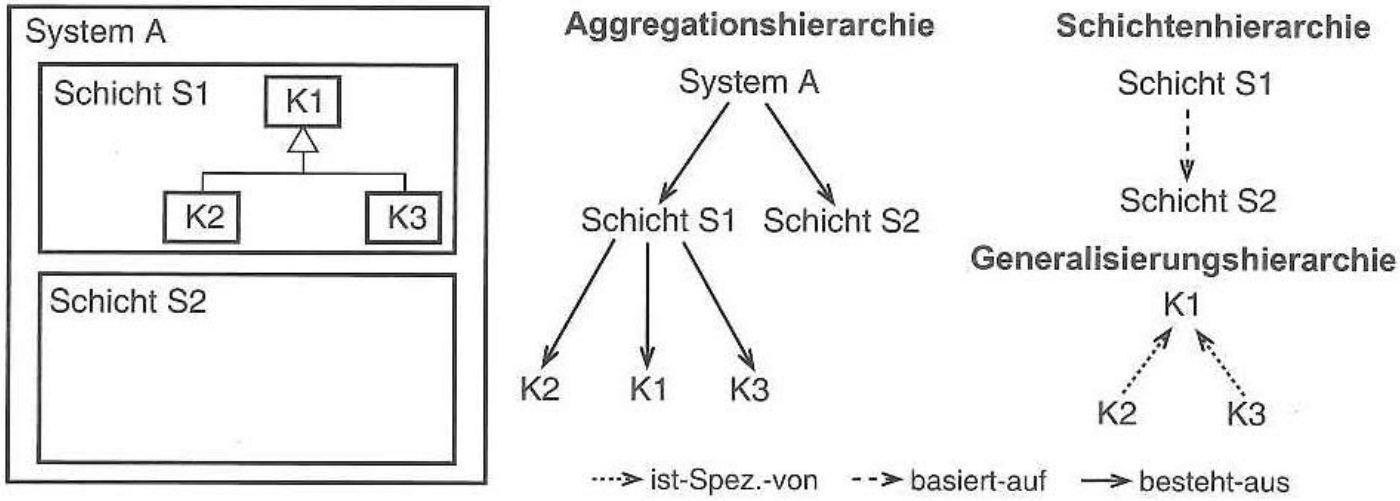
Prinzipien des Architekturentwurfs - Hierarchische Gliederung

- Hierarchische Gliederung ist ein bewährtes Vorgehen, um Komplexität zu reduzieren.

- Definition Hierarchie, Monohierarchie, Polyhierarchie:
  - Eine Hierarchie ist eine Struktur von Elementen, die durch eine (hierarchische) Beziehung in eine Rangfolge gebracht werden.
  - Monohierarchie:
    - Jedes Element, außer dem Wurzelelement, besitzt genau ein übergeordnetes Element.
  - Polyhierarchie:
    - Ein Element kann mehrere übergeordnete Elemente besitzen.

Prinzipien des Architekturentwurfs - Hierarchische Gliederung

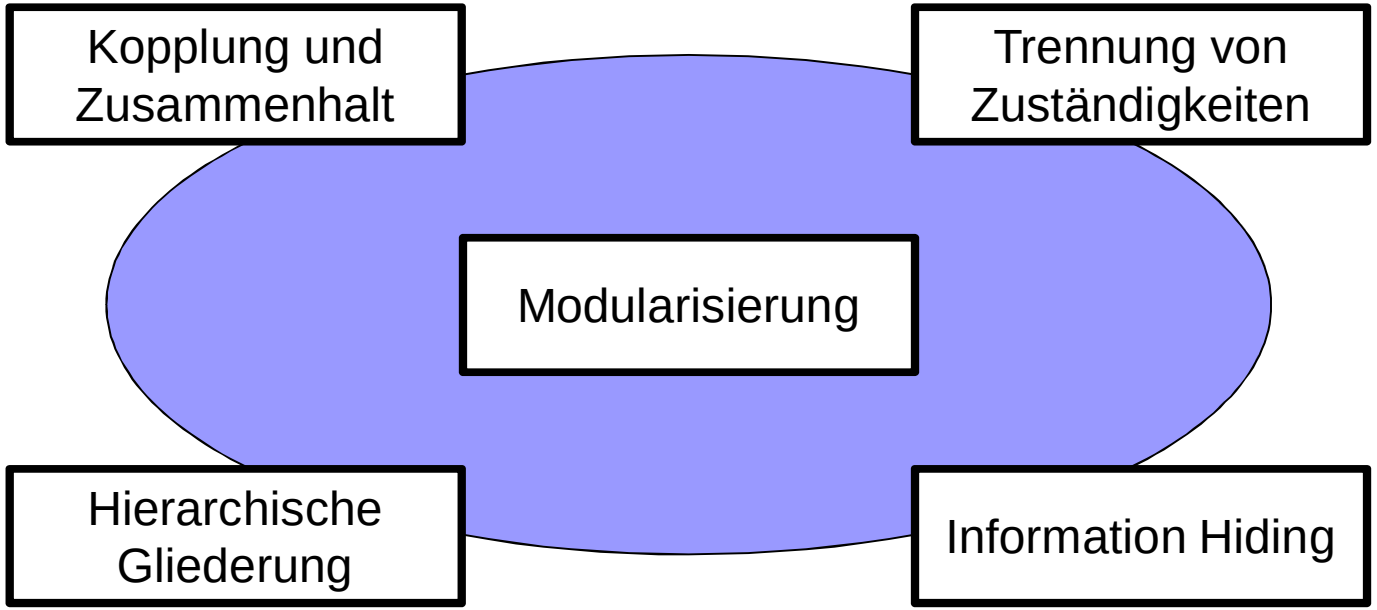
- Beispiele für Hierarchien:
  - Aggregationshierarchie: „besteht-aus“ bzw. „ist-Teil-von“ Beziehungen.
  - Schichtenhierarchie: Schicht basiert auf darunterliegenden Schicht.
  - Generalisierungshierarchie: z.B. Vererbungshierarchie.



