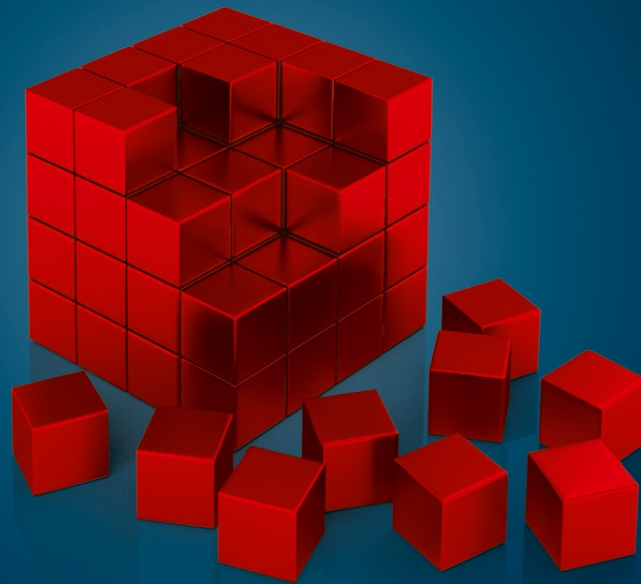




stefan TOTH

VORGEHENS-MUSTER FÜR SOFTWARE- ARCHITEKTUR

3. Auflage



KOMBINIERBARE PRAKTIKEN
IN ZEITEN VON AGILE UND LEAN



Im Internet: www.swamuster.de

HANSER



Bild 1.3 Kapitelstruktur

[Kapitel 2](#) vermittelt die inhaltliche Vision dieses Buchs und schafft wichtige Grundlagen für die weitere Lektüre. Es dient als Einstieg und Wegweiser für die 30 folgenden Vorgehensmuster.

Die Reihenfolge der Patternkapitel ist nicht zufällig gewählt. Ausgehend von Anforderungen und Problemstellungen ([Kapitel 3](#)), bespreche ich Vorgehensaspekte beim Treffen von Architekturentscheidungen ([Kapitel 4](#)) und unterstützende Praktiken für dynamische Zusammenarbeit ([Kapitel 5](#)). Schließlich verbinde ich Architekturideen und -anforderungen mit den tatsächlich beobachtbaren Eigenschaften des Systems ([Kapitel 6](#)). Sie können die Kapitel und die enthaltenen Muster gut von vorne nach hinten lesen, selbstverständlich ist aber auch eine nichtlineare Arbeit mit dem Buch möglich. Die Muster verweisen aufeinander und ermöglichen Ihnen Sprünge zu interessanten Ansatzpunkten.

Der zweite große Teil des Buchs zeigt, wie Sie Muster in unterschiedlichen Kontexten anwenden können bzw. welche Anknüpfungspunkte es zu Vorgehensmodellen, etablierten Rollenmodellen und agilen Frameworks gibt. In [Kapitel 7](#) fokussiere ich auf die allgemeine Etablierung von Vorgehensmustern in Projekten oder Unternehmen. Wie experimentieren Sie richtig mit den Ideen dieses Buchs? An welchen Stellen passen die Muster in agile Prozesse wie Scrum? Und welche Auswirkungen hat die

Musteranwendung auf die Rolle Architekt? [Kapitel 8](#) ist schließlich der agilen Skalierung gewidmet: Wie spielen die Ideen dieses Buchs mit agilen Skalierungsframeworks zusammen? Wie können große Entwicklergruppen agil an Architektur arbeiten? Und wie fügen sich evolutionäre Architekturansätze in die Welt der beschriebenen Vorgehensmuster?

1.6.2 Patterns lesen

Für die Beschreibung der Muster in diesem Buch habe ich den „Alexandrischen Stil“ gewählt. Diese von Christopher Alexander¹⁰ verwendete Form der Musterbeschreibung zeichnet sich vor allem durch ihre gute Lesbarkeit aus. Die einzelnen Teile des Musters sind nicht durch Überschriften getrennt, sondern lediglich optisch abgesetzt – durch die Verwendung von Bildern, Balken oder Kästen. Um den Lesefluss und die Lebendigkeit der Beschreibung weiter zu erhöhen, habe ich den Musterteil der „Forces“ oder „Einflüsse“ in die Musterbeschreibung integriert und kann so die Lösung direkt im Text motivieren. [Bild 1.4](#) zeigt die Teile eines Musters im Überblick.

