



Technische Hochschule
Ingolstadt

Fakultät für Elektrotechnik
und Informatik

*Zukunft in
Bewegung*

Prinzipien für einen guten Entwurf

Architektur- und Entwurfsmuster der Softwaretechnik

Prof. Dr. Bernd Hafenrichter



Prinzipien einer guten Architektur (Struktursicht)

- Auf der Struktursicht ist besonders auf eine hohe Kohäsion und niedrige Kopplung zu achten.
- Dadurch können Subsystem ausgetauscht bzw. verändert werden mit der Idee die Auswirkungen lokal begrenzen zu können

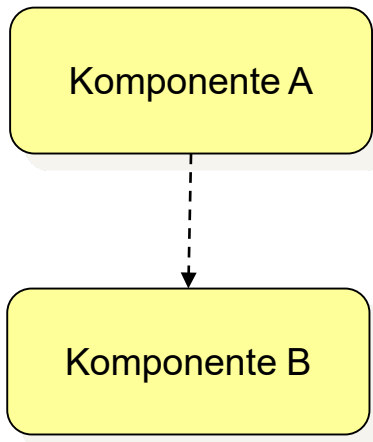
Hohe Kohäsion:

- Kohäsion = "Zusammenhalt"
- Die Dinge sollen in Struktureinheiten zusammengefasst werden, die inhaltlich zusammengehören.

Niedrige Kopplung:

- Kopplung = Abhängigkeiten
- Einzelne Struktureinheiten sollen möglichst unabhängig voneinander sein

Prinzipien einer guten Architektur (Struktursicht)



Hohe Kohäsion:

Innerer Zusammenhalt der Komponenten. Die Internen Klassen haben intern eine hohe Abhängigkeit und bearbeiten gemeinsam ein Thema.

Niedrige Kopplung:

Es muss möglich sein, Komponente A weitgehend auszutauschen oder zu verändern, ohne Komponente B zu verändern.

Änderungen von Komponente B sollten nur möglichst einfache Änderungen in Komponente A nach sich ziehen.



Prinzipien einer guten Architektur (Struktursicht)

Kohäsion und Kopplung können auf der Grundlage von Abhängigkeiten zwischen Softwarekomponenten definiert werden

Wann immer eine Softwarekomponente A eine Softwarekomponente B benötigt, besteht eine Abhängigkeit zwischen diesen beiden Artefakten.

Arten von Abhängigkeiten

- Komponente A ruft eine Methode in Komponente B auf (Schnittstellenkopplung)
- Klasse A hängt von der Struktur von Klasse B ab (Strukturkopplung)
- Klasse A kann auf Eigenschaften von Klasse B zugreifen (Datenkopplung)
- Geringe Kopplung: Minimierung der Anzahl von Abhängigkeiten zwischen Komponenten
- Hohe Kohäsion: Alle Klassen in einem Modul haben eine große Anzahl von Abhängigkeiten

Prinzipien einer guten Architektur (Struktursicht)

Datenkopplung: Klasse A kann auf Eigenschaften von Klasse B direkt, ohne Prüfung zugreifen. Kommunikation über globale Daten

