



Fachhochschule Osnabrück  
University of Applied Sciences

# Aspektorientierte Programmierung mit .NET



Inhalt

Motivation

Problemstellung

Crosscutting  
Concerns

Was ist AOP?

Vorgehen bei AOP

Arten von AOP

Vorteile

Anwendungen

Entwicklungsstand

Quellen

1. Motivation
2. Problemstellung
3. Crosscutting Concerns
4. Was ist AOP?
5. Vorgehen beim AOP
6. Arten von AOP
7. Vorteile durch AOP
8. Anwendungen
9. Entwicklungsstand
10. Quellen



# 1. Motivation

Inhalt

Motivation

Problemstellung

Crosscutting  
Concerns

Was ist AOP?

Vorgehen bei AOP

Arten von AOP

Vorteile

Anwendungen

Entwicklungsstand

Quellen

## Motivation

- AOP ist ein mächtiges „Werkzeug“ für den „Entwickler-Werkzeugkoffer“
- AOP könnte zunehmend an Bedeutung gewinnen
  - kommerzielle Nutzung
  - Aspect#, Spring.NET
  - immer mehr Artikel, Bücher, Vorträge

## Ziel

- Grundverständnis über AOP
- Mächtigkeit und Möglichkeiten erkennen
- Interesse wecken



## 2. Problemstellung Core Concern (Kern-Anliegen/Verantwortung)

Inhalt

Motivation

Problemstellung

Crosscutting  
Concerns

Was ist AOP?

Vorgehen bei AOP

Arten von AOP

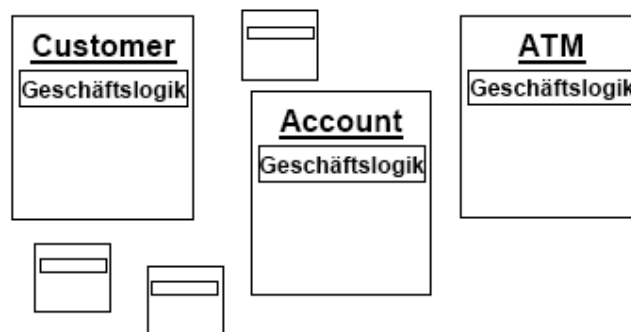
Vorteile

Anwendungen

Entwicklungsstand

Quellen

- Jede Anwendung hat Kern-Anliegen/Verantwortung (in AOP „Core Concerns“ genannt).  
Beispiel: Geschäftsanwendung: Geschäftslogik
- Primäre Aufgabe des Entwicklers ist die Umsetzung der Kern-Verantwortung.
- Kapselung durch Aufteilung der Verantwortlichkeiten in Klassen.  
Unterstützung: Vererbung, Trennung von Schnittstelle/Implementierung.
- reale Welt(Geschäftsmodel) durch OO-Design gut abbildbar.





## 2. Problemstellung Crosscutting Concerns

Inhalt

Motivation

Problemstellung

Crosscutting  
Concerns

Was ist AOP?

