

# Advanced Software Engineering

## Anti-Patterns

## Agenda

### Einführung

- Motivation
- Was sind Anti-Patterns?

### Anti-Pattern Reference Model

- Hauptursachen
- Einfluss
- Software Design-Level Model

### Templates

- Pattern Templates
- Anti-Pattern Templates

### Anti-Patterns

# Einführung

## Einführung | Motivation

## Chaos Study

- Erfolgs- und Misserfolgsfaktoren von IT Projekten
  - **Typ 1 - Projekt erfolgreich abgeschlossen:** *Das Projekt wurde rechtzeitig, ohne Kostenüberschreitung und mit dem ursprünglich geforderten Funktionsumfang abgeschlossen*
  - **Typ 2 - Projekt teilweise erfolgreich:** *Das Projekt wurde abgeschlossen, es kam jedoch zu Kosten- und/oder Zeitüberschreitungen oder es wurde nicht der vollständige geplante Funktionsumfang erreicht.*
  - **Typ 3 - Projekt nicht erfolgreich:** *Das Projekt wurde abgebrochen.*

## Chaos Study 2011

- Typ 1 - Projekt erfolgreich abgeschlossen: 34%
- Typ 2 - Projekt teilweise erfolgreich: 51%
- Typ 3 - Projekt nicht erfolgreich: 15%

## Warum beschäftigen wir uns mit Anti-Patterns?

- Anti-Patterns sind überall um uns herum
  - Wesentlich mehr Software Projekte scheitern als gelingen
  - Schlechte Software Designs, Entscheidungen und Projekte sind verbreiteter als gute

## Einführung | Motivation

- Anti-Patterns erzählen die Wahrheit über die Software Industrie
- Kommerzielle Software Technologie ist geplagt von Defekten, Widersprüchen, falschen Versprechungen und Anti-Patterns
- Anti-Patters dokumentieren die essentiellen Informationen daraus

Einführung | **Motivation**

- Anti-Patterns erzählen die Realität von Software Projekten
  - Die meisten Software Projekte sind chaotisch, unvorhersehbar und gefährlich
  - Anti-Patterns erklären, wie Software Projekte wirklich funktionieren und wie man die vermeidbaren Konsequenzen managt