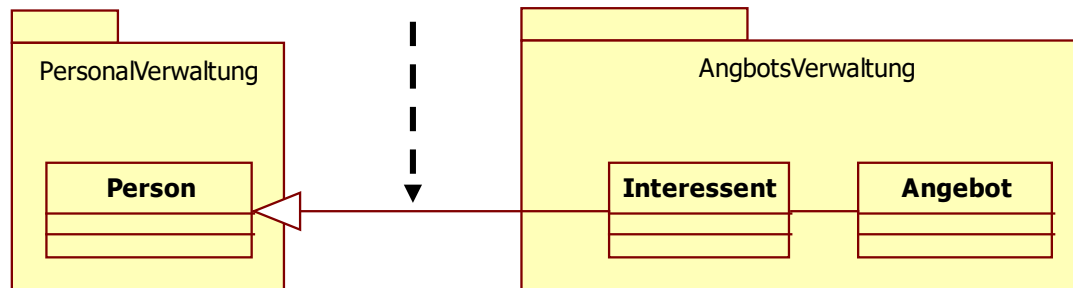


- Evaluation :
 - Zugriff auf interne Datenstrukturen einer Komponente
 - Wissen über die Verwendung der Variablen ist notwendig
 - Problematisch im Falle von Änderungen.
 - Einhaltung von Invarianten nicht möglich (Variablen können anders benutzt werden als von der Klasse vorgesehen)

Prinzipien einer guten Architektur (Struktursicht)

Strukturkopplung: Klasse A wird von Klasse B abgeleitet

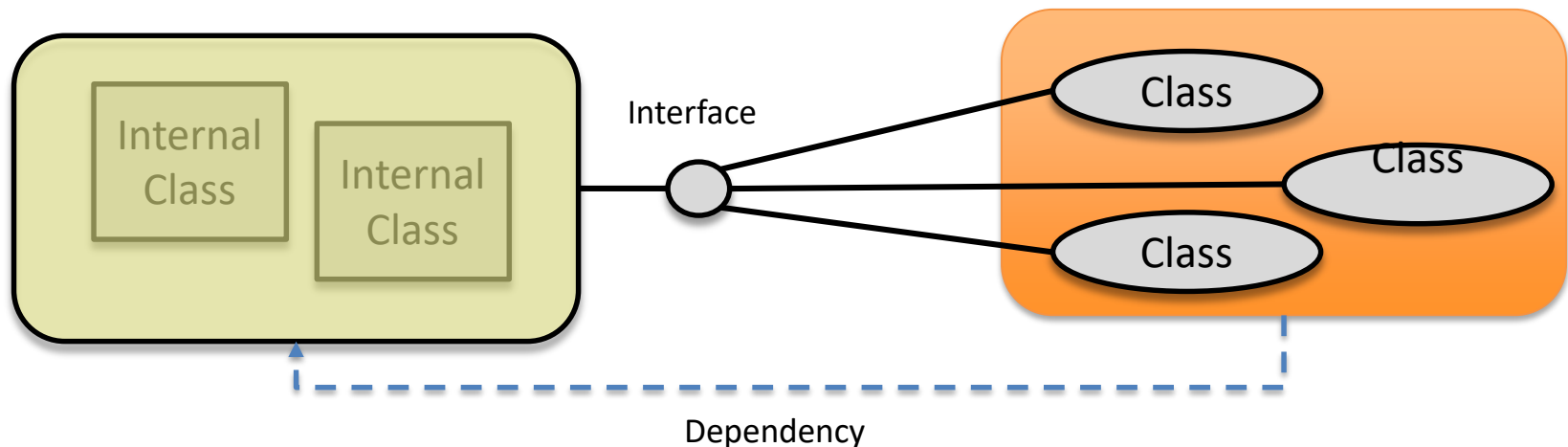
Abhängigkeiten: Strukturabhängigkeit



- Evaluation :
 - Ein Modul ist von Strukturelementen eines anderen Moduls abhängig
 - Teilsysteme sollten unabhängig voneinander entwickelt/ausgetauscht werden können.
 - Änderungen and der Struktur ziehen Änderungen in allen Modulen nach sich welche die selbe Struktur verwenden
 - Die "strukturelle Kopplung" führt zu einer sehr hohen Abhängigkeit, die sich negativ auf die Stabilität des Systems auswirkt.