Vorgehen bei AOP

Arten von AOP

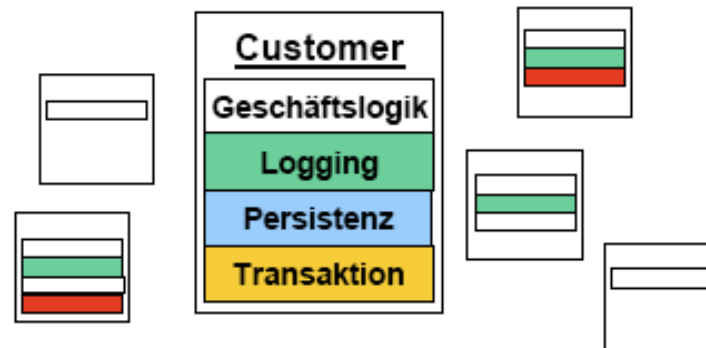
Vorteile

Anwendungen

Entwicklungsstand

Quellen

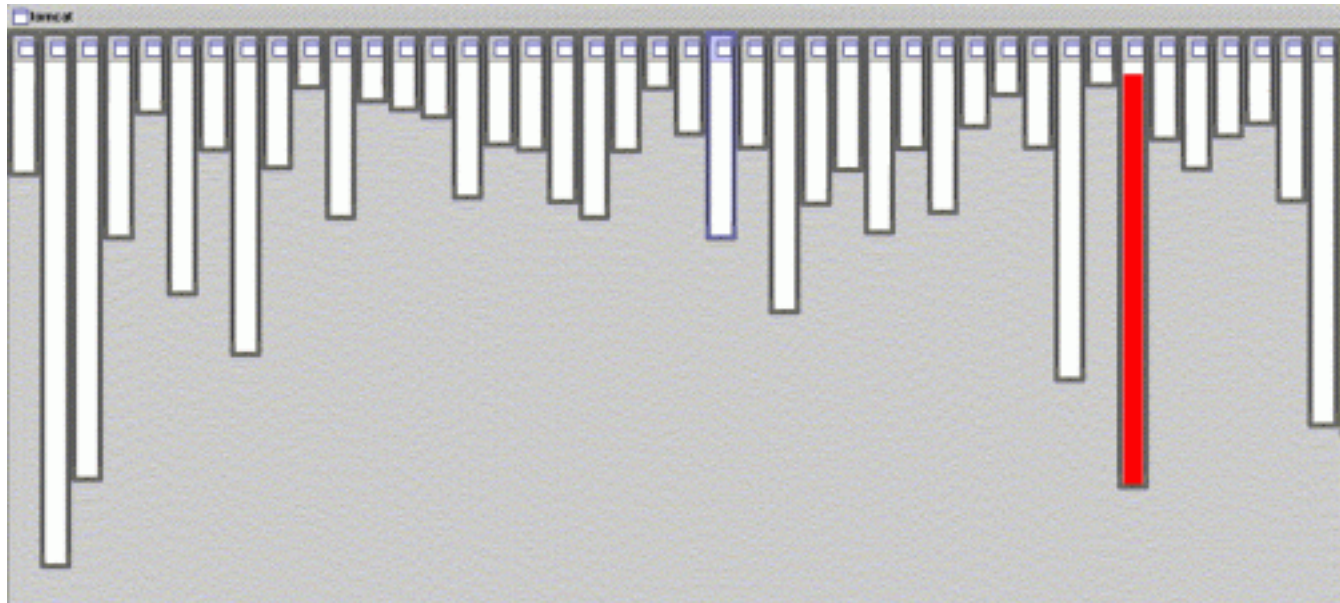
- weitere Aufgabe des Entwicklers ist die Umsetzung technischer Anliegen, z.B. Logging, Persistenz, Ausnahmebehandlung, usw.
- Logging, Persistenz, usw. sind systemweit verteilt und werden in AOP daher „Crosscutting Concerns“ (querschneidende Verantwortlichkeiten) genannt.
- Crosscutting Concerns müssen an vielen Stellen implementiert werden





### 3. Crosscutting Concerns (OOP)

#### Quellcode des Apacheservers

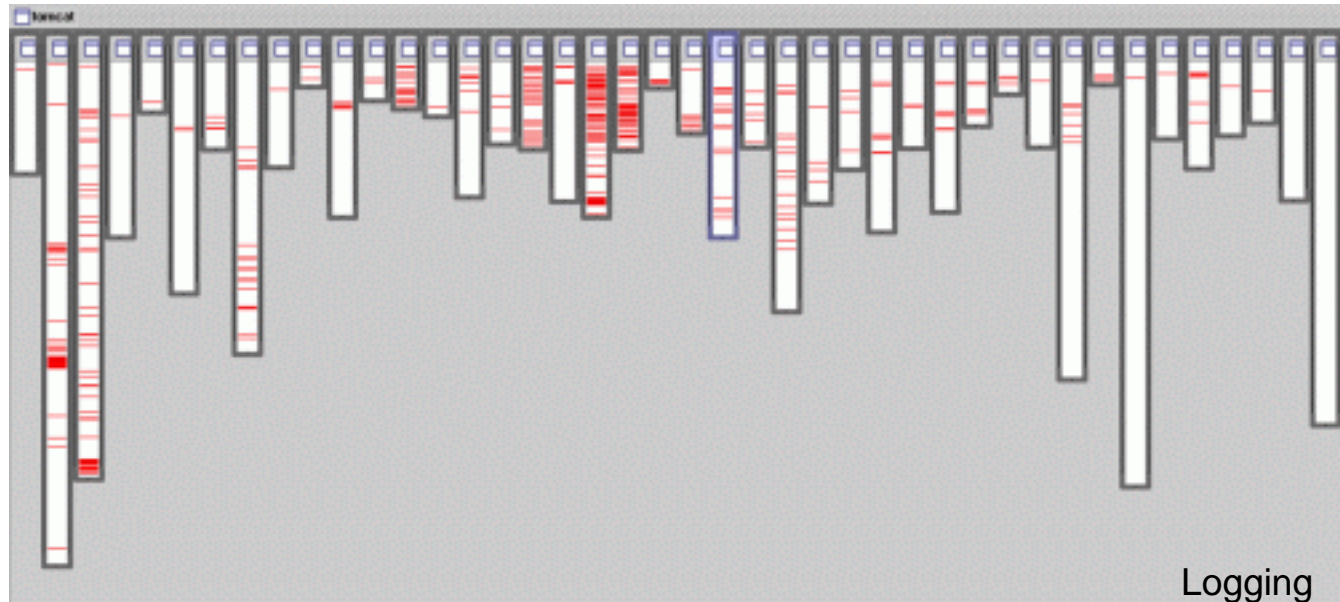


- Problem in Objekte zerlegt
- gut gekapselt



### 3. Crosscutting Concerns (OOP)

Logging ist aber über den gesamten Code verteilt



- gewisse Problemaspekte lassen sich nicht ordentlich herausfaktorisieren
- sind über weite Teile des Codes verteilt



### 3. Crosscutting Concerns Symptome

Inhalt

Motivation

Problemstellung

Crosscutting  
Concerns

Was ist AOP?