## Einführung | Motivation

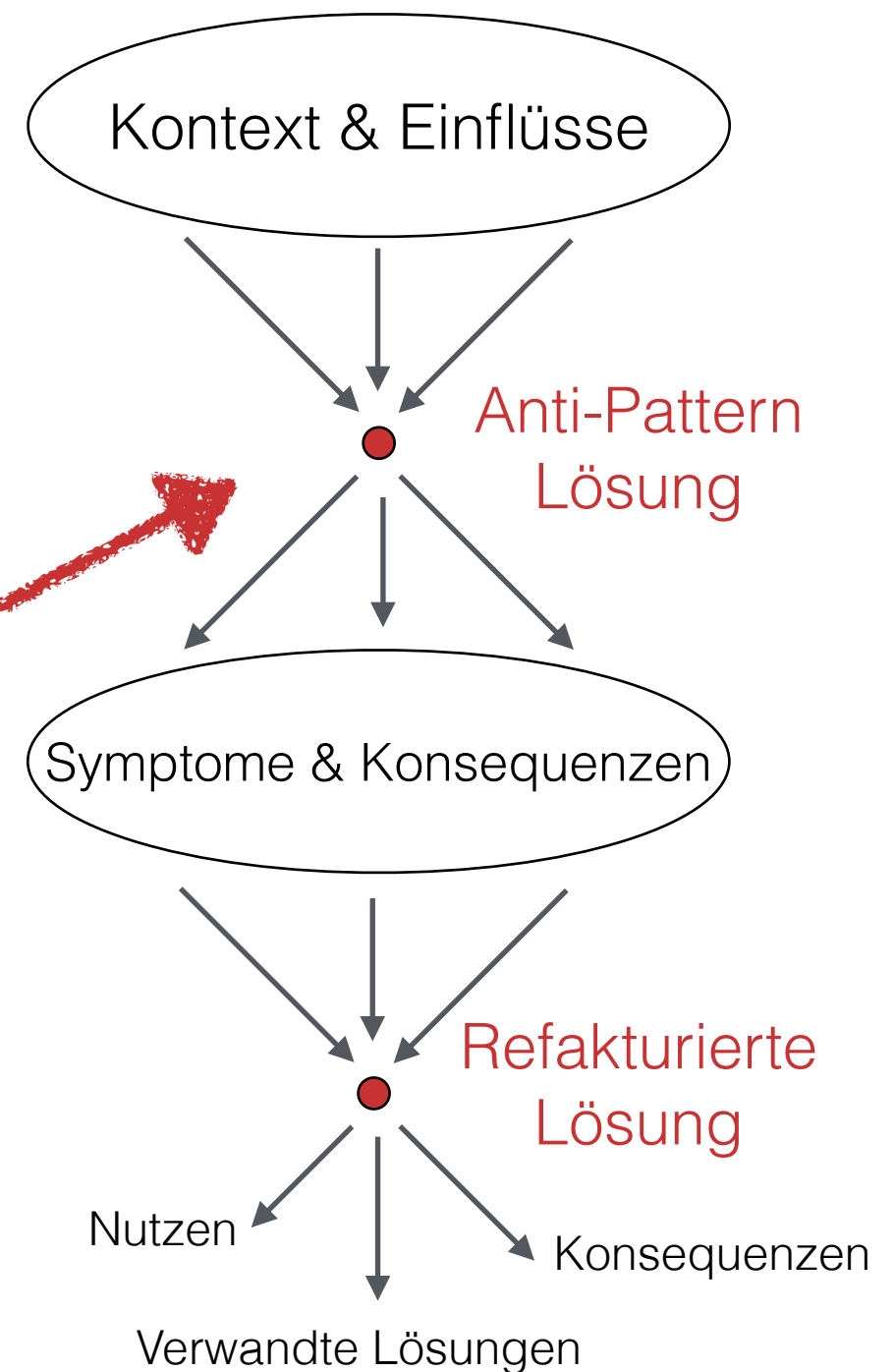
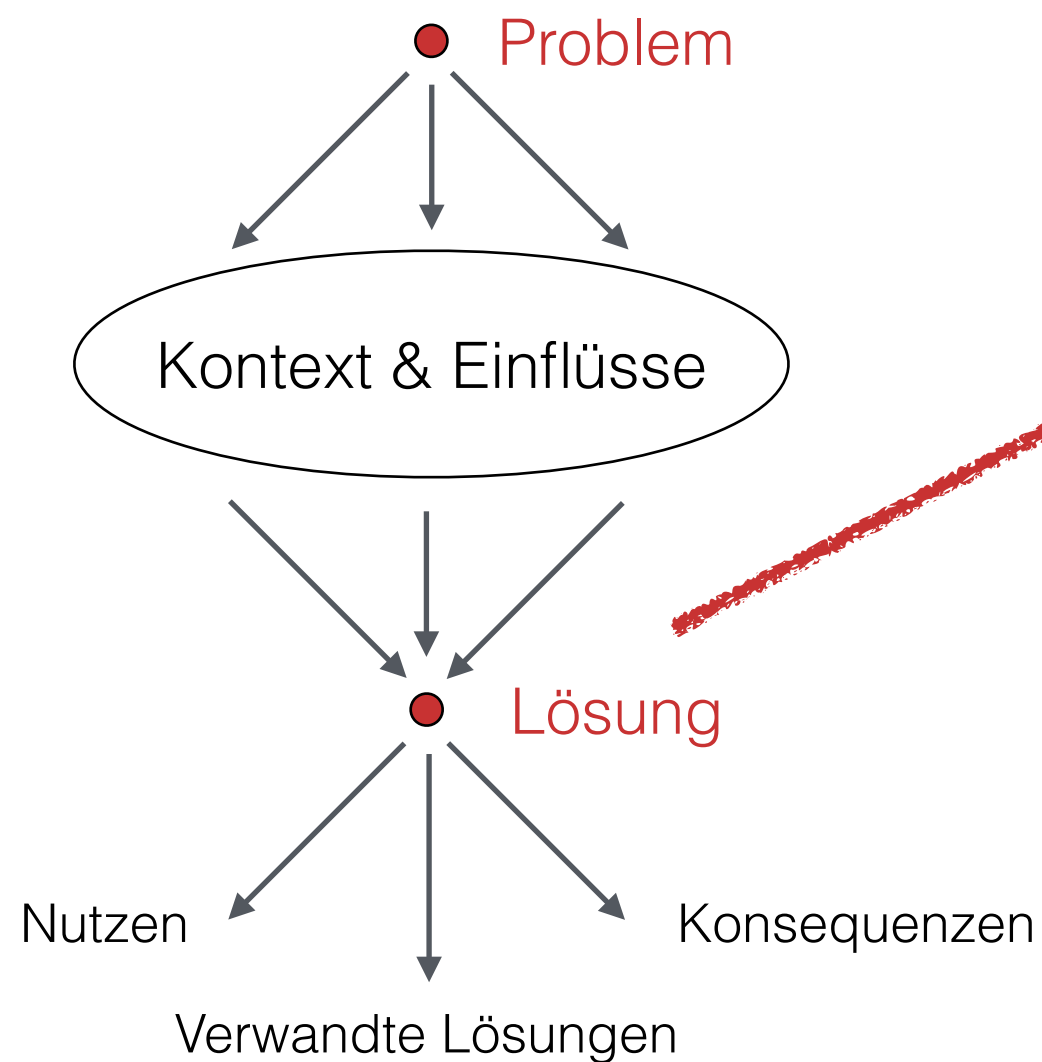
- Anti-Patterns sind notwendig für Change Management
  - Anti-Patterns etablieren eine klare Definition negativer Software Praktiken
  - Anti-Patterns sind nützliche Beschreibungen von Praktiken, die eine Organisation ändern will
- Anti-Patterns sind eine effektivere Form von Design Patterns