Prinzipien einer guten Architektur (Struktursicht)

Abhängigkeiten: Schnittstellenkopplung

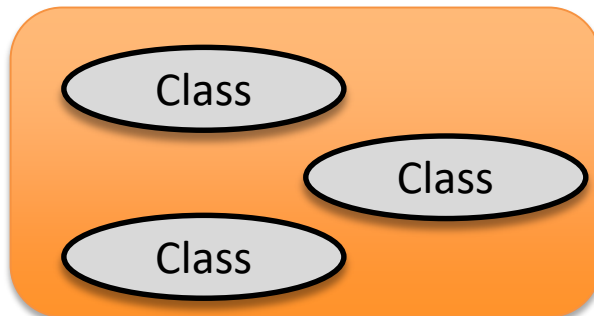


- Evaluation :
 - Zugang nur über wohldefinierte Schnittstellen/Methoden, die Implementierungsdetails verbergen.
 - Annehmbar. Interne Änderungen haben keine Auswirkungen auf den Aufrufer
 - Problematisch: Änderung der Schnittstelle

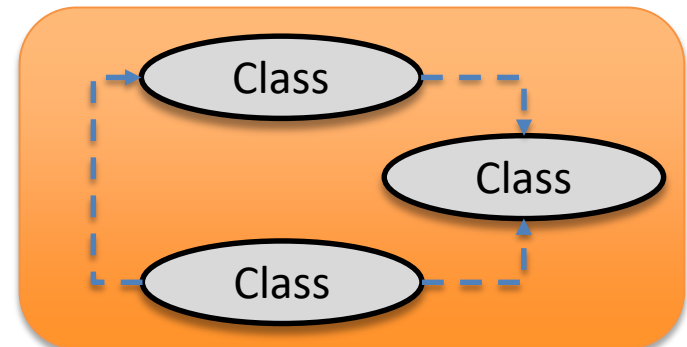
Prinzipien einer guten Architektur (Struktursicht)

Kohäsion :

- Kohäsion ist ein Maß dafür, wie gut die Teile einer Komponente zusammengehören.
- Eine hohe Kohäsion einer Komponente erleichtert das Verständnis, die Wartung und die Anpassung.
- Frühere Kohäsionsansätze, wie das Zusammenfassen ähnlicher Funktionalitäten, führten nicht unbedingt zu einer stabilen Systemstruktur.
- Abhängigkeiten entstehen durch den Zugriff auf unterschiedliche Quellcode-Artefakte.



Schlechte Kohäsion aufgrund sehr geringer Abhängigkeiten innerhalb der Komponente. Die Klassen tun unterschiedliche Dinge und arbeiten nicht zusammen.