Vorgehen bei AOP

Arten von AOP

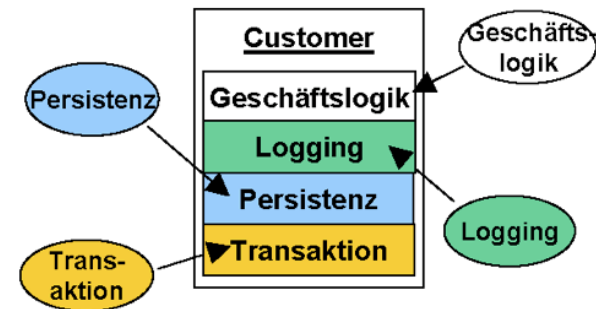
Vorteile

Anwendungen

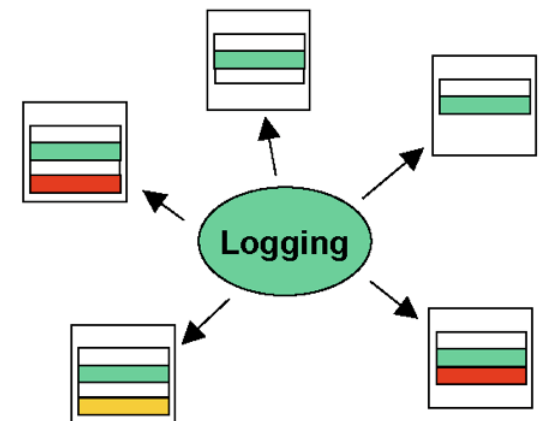
Entwicklungsstand

Quellen

- Code Tangling (Code-Vermischung)
  - Eine Klasse ist für mehrere Anliegen verantwortlich
- Beispiel: Geschäftslogik, Logging, Persistenz, Transaktion



- Code Scattering (Code-Verstreung)
  - Ein Anliegen ist über mehrere Klassen verteilt
  - Beispiel: Logging







## 3. Crosscutting Concerns Nachteile

Inhalt

Motivation

Problemstellung

Crosscutting  
Concerns

Was ist AOP?

Vorgehen bei AOP

Arten von AOP

Vorteile

Anwendungen

Entwicklungsstand

Quellen

Durch Code-Vermischung und -Vertreuung:

- Entsteht Redundanter Code. Gleiche oder ähnliche Code-Fragmente finden sich an vielen Stellen
- Sind Änderungen umfangreicher und fehleranfälliger
- Ist Debugging, Wartung, Erweiterung schwerer
- Ist Wiederverwendung und Produktivität geringer.  
(z.B Geschäftslogik und technische Details vermischt)
- Ist Code-Qualität schlechter



## 4. Was ist AOP?

Inhalt

Motivation

Problemstellung

Crosscutting  
Concerns

Was ist AOP?

Vorgehen bei AOP

Arten von AOP

Vorteile

Anwendungen

Entwicklungsstand

Quellen

- Erweiterung bestehender Paradigmen (OO, Prozedural/Imperativ) zur Modularisierung von Crosscutting Concerns
- OO: „Betrachten“ von Objekten als Modularisierungseinheit. Crosscutting Concerns als nicht modularisierbare Objekte(Aspekte)
- AOP: „Betrachten“ von Aspekten als Modularisierungseinheit
- Crosscutting Concerns als Aspekte modularisieren

Aspekt: Gesichtspunkt unter dem man etwas betrachtet



## 5. Vorgehen beim AOP

Inhalt

Motivation

Problemstellung

Crosscutting  
Concerns

Was ist AOP?

Vorgehen bei AOP

Arten von AOP

Vorteile

Anwendungen

Entwicklungsstand

Quellen

- Ein Programm folgt einem Programmablaufplan
- ein Aspekt ist eine Schablone, die über diesen Originalplan gelegt wird
- führt verschiedene Änderungen oder Ergänzungen am Ablaufschema durch
- Änderungen an der Schablone lassen den Originalplan unangetastet
- die Schablone lässt sich jederzeit austauschen oder entfernen



## 5. Vorgehen beim AOP

### Crosscutting Concerns mit OO and AOP

Inhalt

Motivation

Problemstellung

Crosscutting  
Concerns

Was ist AOP?