Prinzipien des Architekturentwurfs

Wie sollen die Komponenten miteinander vernetzt sein?

Welche Eigenschaften müssen zusammengefasst werden?



Wie sollen die Komponenten verfeinert und strukturiert werden?

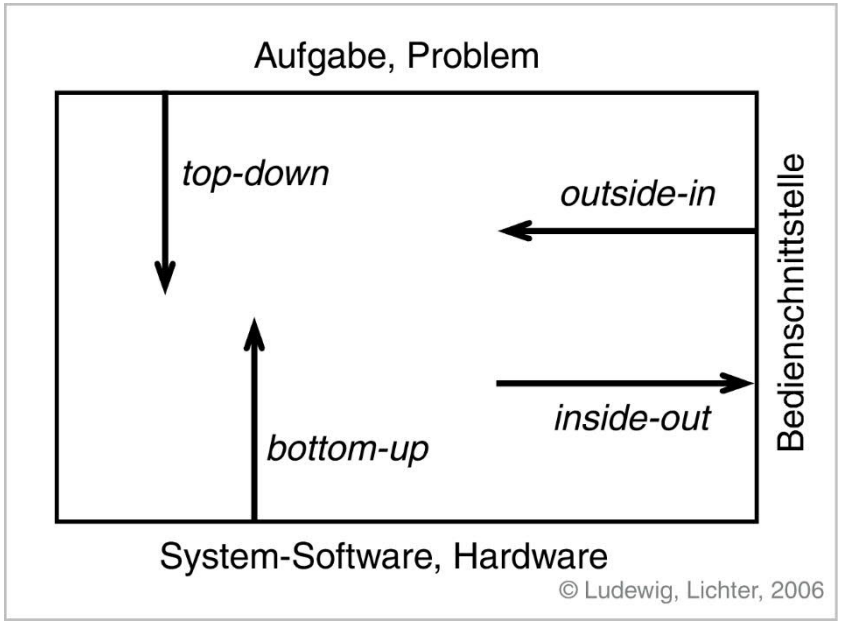
Was muss versteckt, was an der Schnittstelle angeboten werden?

Inhalt

- Ziele des Architekturentwurfs
- Begriffe
- Prinzipien des Architekturentwurfs
- **Architekturmuster**
- Entwurfsmuster
- Qualität von Softwarearchitekturen
- Lernziele

Architekturmuster

- Entweder man muss selber entwerfen …
  - Vorgehensrichtungen beim Software-Entwurf.



- … oder man kann auf bewährte Architekturmuster zurückgreifen.