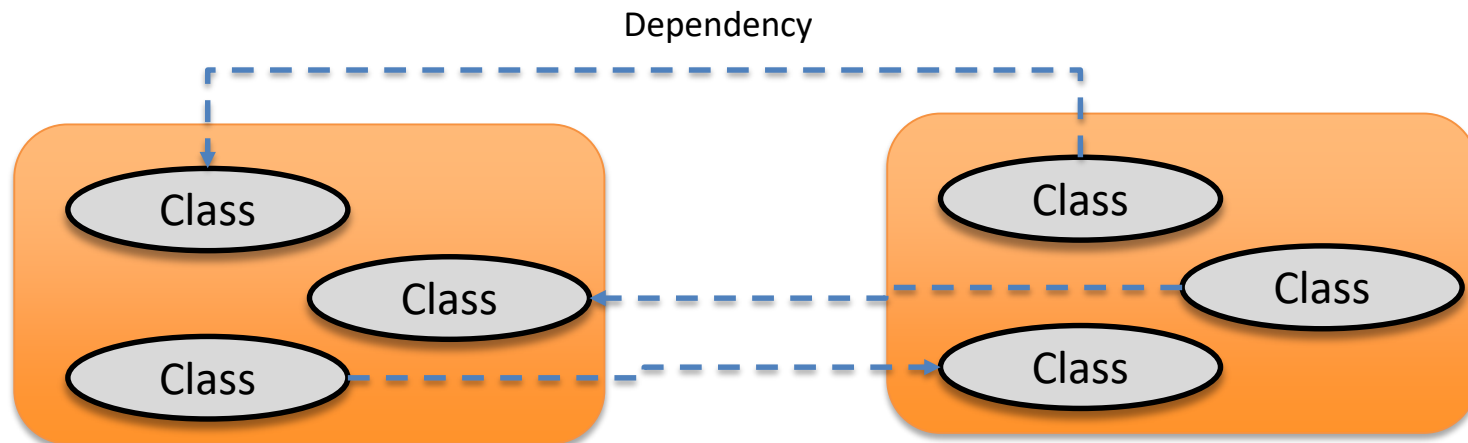


Guter Zusammenhalt, da die Klassen eine Aufgabe gemeinsam erfüllen. Die Klassen sind voneinander abhängig. Sie arbeiten zusammen.

Prinzipien einer guten Architektur (Struktursicht)

Kopplung :

- Die Kopplung ist ein Maß für die Abhängigkeiten zwischen den Komponenten.
- Eine geringe Kopplung erleichtert die Wartbarkeit und macht die Systeme stabiler.

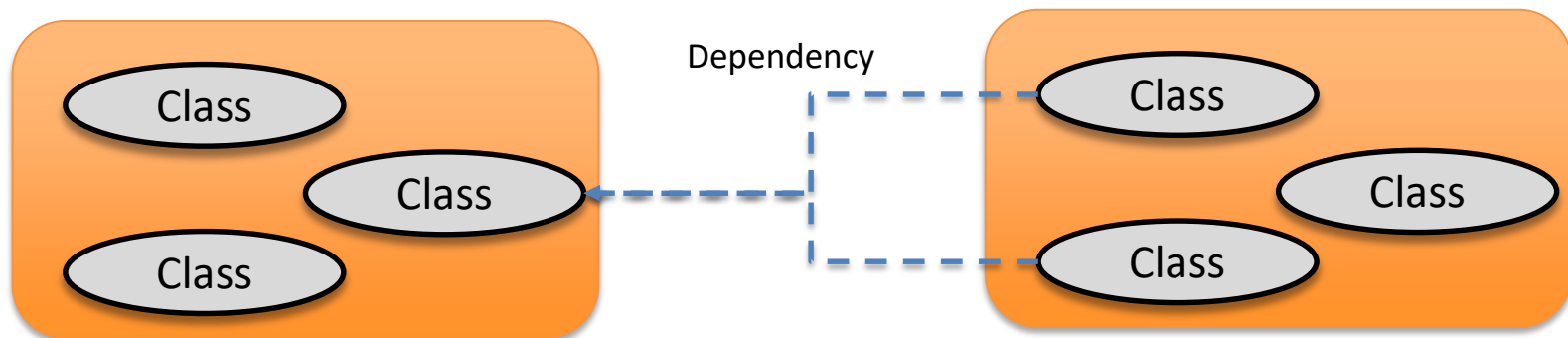


Hohe Kopplung: Sehr hohe Abhängigkeiten zwischen Modulen. Eine Änderung an einem Modul hat eine (unbeabsichtigte) Auswirkung auf andere Module.

Prinzipien einer guten Architektur (Struktursicht)

Kopplung :