Vorgehen bei AOP

Arten von AOP

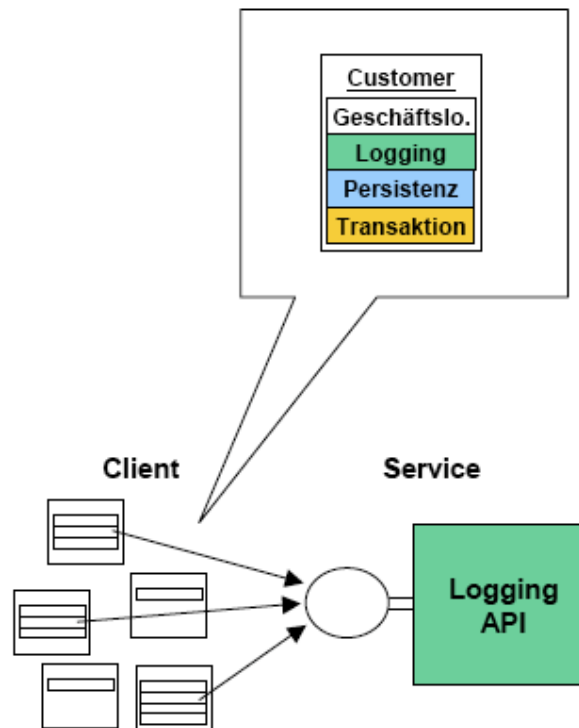
Vorteile

Anwendungen

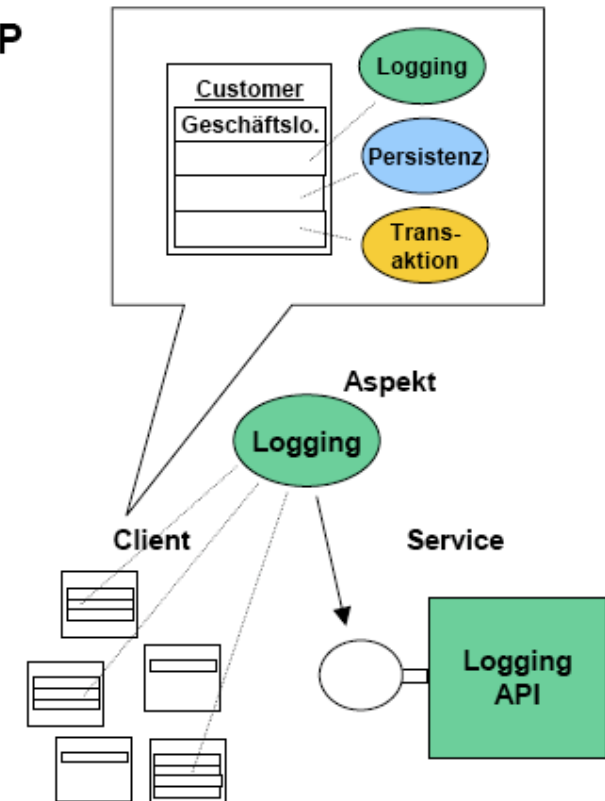
Entwicklungsstand

Quellen

**OO**



**AOP**





## 5. Vorgehen beim AOP Join Point Modell

Inhalt

Motivation

Problemstellung

Crosscutting  
Concerns

Was ist AOP?

Vorgehen bei AOP

Arten von AOP

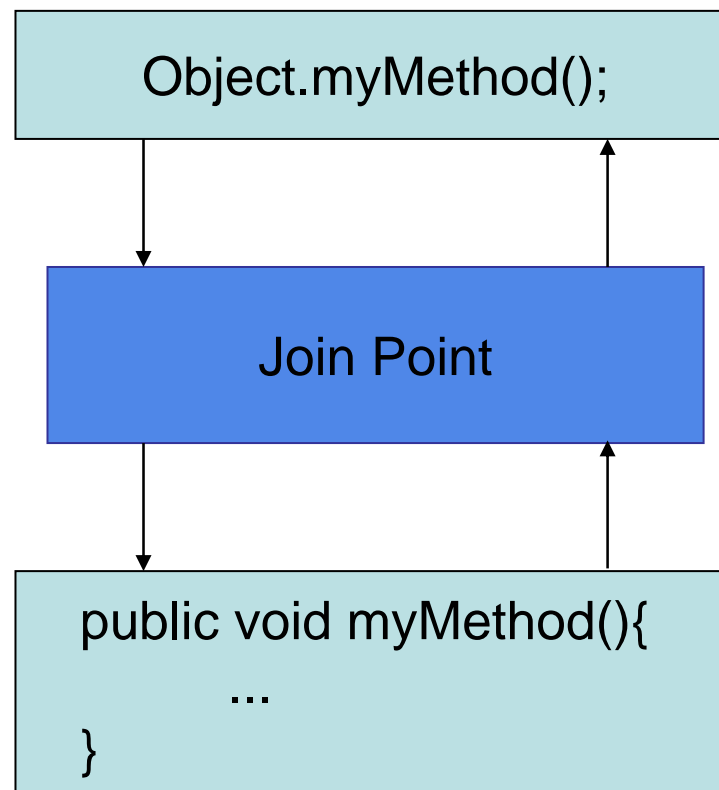
Vorteile

Anwendungen

Entwicklungsstand

Quellen

"**Wenn** Ereignis(JoinPoint) eintritt, **dann** führe Aktion(Advice) aus"





## 5. Vorgehen beim AOP Weitere Begriffe

Inhalt

Motivation

Problemstellung

Crosscutting  
Concerns

Was ist AOP?

