



Technology Day / Online / 01.12.2021

Architecture Anti-Patterns

INNOQ

Agenda

- Vorgehen
- Was sind Anti-Pattern?
- Beispiele
 - Cargo Culting
 - Misapplied Genericity
 - Never Touch a Running System
- Ausblick



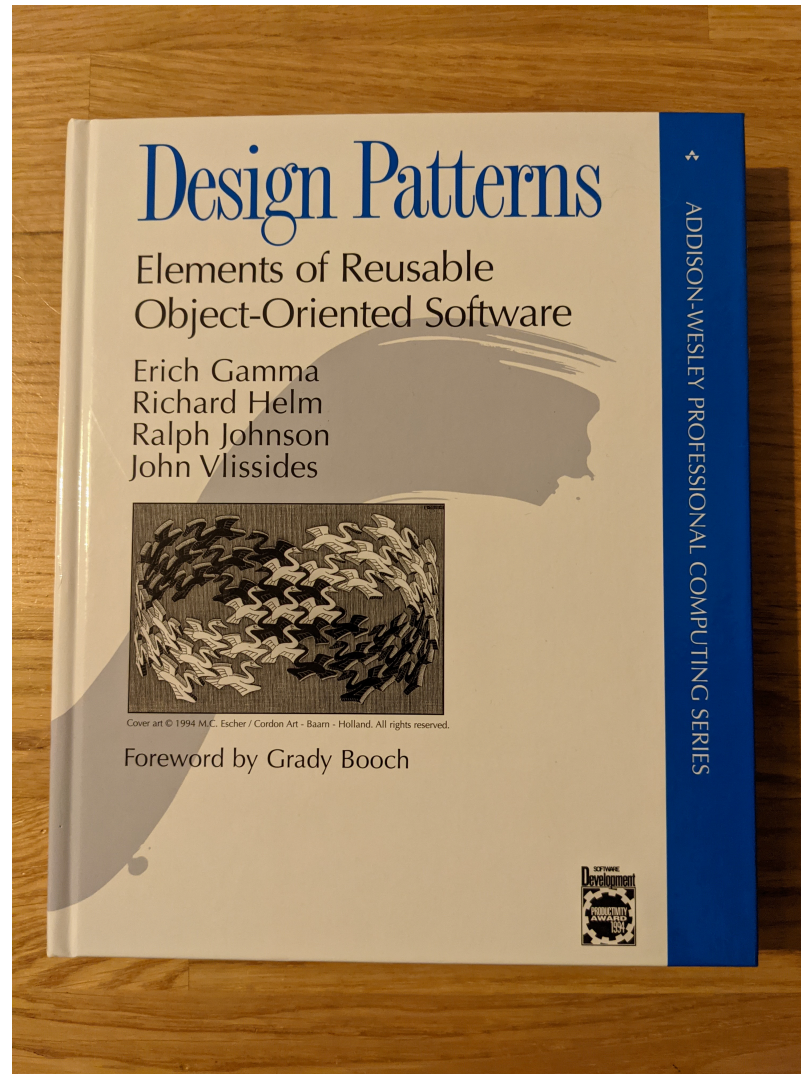
Mitmachen 🙋🏻 🙋🏼 🙋🏾

Bild: Felicia Buitenwerf [6]

Vorgehen

- Regelmäßiges Treffen von INNOQ
- Git Repository
 - Pattern
 - Use Cases
- Vorbereitung von einzelnen Pattern
- Diskussion in der größeren Runde
 - Erfahrungsaustausch
 - Wissensaufbau

Was sind Pattern?