## Einführung | Was sind Anti-Patterns?

- Ein häufig anzutreffender schlechter Lösungsansatz für ein bestimmtes Problem
- Anti-Patterns helfen...
  - ...existierende Probleme zu erkennen
  - ...existierende Probleme zu lösen
  - ...die Wiederholung dieser Probleme zu vermeiden

Einführung | Was sind Anti-Patterns?

# Patterns und Anti-Patterns



## Einführung | Was sind Anti-Patterns?

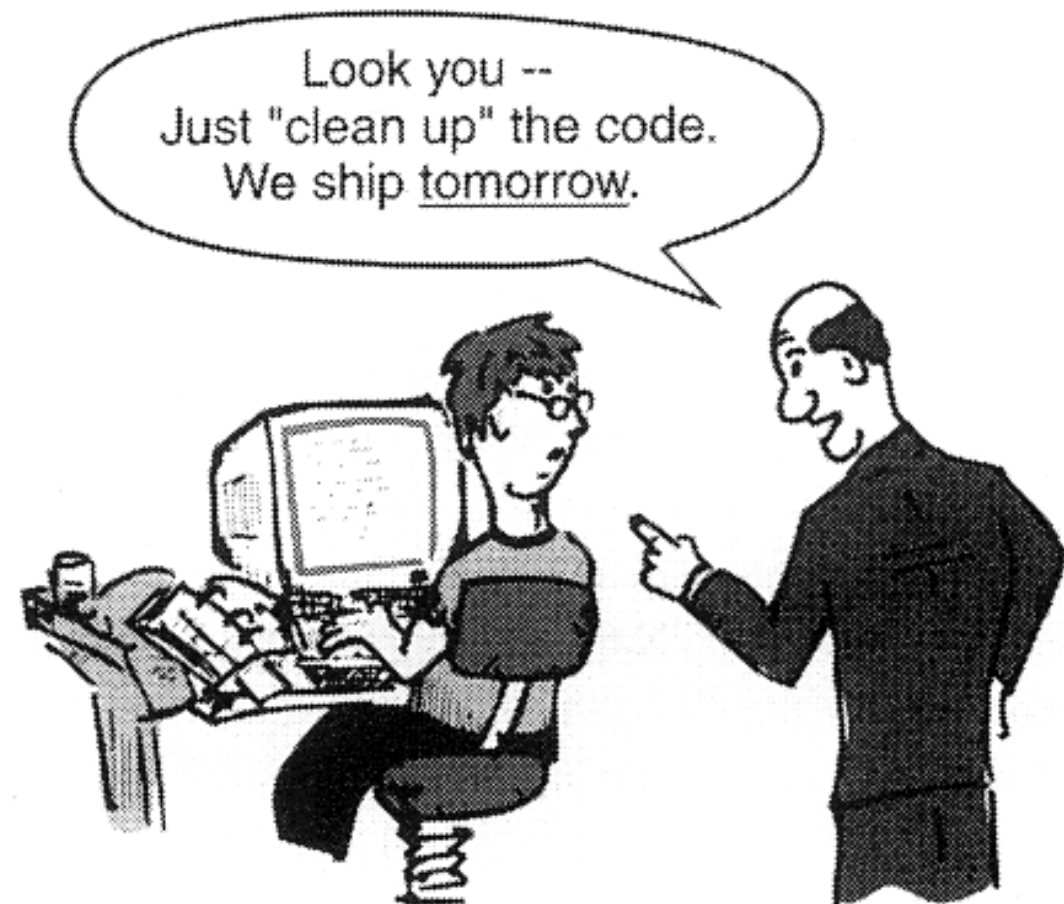
- Kategorisierung
  - Management
  - Software Architektur
  - Software Entwicklung (Programmierung)