Vorgehen bei AOP

Arten von AOP

Vorteile

Anwendungen

Entwicklungsstand

Quellen

- Pointcuts

- Zusammenfassung eines oder mehrerer Join Points
- Filter für Join Points
- Verwendet zusätzliche Infos wie z.B. Variablenbelegung, Parameter eines Methodenaufrufs

- Advices

- Bestimmt was an einem Pointcut getan werden soll
- Enthält Programmcode, der ausgeführt werden soll
- bestimmt an welcher Stelle eines Pointcuts eingegriffen werden soll
  - before
  - after
  - around



# 5. Vorgehen beim AOP

## Entwicklungsphase OO und AOP

Inhalt

Motivation

Problemstellung

Crosscutting  
Concerns

Was ist AOP?

Vorgehen bei AOP

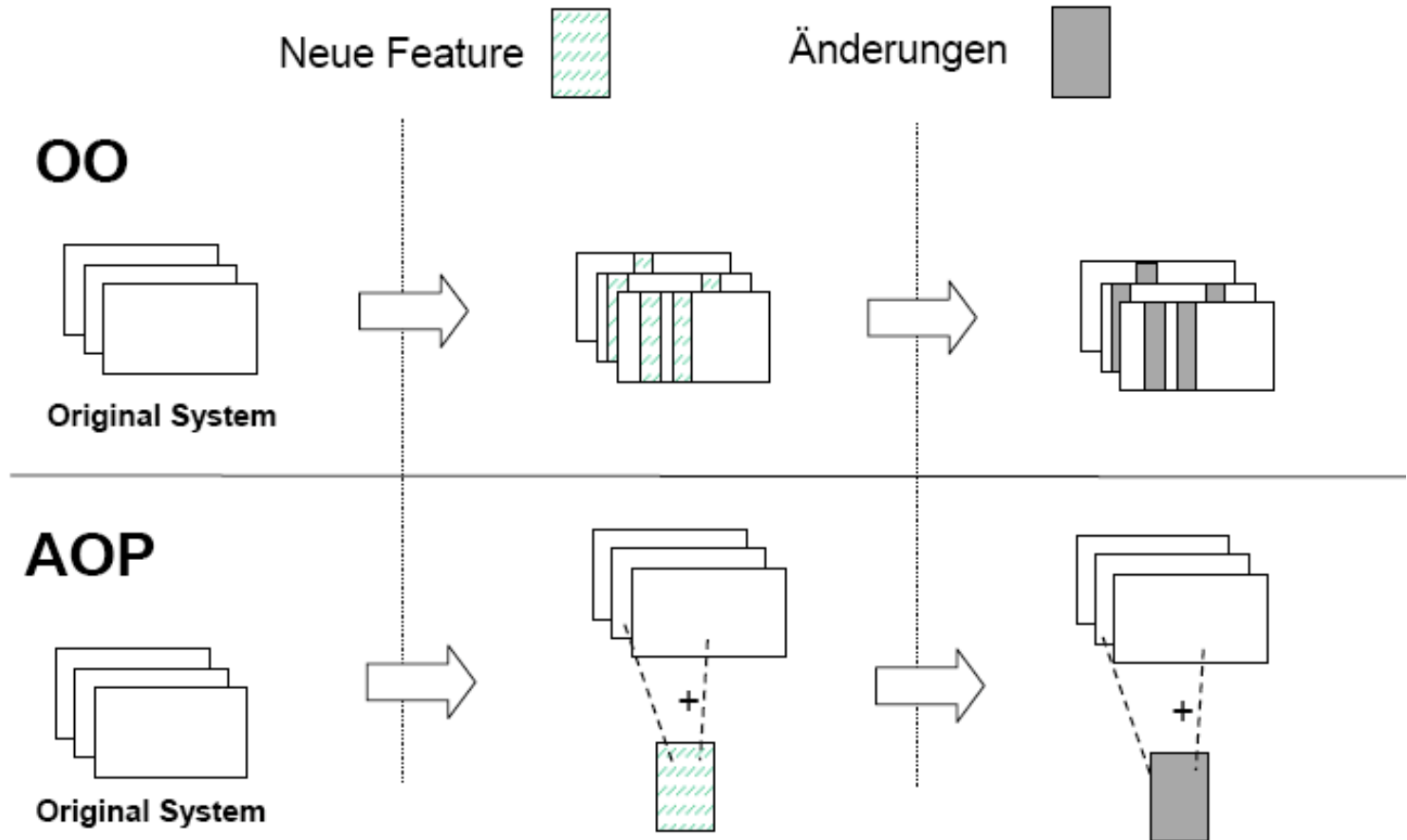
Arten von AOP

Vorteile

Anwendungen

Entwicklungsstand

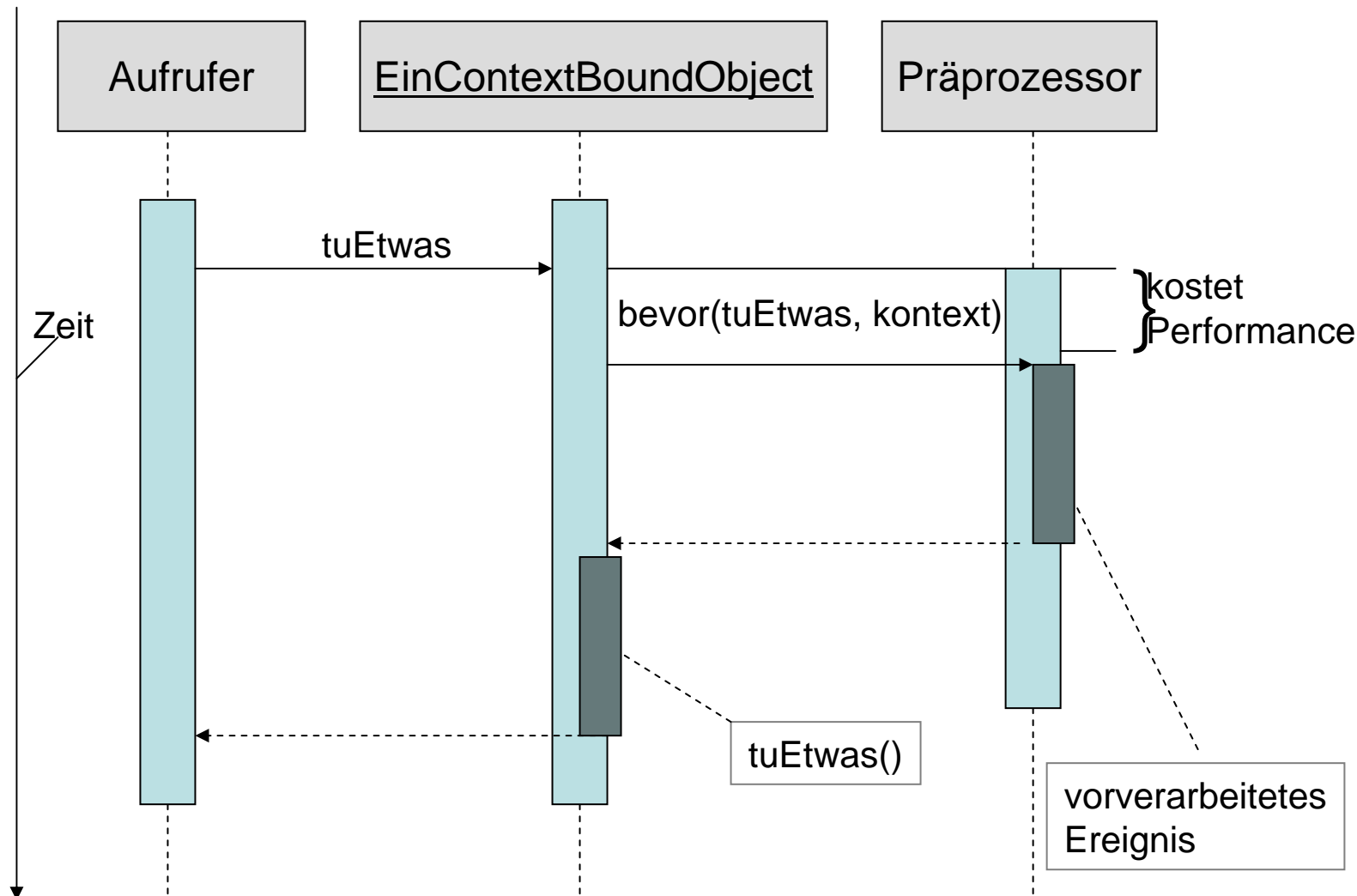
Quellen





## 6. Arten von AOP Proxies und Kontexte

Inhalt
Motivation
Problemstellung
Crosscutting Concerns
Was ist AOP?
Vorgehen bei AOP
Arten von AOP
Vorteile
Anwendungen
Entwicklungsstand
Quellen