Was sind Anti-Pattern

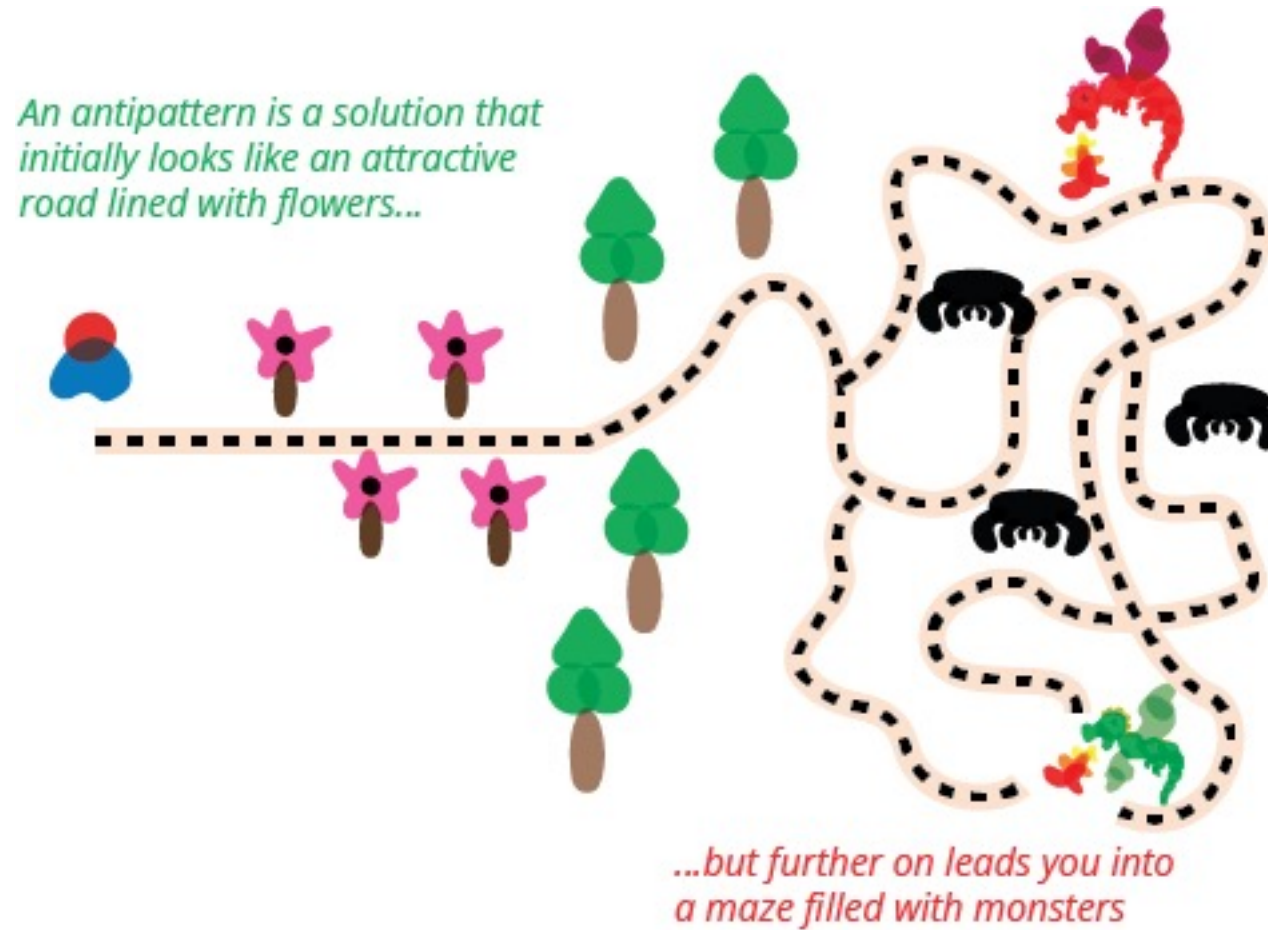
An antipattern is just like a pattern, except that instead of a solution it gives something that looks superficially like a solution but isn't one. – A. Koenig, *The Patterns Handbook* [1]

An AntiPattern is a literary form that describes a commonly occurring solution to a problem that generates decidedly negative consequences. – W. J. Brown et al, *Anti Patterns* [2]

... design anti-patterns are 'obvious, but wrong, solutions to recurring problems'. – D. Budgen, *Software design* [3]

... common approaches to solving recurring problems that prove to be ineffective. These approaches are called antipatterns. – S. W. Ambler, *Process patterns* [4]

Was sind Anti-Pattern



Bildquelle: Martin Fowler, martinfowler.com [5]