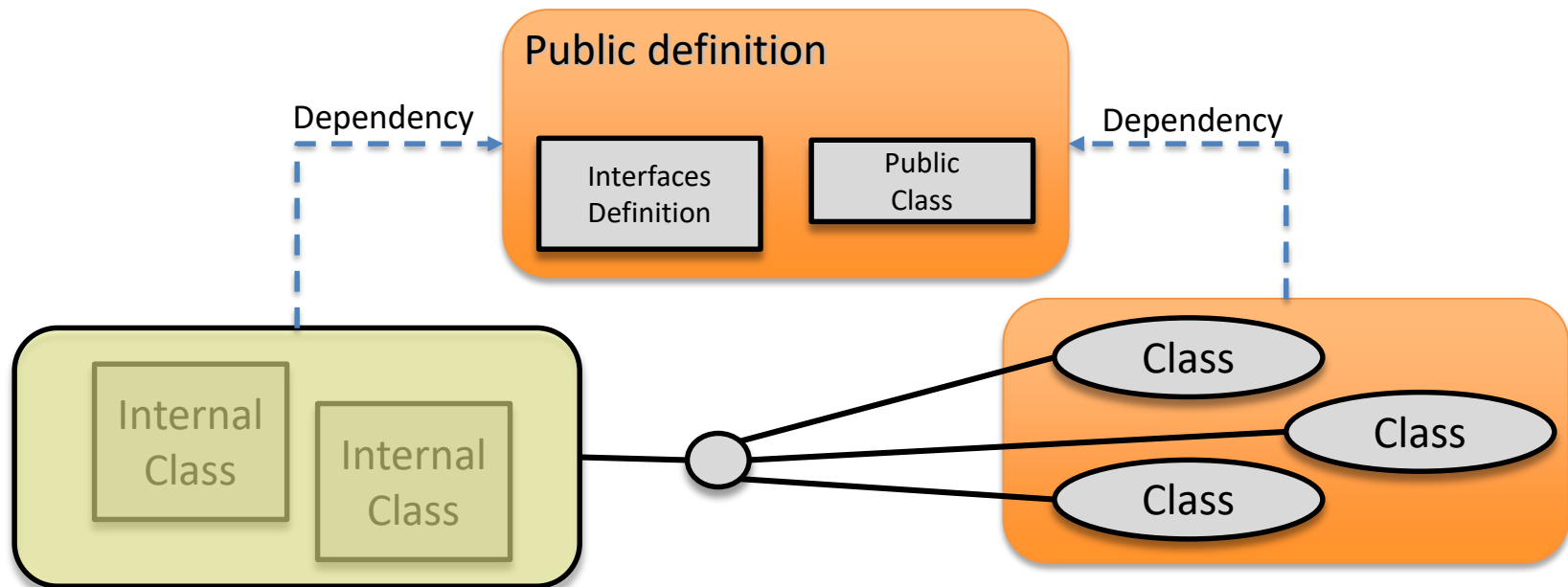
- Die Kopplung ist ein Maß für die Abhängigkeiten zwischen den Komponenten.
- Eine geringe Kopplung erleichtert die Wartbarkeit und macht die Systeme stabiler.



Geringe Kopplung: Die Abhängigkeiten bestehen zu wenigen Elementen des gekoppelten Moduls. Eine Änderung an einem Modul hat weniger Auswirkungen auf andere Module

Prinzipien einer guten Architektur (Struktursicht)

Schnittstellenkopplung über öffentliche Schnittstellen



- Evaluation :
 - Die Module sind nicht mehr direkt voneinander abhängig. Eine getrennte Entwicklung ist möglich.
 - Änderungen an der öffentlichen Definition erfordern Änderungen an allen abhängigen Modulen. Öffentliche Definition stabil halten



Prinzipien einer guten Architektur (Struktursicht)

Reduzierung der Kopplung zwischen Modulen auf ein Minimum

- Reduzierung der Kopplung :
 - Die Kopplung kann nie auf Null reduziert werden (keine Kommunikation zwischen den Modulen).
 - Reduzieren Sie die Kopplung auf ein Minimum
 - Vermeiden Sie Datenkopplung zwischen Modulen
 - Verwenden Sie Strukturkopplung nur falls notwendig.
 - Schnittstellenkopplung ist wegen der höheren Flexibilität akzeptabel
 - Wenn möglich, Schnittstellendefinition in öffentliche Pakete verschieben.