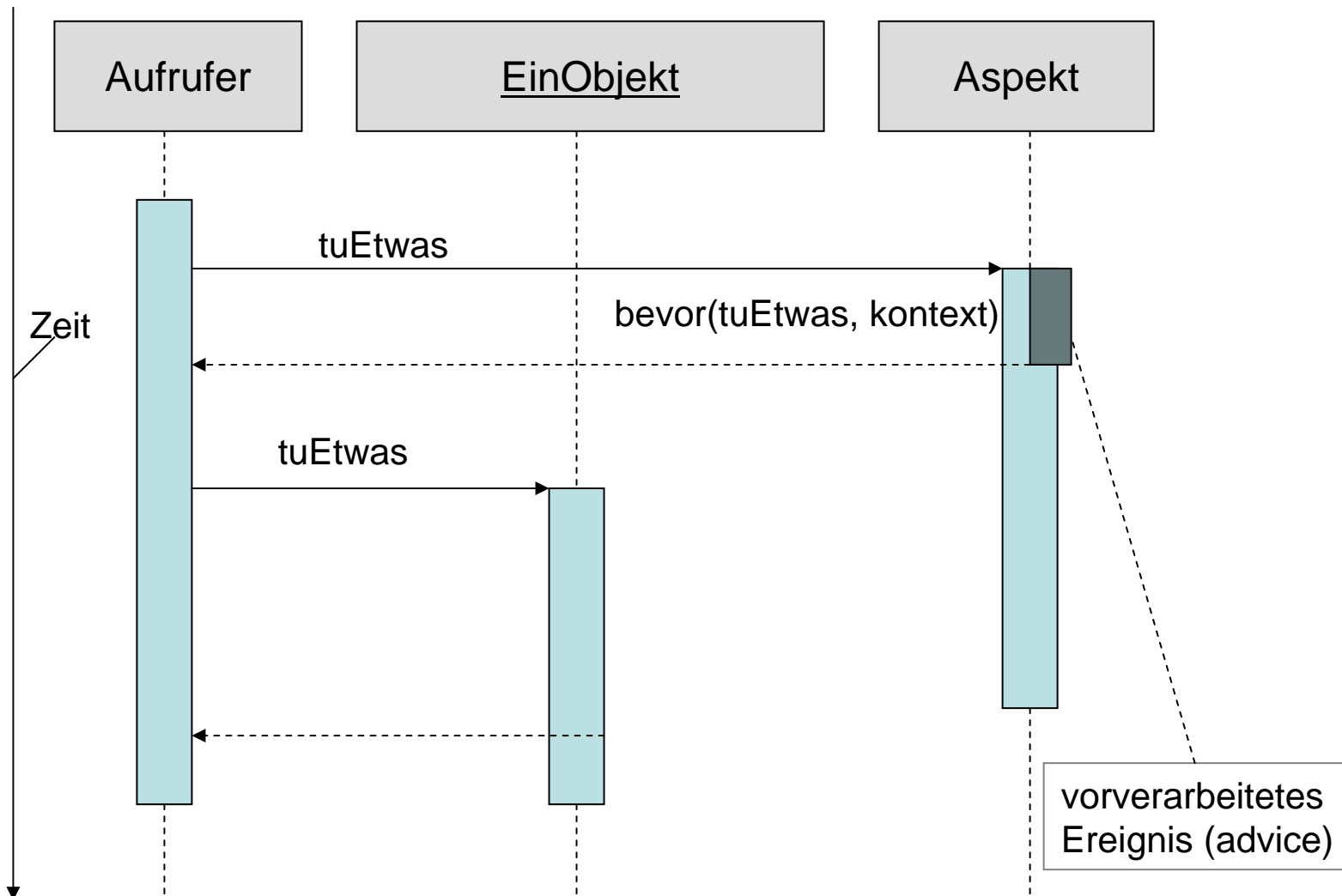






## 6. Arten von AOP Webe-basierte Ansätze

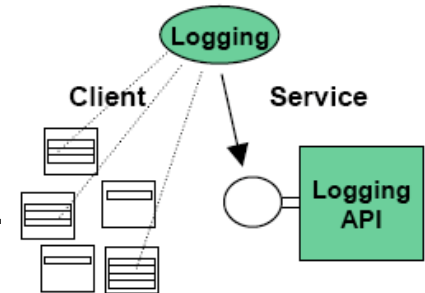
Inhalt
Motivation
Problemstellung
Crosscutting Concerns
Was ist AOP?
Vorgehen bei AOP
Arten von AOP
Vorteile
Anwendungen
Entwicklungsstand
Quellen



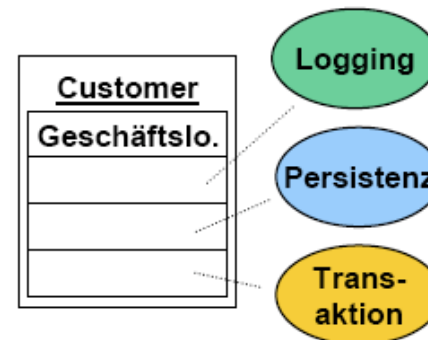
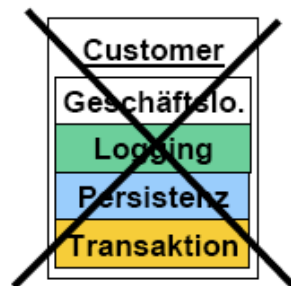


## 7. Vorteile durch AOP

- Stärkere Modularisierung:
  - Zentrale Entwicklung/Wartung  
Keine Code-Verstreung.
  - Weniger Coderedundanz, kürzerer Code.
  - Saubere, systemweit einheitliche Umsetzung
  - Unternehmensweite Richtlinien besser umsetzbar