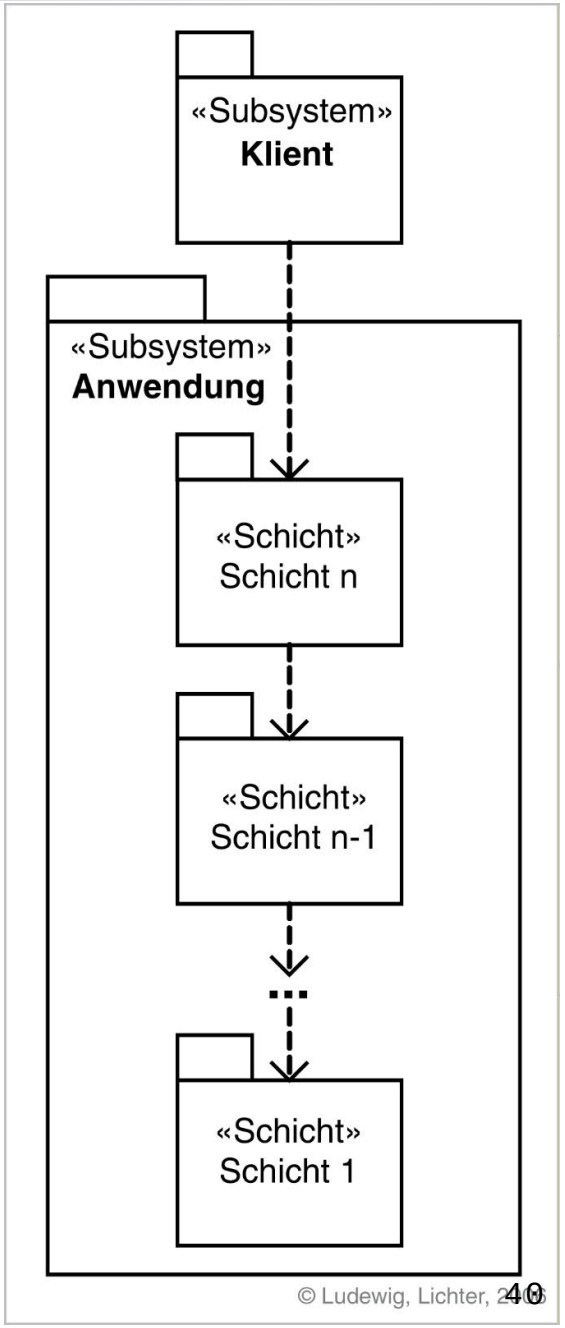
Architekturmuster

- Definition: architectural pattern
  - An **architectural pattern** expresses a fundamental structural organization schema for software systems. It provides a set of predefined subsystems, specifies their responsibilities, and includes rules and guidelines for organizing the relationships between them.
    - (Buschman et al., 1996)

- Beispiele für Architekturmuster
  - Software-Schichtenarchitektur
  - Pipe-Filter-Architekturmuster
  - Model-View-Control-Architekturmuster
  - Plug-in-Archtekturmuster

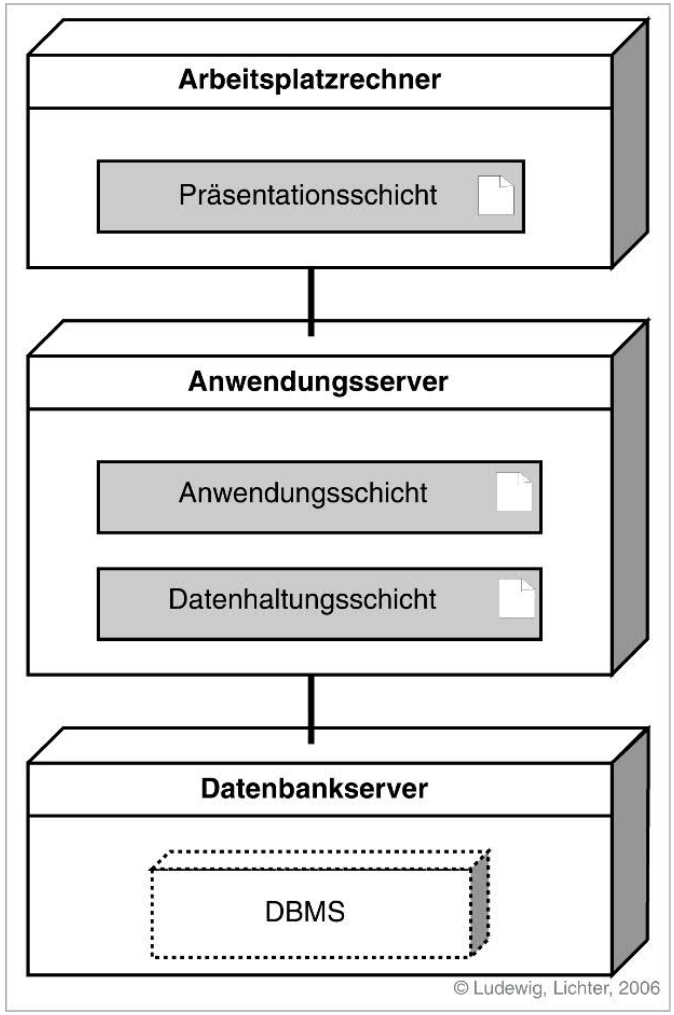
Architekturmuster - Software-Schichtenarchitektur

- Regeln für Schichten
  - Eine Schicht fasst logisch zusammengehörende Komponenten zusammen.
  - Eine Schicht stellt Dienstleistungen an ihrer oberen Schnittstelle zur Verfügung.
  - Die Dienstleistungen einer Schicht können nur von Komponenten der direkt darüber liegenden Schicht benutzt werden.
- Vorteile einer Schichtenarchitektur
  - Benachbarte Schichten sind lose über Operationen gekoppelt. Änderungen wirken sich meist nur lokal innerhalb einer Schicht aus.
  - Schichten bestehen häufig aus mehreren entkoppelten Teilen. Änderungen betreffen in der Regel nur einen einzigen Teil.