Achtung! Das Ziel muss immer eine geringe Kopplung zwischen verschiedenen Modulen/Subsystemen sein



Kriterien für "guten" Entwurf

Interne Wiederverwendung:

- Interne Wiederverwendung (reuse) ist ein Maß für die Ausnutzung von Gemeinsamkeiten zwischen Komponenten
- Reduktion der Redundanz
- Erhöhung der Stabilität und Ergonomie
- Hilfsmittel für Wiederverwendung:
 - im objektorientierten Entwurf: Vererbung, Parametrisierung
 - im modularen und objektorientierten Entwurf: Module/Objekte mit allgemeinen Schnittstellen (Interfaces)
- Aber: Wiederverwendung kann die Kopplung erhöhen:
 - Schnittstellenkopplung und Strukturkopplung:

Achtung: Das Ziel muss sein eine hohe Wiederverwendbarkeit zu erreichen.
Hierbei solle man aber nicht um jeden Preis verallgemeinern



Prinzipien einer guten Architektur (Struktursicht)

Information Hiding (Geheimhaltungsprinzip):

- Motivation (Militärisch):
 - Um eine Operation auszuführen darf nur soviel Wissen weitergegeben werden wie unbedingt zur Erfüllung der Aufgabe ist
 - Reduziert die Gefahr des Informationsmissbrauchs
- Parnas hat dieses Prinzip auf die Software-Entwicklung übertragen
 - Die Art und Weise wie ein Systemteil seine Aufgabe erfüllt muss im inneren des Moduls verborgen bleiben
 - Über das Modul sind nach außen nur Informationen bekannt welche über eine Schnittstelle explizit zur Verfügung gestellt worden sind
 - Über eine Schnittstelle wird so wenig wie möglich nach außen hin preis gegeben

Prinzipien einer guten Architektur (Struktursicht)

Information Hiding (Geheimhaltungsprinzip):

- Umsetzung mit Hilfe der Objektorientierung: „Kapselung“
- Zugriff auf den inneren Zustand eines Objektes erfolgt nur über wohldefinierte Methoden
- Das Objekt selbst ist verantwortlich und kann über die Zulässigkeit entscheiden



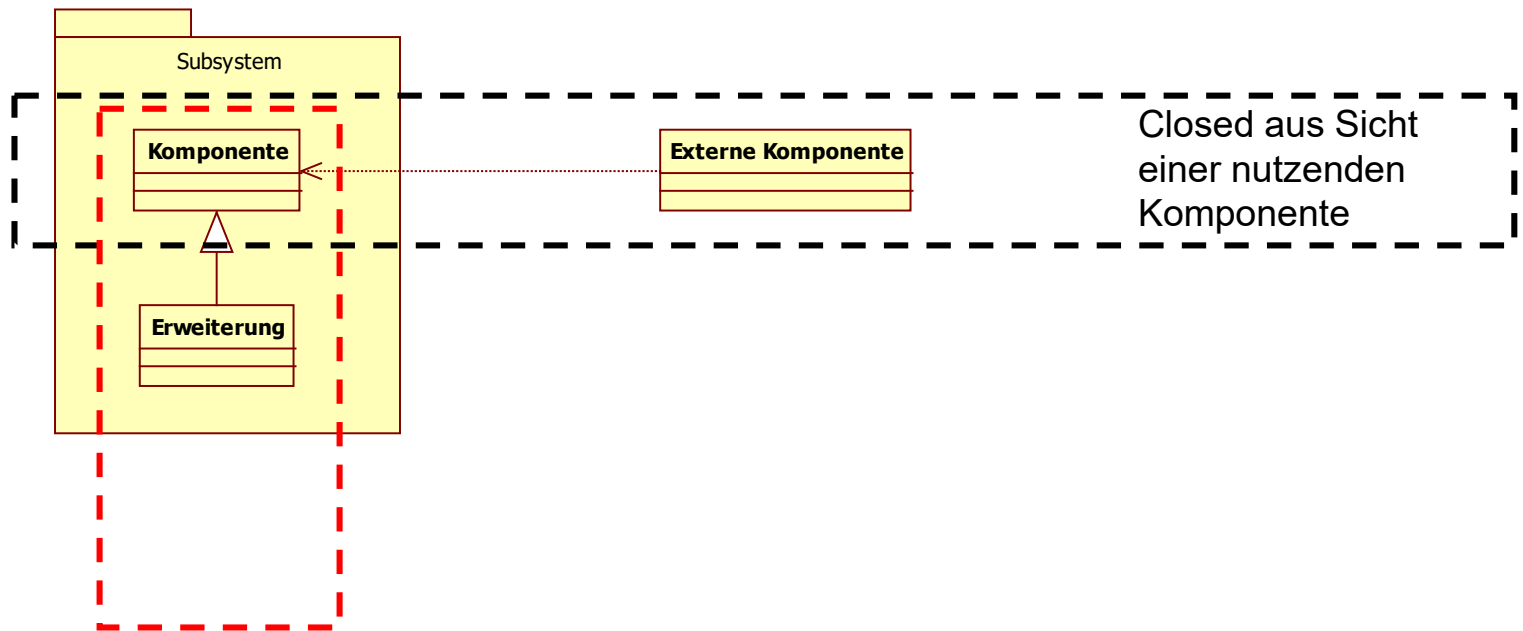
Prinzipien einer guten Architektur (Struktursicht)