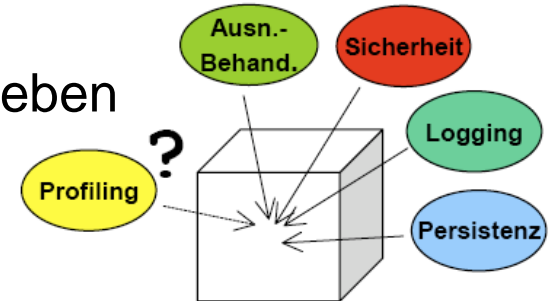
- Klarere Trennung der Verantwortung:
  - Ein Modul, eine Verantwortung. Keine Code-Vermischung.
  - Verständlicher Quellcode(Geschäftslogik, Technische Details)



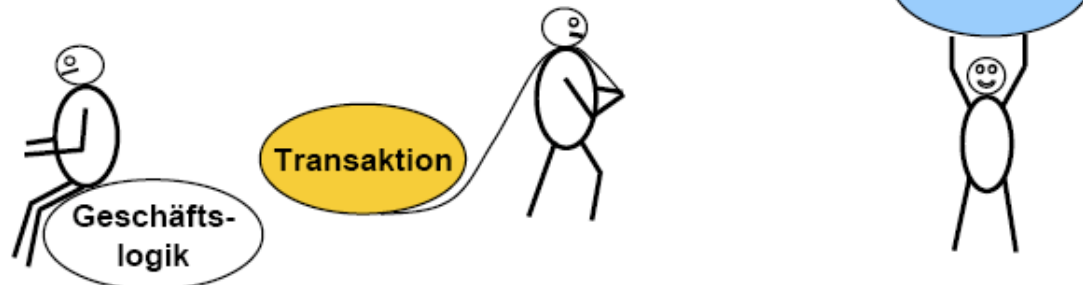


## 7. Vorteile durch AOP

- Stärkere Wiederverwendung:
  - Lose gekoppelte Module
  - System konfigurieren. Module Einweben und Herausnehmen (z.B. Profiling).
  - Aspekt-Bibliothek



- Höhere Produktivität:
  - Entwickler konzentriert sich auf seinen Expertenbereich
  - Höhere Code-Qualität
  - Unabhängige Module parallel entwickeln
  - Eigenen Entwicklungszyklus pro Modul





## 8. Anwendungen für AOP

Inhalt

Motivation

Problemstellung

Crosscutting  
Concerns

Was ist AOP?

Vorgehen bei AOP

Arten von AOP

Vorteile

Anwendungen

Entwicklungsstand

Quellen

- Sicherheitsüberprüfungen,
- Persistierung von Objekten,
- Synchronisation von Threads,
- Fehlerbehandlung,
- Optimierungen und optimale Ausnutzung von Ressourcen (z.B. Netzwerkbandbreite)
- transparentes Caching von Daten.



## 9. Entwicklungsstand von AOP

Inhalt

Motivation

Problemstellung

Crosscutting  
Concerns

Was ist AOP?

Vorgehen bei AOP

Arten von AOP

Vorteile

Anwendungen

Entwicklungsstand

Quellen

- Steckt heute noch in Kinderschuhen
- Es ist strittig, ob und wie AOP auf Objekt-Ebene zu realisieren ist.
- Zahlreiche bestimmte Erfahrungen müssen erst noch gemacht werden
  - Klassische Entwurfsmuster
  - Refaktorisierungen
  - Fallstricke
- Man geht davon aus, dass aspektorientierte Sprachen sich auf Basis von Java und C# entwickeln werden



## 10. Quellen

Inhalt

Motivation

Problemstellung

Crosscutting  
Concerns

Was ist AOP?

Vorgehen bei AOP

Arten von AOP

Vorteile

Anwendungen

Entwicklungsstand

Quellen

- [www.castleproject.org](http://www.castleproject.org) (Aspect#)
- [www.springframework.net](http://www.springframework.net) (Spring.NET)
- [www.wikipedia.de](http://www.wikipedia.de) (Suchwort: Apektororientierte Programmierung)
- [www.objektspektrum.de](http://www.objektspektrum.de)
- [www.uni-koblenz.de](http://www.uni-koblenz.de) (Post-Objekt-Orientierte Ansätze zur Softwareentwicklung)
- [wwwse.fhs-hagenberg.ac.at](http://wwwse.fhs-hagenberg.ac.at) (AOP.NET)
- [users.informatik.haw-hamburg.de/~raasch/akot-aop.PDF](http://users.informatik.haw-hamburg.de/~raasch/akot-aop.PDF)



# Wozu ein neuer Ansatz?

Inhalt

Motivation

Problemstellung

Crosscutting  
Concerns

Was ist AOP?

Vorgehen bei AOP

Arten von AOP

Vorteile

Anwendungen

Entwicklungsstand

Quellen

Braucht man aspekt-orientierte Programmierung?

- Nein, alles ist mit Objekten beschreibbar!

Braucht man objekt-orientierte Programmierung?

- Nein, alles ist mit Funktionen modellierbar!

Braucht man Funktionen?

- Nein, alles ist mit Sprüngen modellierbar!

etc. etc.