Architekturmuster – Software-Schichtenarchitektur

- Beispiel:
  - Drei-Schichten-Architektur mit physikalischer Verteilung für ein interaktives System.



Architekturmuster

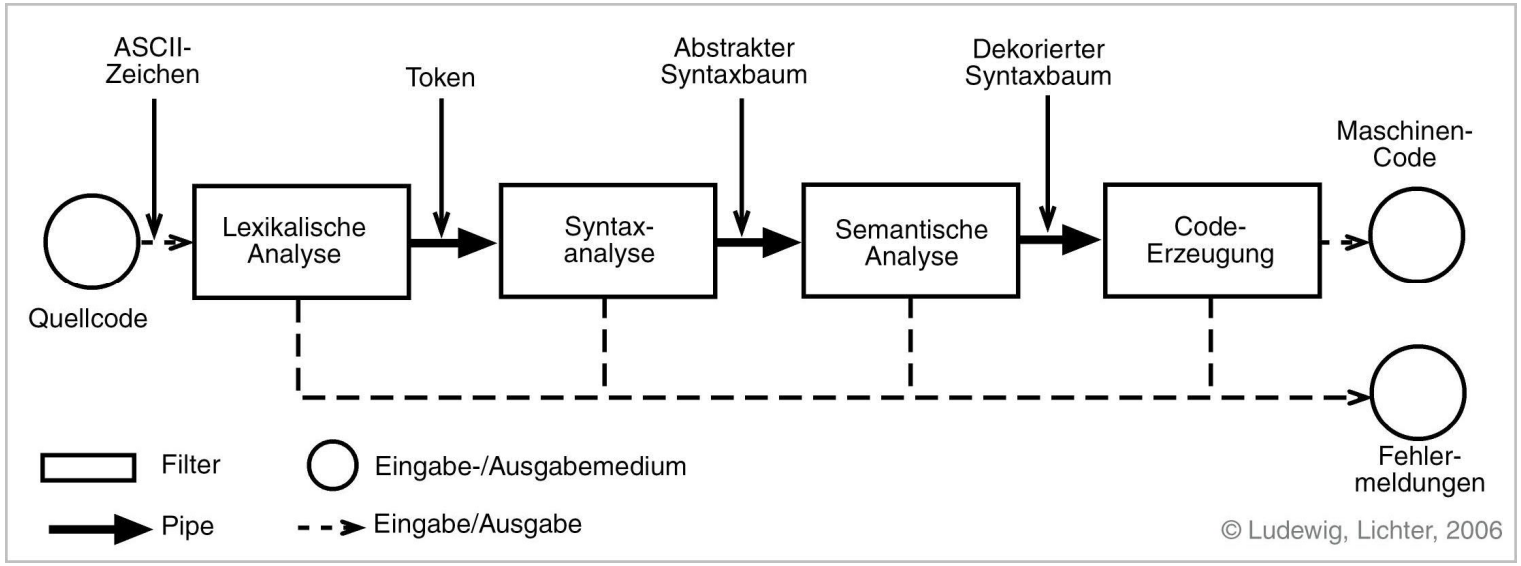
- Definition: architectural pattern
  - An **architectural pattern** expresses a fundamental structural organization schema for software systems. It provides a set of predefined subsystems, specifies their responsibilities, and includes rules and guidelines for organizing the relationships between them.
    - (Buschman et al., 1996)

- Beispiele für Architekturmuster
  - Software-Schichtenarchitektur
  - **Pipe-Filter-Architekturmuster**
  - Model-View-Control-Architekturmuster
  - Plug-in-Archtekturmuster

Architekturmuster – Pipe-Filter-Architekturmuster

- Das Pipe-Filter-Architekturmuster dient dazu eine Anwendung zu strukturieren, die Daten auf einem virtuellen Fließband verarbeiten.