Open-Closed-Principle:

- Software-Einheiten sollten offen für Erweiterungen, aber geschlossen für Modifikationen sein.
 - Eine Erweiterung in diesem Sinne verändert das vorhandene Verhalten der Einheit nicht, vielmehr erweitert es die Einheit um zusätzliche Funktionen oder Daten
 - Eine Modifikation hingegen, würde das bisherige Verhalten der Einheit ändern.

Prinzipien einer guten Architektur (Struktursicht)

Open-Closed-Principle:



Erweiterbar aus ohne das „closed“ zu verletzen

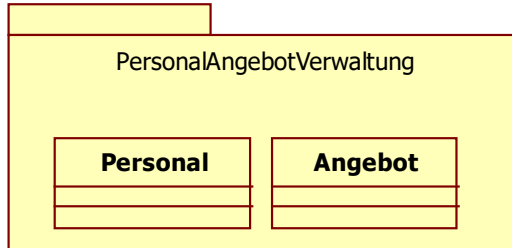
Prinzipien einer guten Architektur (Struktursicht)

Seperation-Of-Concerns:

- Definition von Concern:
 - Spezifische Anforderung oder Gesichtspunkt, welche(r) in einem Software-System behandelt werden muss, um die übergreifenden Systemziele zu erreichen (Ramnivas Laddad, 2003).
 - Maßgebliches Systemziel auf oberster Ebene, das üblicherweise aus den kritischen Geschäftszielen des Auftraggebers abgeleitet wird
- Unterschiedliche „Concerns“ sollen durch unterschiedliche Systemteile realisiert werden
- Verschiedene Elemente der Aufgabe sollten möglichst in verschiedenen Elementen der Lösung repräsentiert werden.

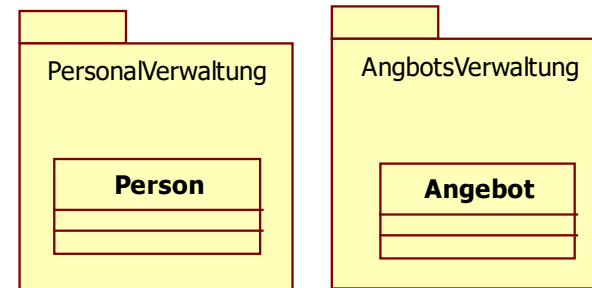
Prinzipien einer guten Architektur (Struktursicht)

Seperation-Of-Concerns:



Schlechte Struktur

Die Wissensbereiche Personal und Angebot sind miteinander in einem Subsystem vermischt



Gute Struktur

Die Wissensbereiche Personal und Angebot sind separiert und können somit unabhängig voneinander entwickelt werden



Prinzipien einer guten Architektur (Struktursicht)

Weitere Qualitätskriterien:

Korrektheit, Anpassbarkeit, Verständlichkeit, Ressourcenschonung