# Anti-Pattern Reference Model

## Reference Model | Hauptursachen

### Haste - Hast/Eile

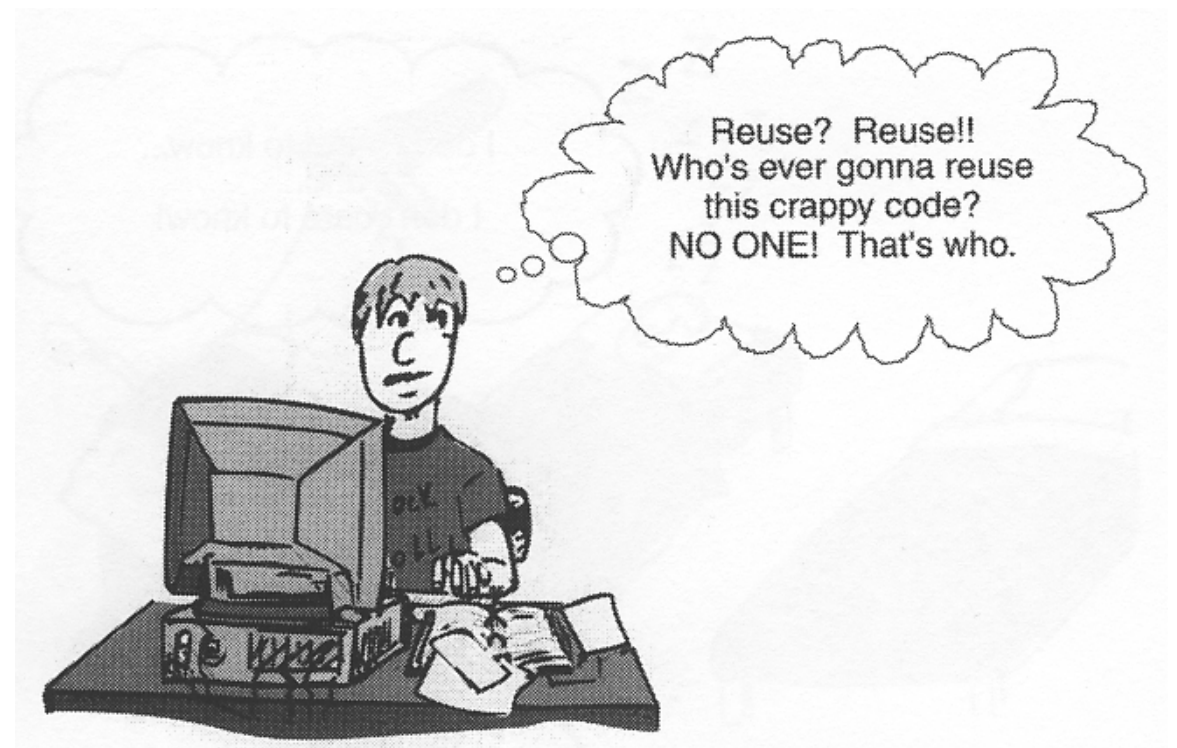
- Vorschnelle Entscheidungen führen zu Kompromissen in der Software Qualität
- Zeitpläne und Stress - Projektmanager werden darauf getrimmt Budgets und Zeiten einzuhalten
- Werden Deadlines überschritten, wird alles „Lauffähige“ akzeptiert, egal in welcher Qualität
- Hauptopfer: Tests! (*Unit Test Coverage, Integrationstests, Regression Tests*)



**Haste makes waste!**

## Apathy - Gleichgültigkeit

- Gleichgültigkeit...
  - ...bekannte Probleme zu lösen
  - ...bzgl. objektorientierter Strukturierung (Key-Aspekt Objektorientierter Architektur)
- Subsystem Changes bewirken auch Core Changes, was nie gemacht wird.