- Beispiel: Pipe-Filter-Architektur eines Compilers



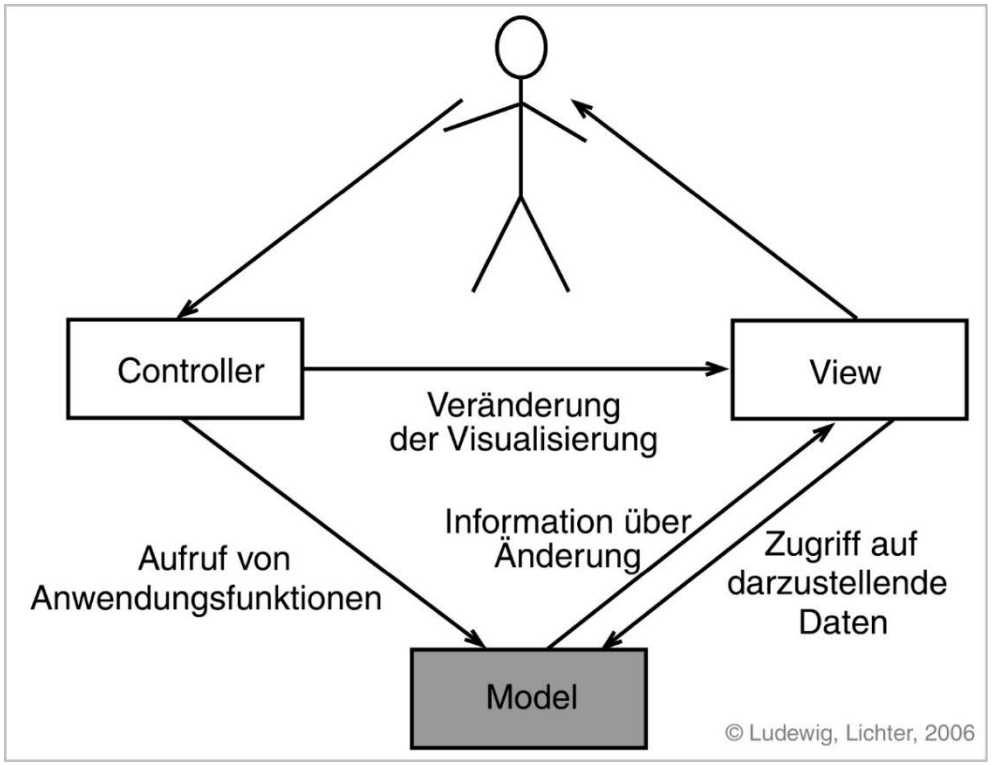
Architekturmuster

- Definition: architectural pattern
  - An **architectural pattern** expresses a fundamental structural organization schema for software systems. It provides a set of predefined subsystems, specifies their responsibilities, and includes rules and guidelines for organizing the relationships between them.
    - (Buschman et al., 1996)

- Beispiele für Architekturmuster
  - Software-Schichtenarchitektur
  - Pipe-Filter-Architekturmuster
  - **Model-View-Control-Architekturmuster**
  - Plug-in-Archtekturmuster

Architekturmuster – Model-View-Controller (MVC)

- MVC Komponenten
  - **Model** realisiert die fachliche Funktionalität der Anwendung (und kapselt die Daten der Anwendung).
  - **View** präsentiert dem Benutzer die Daten des Models.
  - Jeder View ist ein **Controller** zugeordnet. Dieser empfängt die Eingaben des Benutzers und reagiert darauf.



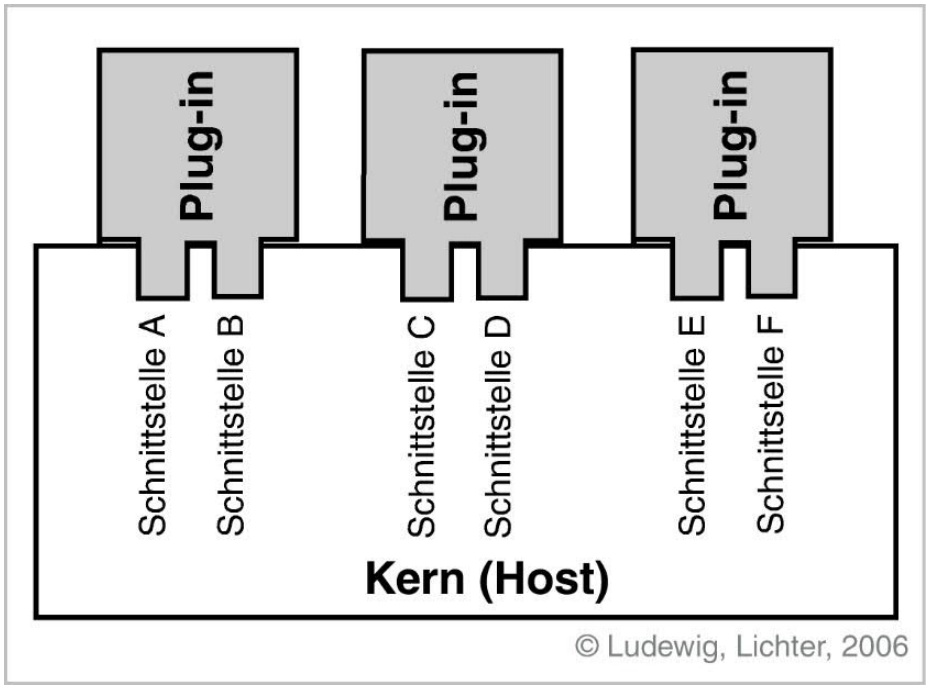
Architekturmuster

- Definition: architectural pattern
  - An **architectural pattern** expresses a fundamental structural organization schema for software systems. It provides a set of predefined subsystems, specifies their responsibilities, and includes rules and guidelines for organizing the relationships between them.
    - (Buschman et al., 1996)