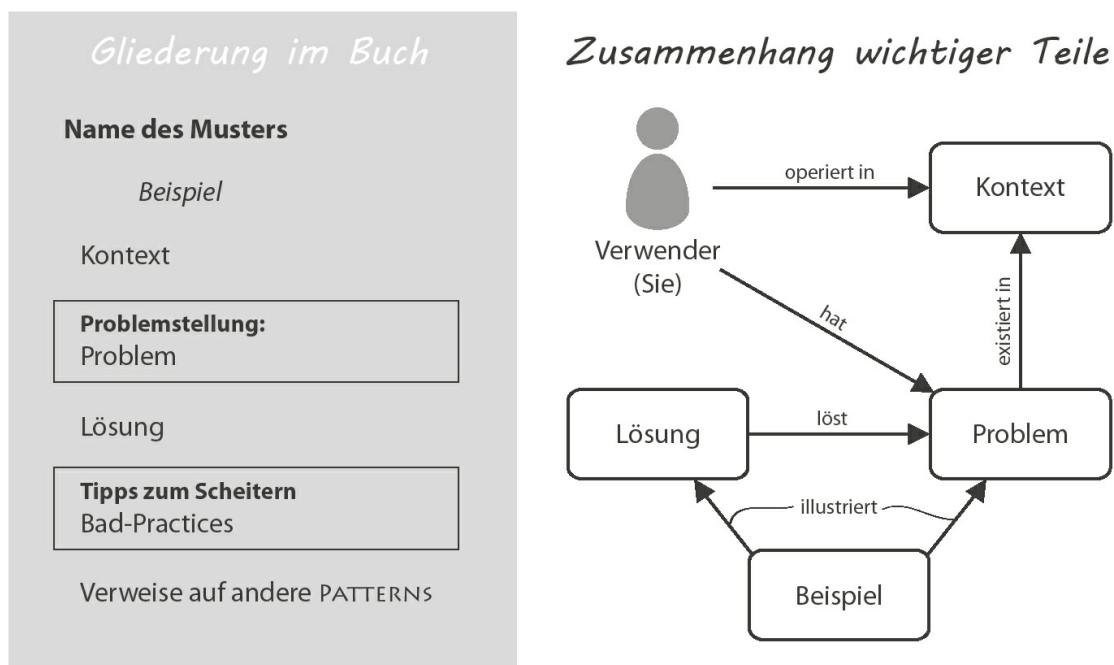


Bild 1.4 Aufbau der Muster in diesem Buch

Generell sollten die beschriebenen Muster gut von oben nach unten lesbar sein. Das Beispiel illustriert einen wichtigen Ausschnitt des Patterns und gibt Ihnen eine grobe

Idee. Danach setzt der Kontext die Bühne für das als Frage formulierte Problem. Der Hauptteil des Musters kümmert sich dann um die Lösung dieses Problems. In einem abschließenden Kasten finden Sie schlechte Anwendungsbeispiele des Musters – Anti-Patterns. Die Verweise zu anderen Mustern sind vor allem spannend, wenn Sie das Muster bei sich anwenden wollen. Sie können die Verweise auf verwandte Muster auch nutzen, um das Buch netzwerkartig von interessanten Mustern ausgehend zu lesen, anstatt streng linear vorzugehen.

1.6.3 Patterns anwenden

Die stumpfe Übernahme von in Mustern beschriebenen Praktiken wird Ihnen und Ihrem System nur bedingt helfen. Finden Sie interessante Ansätze und Ideen, können Sie gerne so schnell wie möglich loslegen. Um die Musteranwendung jedoch erfolgreich und zielführend zu gestalten, lesen Sie die Hinweise, die ich in [Kapitel 7](#) („Vorgehensmuster anwenden“) gesammelt habe, insbesondere [Abschnitt 7.1](#) („Muster richtig einsetzen“).

1.7 Webseite

Auf der Webseite

www.swamuster.de

finden Sie einen schlanken Musterüberblick, einige Inhalte, die aus Platzgründen nicht mehr in das Buch gepasst haben, sowie Links und weiterführende Informationen zum Thema und zu einzelnen Vorgehensmustern. Ich gebe dort auch aktuelle Vortrags- und Veranstaltungshinweise.