**Apathy is the worst form of flattery!**

## Reference Model | Hauptursachen

### Narrow-Mindedness - Engstirnigkeit

- Ablehnen praktischer Lösungen, die bereits als effektiv bekannt sind
- „Reinventing the wheel“



**The Unteachables!**



## Reference Model | Hauptursachen

### Sloth - Faulheit

- Träger Entwickler/Manager trifft schlechte Entscheidungen aufgrund von „einfachen“ Antworten

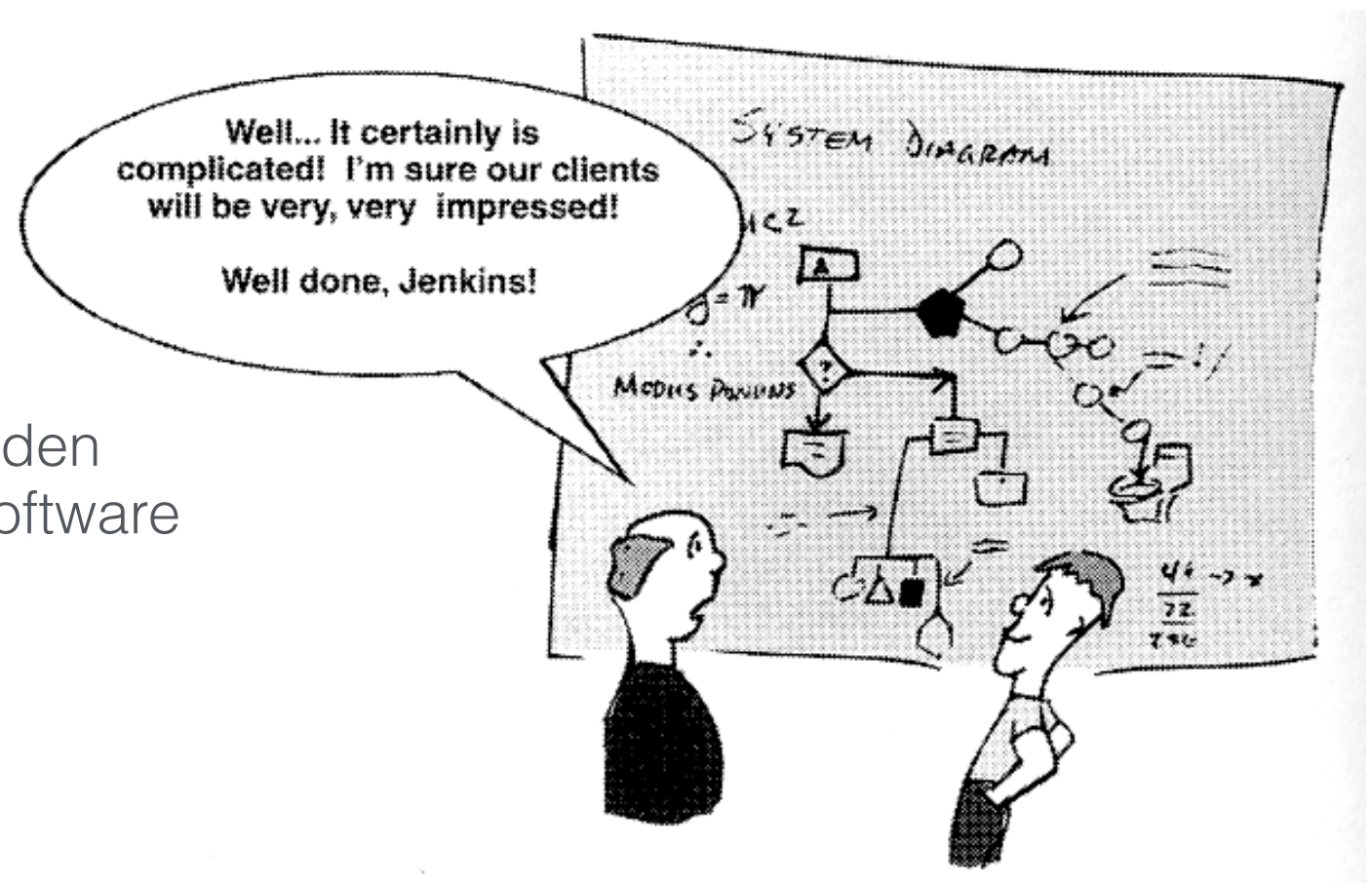


**Sloth usually ends with sudden clarity!**

## Reference Model | Hauptursachen

### Avarice - Gier

- Bezogen auf Geld...
- ...führt Gier zu unpassenden Entscheidungen in der Software Entwicklung
- Architektonische Gier...
- ...führt zu übertrieben detailliert modellierten Komponenten
- ...teuer in Test, Wartung, Weiterentwicklung, Dokumentation