- Beispiele für Architekturmuster
  - Software-Schichtenarchitektur
  - Pipe-Filter-Architekturmuster
  - Model-View-Control-Architekturmuster
  - **Plug-in-Archtekturmuster**

Architekturmuster – Plug-in-Architekturmuster

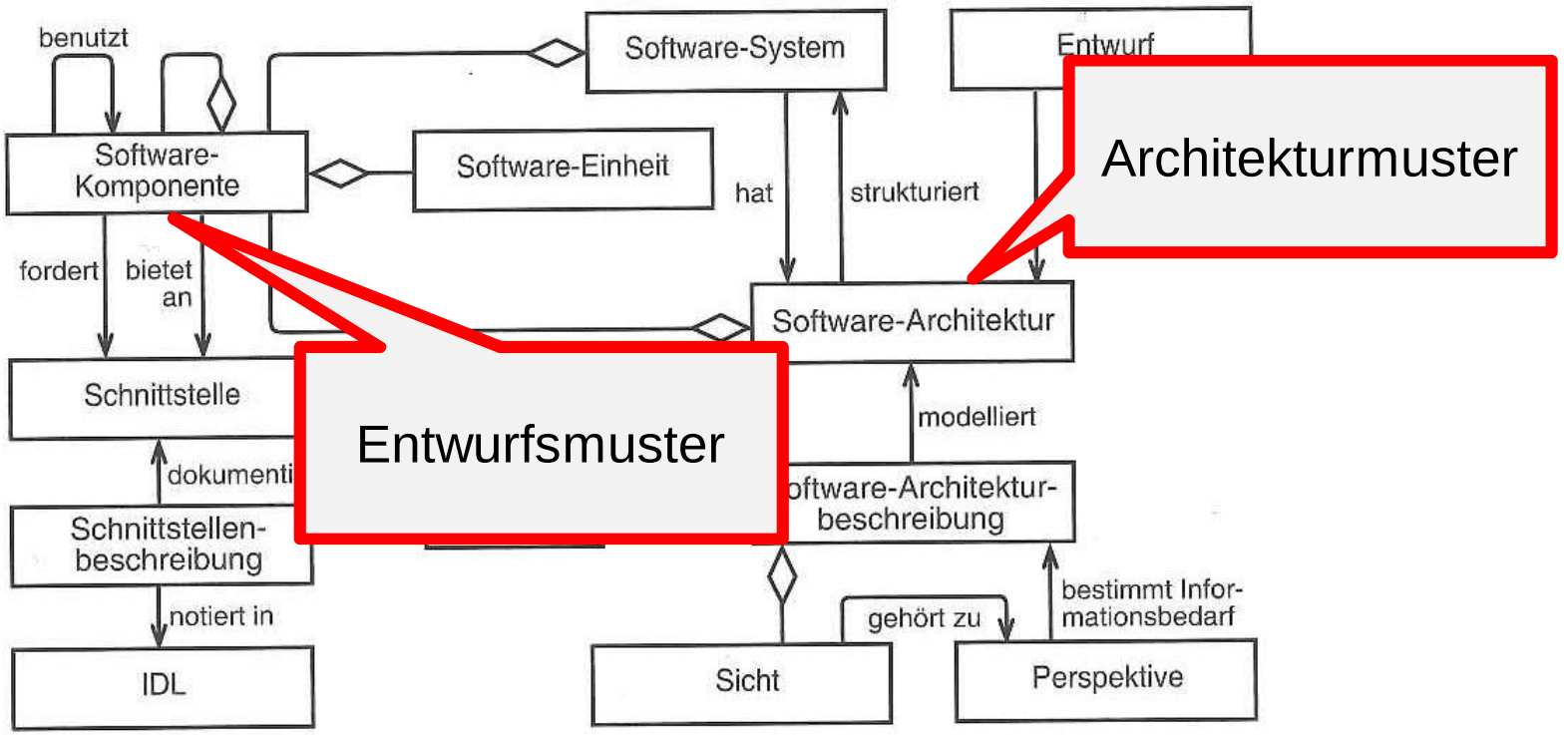
- Kern, der durch Plug-ins um neue Funktionen erweitert werden kann.
- Kern definiert Schnittstellen (Erweiterungspunkte) auf die ein Plug-in Bezug nehmen kann.
- Beispiele, die diese Architektur konsequent umsetzen:
  - Eclipse
  - Mozilla-Suite