Pattern und Anti-Pattern Konzept

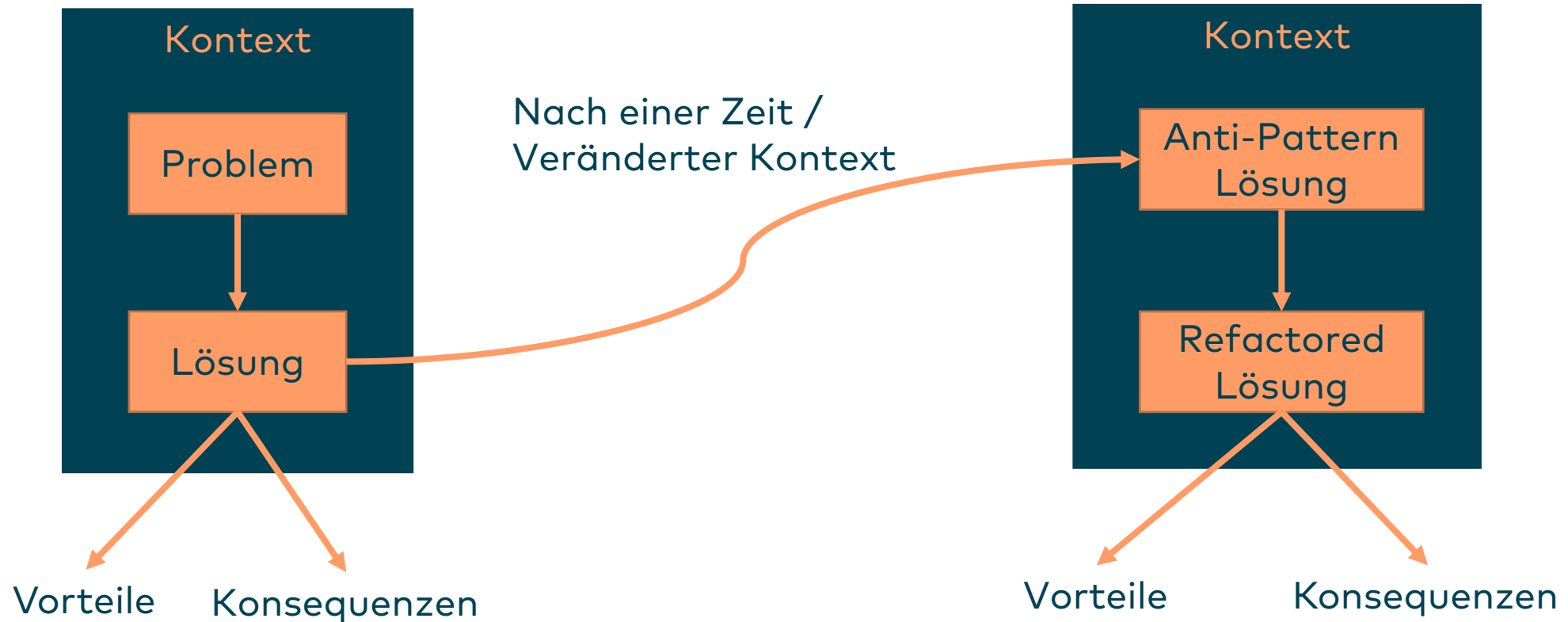


Bild nach: W. J. Brown et al, Anti Patterns [2]

Anti-Pattern Template

- Name
- Beschreibung
- Ursache
- Lösung + Vorbeugen

- Use-Cases / Beispiele
 - Hintergrund
 - Lösungsidee
 - Negativen Konsequenzen (Wende- bzw. Kipppunkt)

Cargo Culting

- Adopting processes, technologies or methodologies without understanding why and how they work
- Examples
 - Using microservices because Netflix, is using them
 - Implementing unsuitable processes for your company culture, like the Spotify model
 - BPMN engine for everything
- Why does this happen?
 - Adopting the new behavior takes so much time that the reflecting and adapting phases are skipped

Misapplied Genericity

- Building a solution that is so generic that it ends up satisfying no one
- Example
 - Super-generic framework
 - An application for 100+ countries
- Why does this happen?
 - „we could build this functionality in a generic way so that we can easily reuse it elsewhere in the future“
 - Developers like to solve very hard problems

Never Touch a Running System

- Fear of breaking existing critical or complex system
- People are discouraged from even trying to change the system
- Example
 - System written in a legacy language
 - Old system that does not get updated and has a lot of legacy libraries
- Why does this happen?
 - Developed by a project team and then given to a maintenance team
 - Head monopolies

Ausblick

- Weitere Pattern
 - Ziel bis Ende Jahres 10 Pattern
- Vorhandene Pattern verfeinern
- Dran bleiben

Danke! Fragen?



innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim
+49 2173 3366-0

Ohlauer Str. 43
10999 Berlin

Ludwigstr. 180E
63067 Offenbach

Kreuzstr. 16
80331 München

Hermannstrasse 13
20095 Hamburg

Erftstr. 15-17
50672 Köln

Königstorgaben 11
90402 Nürnberg

Quellen

- [1] - A. Koenig, The Patterns Handbook
- [2] - W. J. Brown et al, Anti Patterns
- [3] - D. Budgen, Software design
- [4] - S. W. Ambler, Process patterns
- [5] - Martin Fowler, martinfowler.com/bliki/AntiPattern.html
- [6] - Felicia Buitenwerf, unsplash.com/photos/Qs_Zkak27Jk