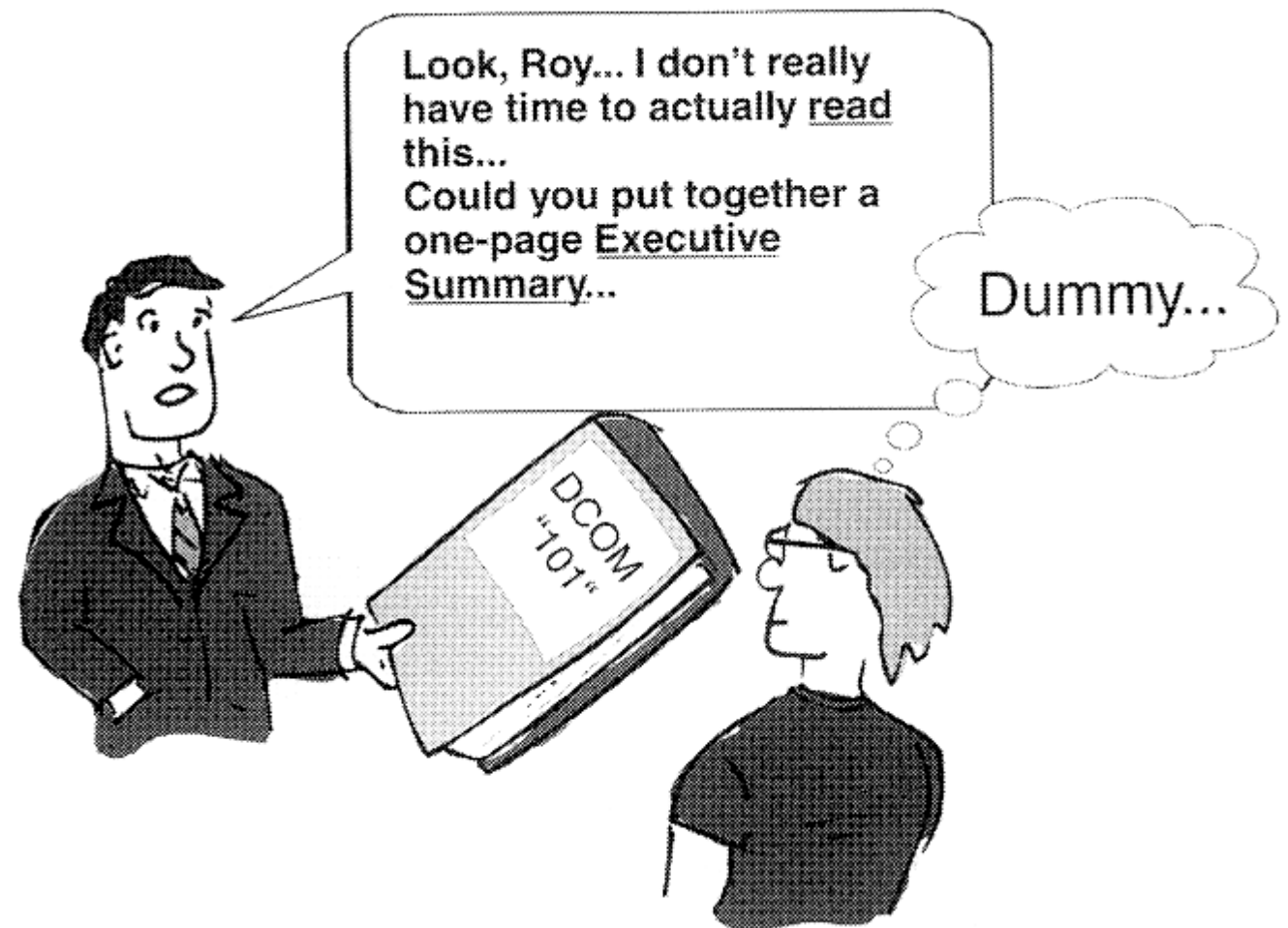


**Addicted to complexity!**

## Reference Model | Hauptursachen

### Ignorance - Ignoranz

- Intellektuelle Faulheit
- Hält Leute dumm



**Some people are too ignorant to change!**

## Reference Model | Hauptursachen

### Pride - Stolz

- Not-invented-here-Syndrom
- Nichtbeachtung von bereits existierendem Wissen
- Wiederverwendung von packages, frameworks, wrapped legacy code