Inhalt

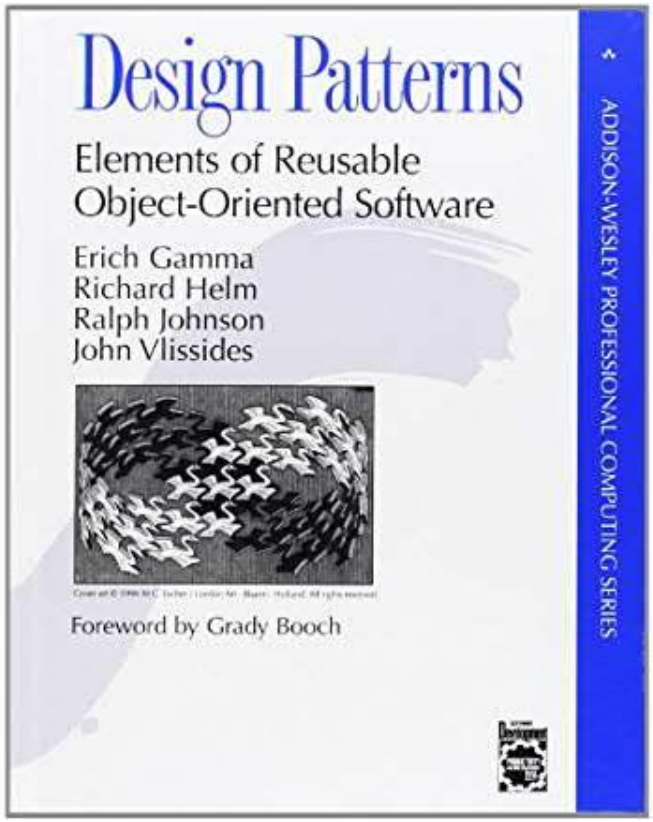
- Ziele des Architekturentwurfs
- Begriffe
- Prinzipien des Architekturentwurfs
- Architekturmuster
- **Entwurfsmuster**
- Qualität von Softwarearchitekturen
- Lernziele