**Pride goeth before the fall!**



Reference Model | Einfluss/Auswirkung

## Management of...

- ...Complexity
- ...Change
- ...IT Ressources
- ...Technology Transfer

## Reference Model | Software Design Level Model

### Micro-Levels:

- Idiom (Objekte & Klassen)
- Micro-Architecture

### Macro-Levels:

- Framework
- Application
- System
- Enterprise
- Global/Industry

# Templates

## Templates | Pattern Templates

- Templates stellen sicher, dass wichtige Fragen zu jedem Pattern beantwortet werden. (*Pattern Language, Pattern Catalog, Pattern System*)
- Ohne Template lediglich jemandes unstrukturierte Prosa.
  - „*Es ist ein Pattern, weil ich ein Experte bin und es sage!*“
- Konsistente rhetorische Struktur. Gut durchdachte Logik der Pattern Beschreibung.
- Jede Sektion beantwortet Schlüsselfragen zum jeweiligen Pattern.
- „Rule of three“ - 3 existierende Anwendungen



## Templates | Pattern Templates

### Degenerate Pattern Form

#### Technical Discussion

- Was hat der Autor über die Konzepte zu sagen?
- Lediglich Content Sektion
- Ohne erkennbare Template Form
- Basiert ausschließlich auf Aussagen des Autors
- Leser muss analysieren
  - Was ist das Problem? In welchem Kontext besteht es?
  - Was sind die Konsequenzen? Was ist die Lösung?

Templates | Pattern Templates

## Micro Pattern Template

### Name

- Wie soll das Pattern von Anwendern genannt werden?

### Problem

- Was motiviert uns dieses Pattern anzuwenden?

### Solution

- Wie lösen wir das Problem?

## Templates | Pattern Templates

### **Inductive Mini Pattern Template** (*Fokus auf Anwendbarkeit*)

#### Name

- Wie soll das Pattern von Anwendern genannt werden?

#### Context

- In welchem Kontext soll das Pattern angewendet werden?

#### Forces

- Welche Designs/Szenarien müssen ausbalanciert werden?

#### Solution

- Wie lösen wir das Problem?

## Templates | Pattern Templates

### **Deductive Mini Pattern Template** (*Fokus auf Ergebnis*)

#### Name

- Wie soll das Pattern von Anwendern genannt werden?

#### Problem

- Was motiviert uns dieses Pattern anzuwenden?

#### Solution

- Wie lösen wir das Problem?

#### Benefits

- Was sind die potentiellen positiven Resultate durch die Anwendung?

#### Consequences

- Was sind die potentiellen Unzulänglichkeiten und Konsequenzen?

## Templates | Pattern Templates

### Gang-of-Four Pattern

#### Pattern Name

- Wie wird das Pattern genannt?

#### Pattern Classification

- Erzeugend, Strukturierend oder Verhaltensweise

#### Intent

- Welches Problem löst das Pattern?

#### Also Known As

- Unter welchen anderen Namen ist das Pattern bekannt?

## Templates | Pattern Templates

### ...Gang-of-Four Pattern

#### Motivation

- Was ist ein Bsp.-Szenario für die Anwendung des Patterns?

#### Applicability

- Wann wird das Pattern angewandt?

#### Structure

- Klassendiagramme dieses Patterns

#### Participants

- Welche Objekte sind in diesem Pattern beteiligt?

## Templates | Pattern Templates

### ...Gang-of-Four Pattern

#### Collaborations

- Wie interagieren die Objekte miteinander?

#### Consequences

- Welche potentiellen Abstriche müssen bei Anwendung gemacht werden?

#### Implementation

- Welche Techniken oder Aspekte werden angewendet?

#### Sample Code

- Bsp. des Pattern als Quelltext

## Templates | Pattern Templates

### ...Gang-of-Four Pattern

#### Known Uses

- Beispiele von reellen Systemen, die das Pattern verwenden

#### Related Patterns

- Welche Anderen Patterns stehen mit diesem Pattern in Beziehung?



## Templates | Anti-Pattern Templates

- Haben 2 Lösungen
  - Starten mit einer bestehenden Lösung mit problematischen Auswirkungen (*Symptome*)
  - Führen zu einer refakturierten Lösung (*Refactored Solution*)
- Für die bestehende Ausgangs-Lösung gilt die „rule of three“
- Refactored Solution ist eine Best Practice Solution

Templates | Anti-Pattern Templates

## Pseudo Anti-Pattern Template

Name

Problem

- Vergleiche Degenerate Pattern Form

Templates | Anti-Pattern Templates

## Mini Anti-Pattern Template

Name

AntiPattern Problem

- Was ist die wiederkehrende Lösung, die negative Auswirkungen zur Folge hat?

Refactored Solution

- Wie können wir das Problem vermeiden, minimieren oder refakturieren?

Templates | Anti-Pattern Templates

## Full Anti-Pattern Template

Name

Also Known As

Most Frequent Scale

- Micro Levels: Idiom (Objekte & Klassen), Micro-Architecture
- Macro Levels: Framework, Application, System, Enterprise, Global/Industry

Templates | Anti-Pattern Templates

## ...Full Anti-Pattern Template

Refactored Solution Name

Refactored Solution Type

- Software - Neue Software durch die Lösung
- Technology - Bringt neue Technologien oder Produkte mit
- Process - Bringt neue Prozesse mit
- Role - Einem Individuum oder einer Gruppe müssen Zuständigkeiten zugeteilt werden

Templates | Anti-Pattern Templates

## ...Full Anti-Pattern Template

### Root Causes

- Haste
- Apathy
- Narrow Mindness
- Sloth
- Ignorance
- Pride

Templates | Anti-Pattern Templates

## ...Full Anti-Pattern Template

### Unbalanced Forces

- Management of Complexity
- Management of Change
- Management of IT Ressourcen
- Management of Technology Transfer

### Anecdotal Evidence

- Oft gehörte Phrasen oder lustiges (Bild-)Material

Templates | Anti-Pattern Templates

## ...Full Anti-Pattern Template

### Background (*optional*)

- Weitere Beispiele, wo die Probleme auftreten.
- Allgemeine nützliche oder interessante Hintergrundinformationen

### General Form of this Anti-Pattern

- Oftmals ein Diagramm über die Charakteristik des Anti-Patterns mit zugehöriger Beschreibung.



## Templates | Anti-Pattern Templates

### ...Full Anti-Pattern Template

#### Symptoms and Consequences

- Liste (Bullet Points) mit Symptomen und Konsequenzen die mit dem Anti-Pattern zusammen hängen

#### Typical Causes

- Liste (Bullet Points) mit eindeutigen Folgen durch das Anti-Pattern

#### Known Exceptions

- Die beschriebenen Verhaltensweisen und Prozesse sind nicht immer falsch.
- Ausnahmen, in welchen Fällen das Anti-Pattern nicht gilt.

Templates | Anti-Pattern Templates

## ...Full Anti-Pattern Template

### Refactored Solution

- Erklärt die refakturierte Lösung, welche die Auswirkungen beseitigt
- Strukturiert in Lösungsschritten

### Variations

- Listet alternative Lösungen auf

Templates | Anti-Pattern Templates

## ...Full Anti-Pattern Template

### Example

- Demonstriert die Anwendung des Lösungsweges
- Typischerweise enthalten: Problem-Beschreibung, Lösungs-Diagramm, Lösungs-Beschreibung

### Related Solutions

- Auflistung aller Anti-Patterns die zu diesem in Beziehung stehen.
- Erläuterung der Unterschiede

Templates | Anti-Pattern Templates

## ...Full Anti-Pattern Template

### Applicability to Other Viewpoints and Scales

- Beschreibt in welcher Weise das Anti-Pattern „Management“, „Architektur“ und Softwareentwicklung beeinflusst
- Was passiert wenn das Anti-Pattern auf unterschiedlichen Leveln angewandt wird?
- Wie effektiv löst es die betroffenen Auswirkungen des Problems?
- Welche neuen Auswirkungen werden eingeführt?

# Anti-Patterns

## Anti-Patterns | Software Development Anti-Patterns

### Bekannteste Anti-Patterns der Kategorie Softwareentwicklung:

- The Blob
- Continuous Obsolescence
- Lava Flow
- Ambiguous Viewpoint
- Functional Decomposition
- Poltergeists
- Boat Anchor
- Golden Hammer
- Dead End
- Spaghetti Code
- Input Kludge
- Walking through a Minefield
- Cut-and-Paste Programming
- Mushroom Management

## Anti-Patterns | Software Architecture Anti-Patterns

### Bekannteste Anti-Patterns der Kategorie Softwarearchitektur:

- Autogenerated Stovepipe
- Stovepipe Enterprise
- Jumble
- Stovepipe System
- Cover Your Assets
- Vendor Lock-In
- Wolf Ticket
- Architecture By Implication
- Warm Bodies
- Design By Committee
- Swiss Army Knife
- Reinvent the Wheel
- The Grand Old Duke of York

## Anti-Patterns | Software Projectmanagement Anti-Patterns

### Bekannteste Anti-Patterns der Kategorie Software PM:

- Blowhard Jamboree
- Analysis Paralysis
- Viewgraph Engineering
- Death By Planning
- Fear of Success
- Corncob
- Intellectual Violence
- Irrational Management
- Smoke and Mirrors
- Project MisManagement
- Throw It over the Wall
- Fire Drill
- The Feud
- E-Mail is Dangerous