Begriffe

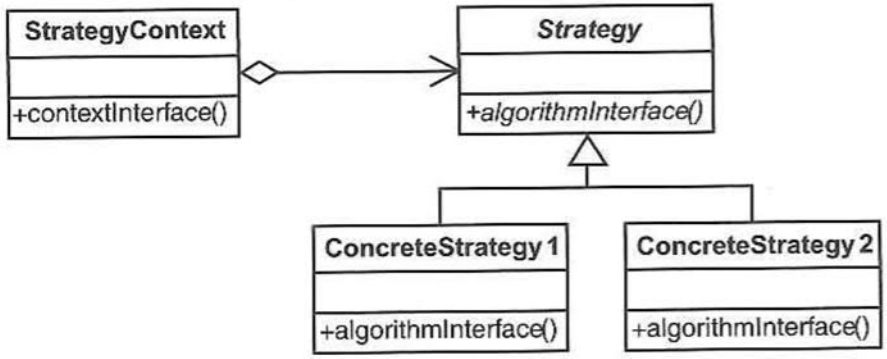


Entwurfsmuster

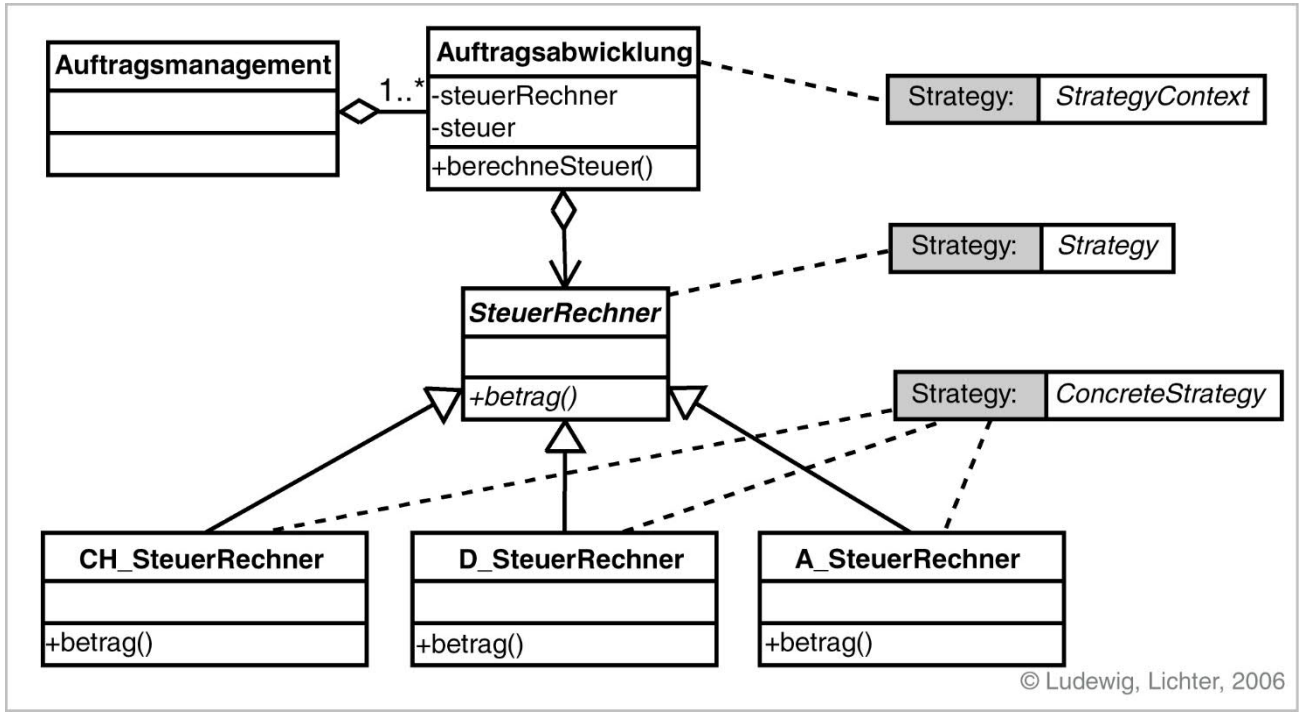
- *Design patterns … are descriptions of communicating objects and classes that are customized to solve a general design problem in a particular context. A design pattern names, abstracts, and identifies the key aspects of a common design structure that make it useful for creating a reusable object-oriented design.*
  - (Gamma et al., 1995)



Entwurfsmuster – Beispiel

Strategie (Strategy)	
Problem	Verwandte Klassen unterscheiden sich lediglich dadurch, dass sie gleiche Aufgaben teilweise durch verschiedene Algorithmen lösen.
Lösung	Die Klassen werden nicht – wie üblich – in einer Vererbungshierarchie angeordnet. Stattdessen wird eine Klasse erstellt, die alle gemeinsamen Operationen definiert (*StrategyContext*). Die Signaturen der Operationen, die unterschiedlich zu implementieren sind, werden in einer weiteren Klasse zusammengefasst (*Strategy*). Die Rolle Strategy legt somit fest, über welche Schnittstelle die verschiedenen Algorithmen genutzt werden. Von dieser Klasse wird für jede Implementierungsalternative eine konkrete Unterklasse abgeleitet (*ConcreteStrategy*). Die Klasse mit der Rolle StrategyContext benutzt konkrete Strategy-Objekte, um die unterschiedlich implementierten Operationen per Delegation auszuführen.
Struktur	

Entwurfsmuster – Beispiel



Entwurfsmuster – Überblick

- **Muster zum Verwalten von Objekten**
  - *Einzelstück (Singleton)*
  - *Memento*
- **Muster zur Entkopplung von Komponenten**
  - *Brücke (Bridge)*
  - *Fassade (Facade)*
  - *Beobachter (Observer)*
- **Muster zur Anbindung vorhandener Klassen oder Komponenten**
  - *Adapter*
  - *Vermittler (Mediator)*
- **Muster zur Trennung der unterschiedlichen von den gemeinsamen Merkmalen**
  - *Abstrakte Fabrik (Abstract Factory)*
  - *Schablonenmethode (Template Method)*
  - *Strategie (Strategy)*
  - *Zustand (State)*

Inhalt

- Ziele des Architekturentwurfs
- Begriffe
- Prinzipien des Architekturentwurfs
- Architekturmuster
- Entwurfsmuster
- **Qualität von Softwarearchitekturen**
- Lernziele

Qualität von Softwarearchitekturen

- Qualitätskriterien
  - Testbarkeit
  - Wartbarkeit, Erweiterbarkeit
  - Portierbarkeit
- Qualitätsprüfung
  - Reviews
  - Metriken

Inhalt

- Ziele des Architekturentwurfs
- Begriffe
- Prinzipien des Architekturentwurfs
- Architekturmuster
- Entwurfsmuster
- Qualität von Softwarearchitekturen
- **Lernziele**

Lernziele

- Was sind die Ziele des Architekturentwurfs?
- Was sind die Prinzipien des Architekturentwurfs?
- Was sind Architekturmuster und welche typischen Architekturmuster wurden in der Vorlesung behandelt?
- Was sind Entwurfsmuster und für welche Design Probleme gibt es Entwurfsmuster?
- Wie bewertet man die Qualität von Softwarearchitekturen?