1.8 Danksagung

Neben der Academy möchte ich mich bei allen Helfern, Unterstützern und Kunden bedanken, die dieses Buchprojekt möglich gemacht haben. Ganz besonderer Dank gebührt Stefan Zörner, der nicht nur große Teile des Buchs gereviewed hat und viele wertvolle Anregungen beisteuerte: Er ist auch verantwortlich dafür, dass ich mich

überhaupt monatelang von allen sozialen Bindungen gelöst habe, um mich dem geschriebenen Wort zu widmen. Danke für den Impuls!

Danke an Markus Wittwer für den wichtigen Input im Bereich der gemeinsamen Entscheidungsfindung und das beigesteuerte Vorgehensmuster aus [Abschnitt 5.2](#).

Danke an meine Diskussionspartner, Reviewer und Unterstützer, die mir mit Hinweisen, Kommentaren und Ideen geholfen haben, dieses Buch zu realisieren: Jan Gentsch, Claudia Schröder, Uwe Vigenschow, Tadeusz Malek, Mischa Soujon, Kai Münch, Jan Dittberner, Wolfgang Werner, Gernot Starke, Axel Müller, Robert Uhl, Stephan Roth, Roland Mast, Niko Köbler, Harm Gnoyke, Simon Brown, Matthias Bohlen, Peter Götz, René Weiß.

Weil ich mir das Beste immer für den Schluss aufhebe, kommt nun der Dank an jene Frau, die mich während der Arbeit an diesem Buch unterstützt, angefeuert und bei Laune gehalten hat: Marion. Danke für die gemeinsame Zeit, die mich zu dem gemacht hat, was ich heute bin. Ich versuche, deine positive und liebevolle Art weiterleben zu lassen. Irgendwann werden wir uns wiedersehen und dann habe ich viel zu erzählen.

¹ Erfinder des Toyota Production Systems und Urvater von Lean

² Originalzitat auf Englisch: Taiichi Ohno: „*there is something called standard work, but standards should be changed constantly. Instead, if you think of the standard as the best you can do, it's all over.*“ Ohno goes on to say that if we establish something as the „*best possible way, the motivation for kaizen [continuous incremental improvement] will be gone.*“

³ Als ich mit diesem Buchprojekt begonnen habe, war mir kein einziges methodisch orientiertes Pattern-Buch bekannt, zwei ernstzunehmende Vertreter dieses Genres konnte ich jedoch mittlerweile in Erfahrung bringen: [Lef10][Els08]).

⁴ Deutsch etwa: Für mich drückt Softwarearchitektur die Idee von Kernelementen des Systems aus, jene Teile, die schwer änderbar sind. Ein Fundament, auf dem der Rest aufbauen muss.

⁵ Deutsch etwa: Softwarearchitektur ist die Menge der Entwurfsentscheidungen, welche, wenn falsch getroffen, Ihr Projekt scheitern lassen können.

⁶ Deutsch etwa: Die Aufmerksamkeit, die Sie Softwarearchitektur entgegenbringen, sollte vom Risiko bestimmt werden, das von Architekturfragen ausgeht. Wenn wenig Architekturrisiko für das Gesamtprojekt besteht, hilft Architekturoptimierung auch wenig.

⁷ REST steht für „Representational State Transfer“ und ist ein Stil für Webanwendungen, in dem sogenannte Repräsentationen von Ressourcen über HTTP ausgetauscht werden und über URLs eindeutig identifizierbar sind.

⁸ Sie lesen dieses Buch *nicht*, um Praktiken auf Entwicklungs- und Architekturebene vorzugeben ...

⁹ E-Mail: Stefan.Toth@embarc.de, Buchwebseite: www.swamuster.de

¹⁰ Christopher Alexander wird von vielen als der Urvater der Beschreibung von Mustern gesehen. Sein einflussreichstes Werk ist „A Pattern Language: Towns, Buildings, Construction“ [Ale78].

2 Zeitgemäße Softwarearchitektur

Auf den nächsten Seiten tauchen Sie in die inhaltliche Vision des Buchs ein. Die übergreifende Idee hinter den 30 Vorgehensmustern für Softwarearchitektur ist in [Abschnitt 2.1](#) detailliert dargestellt und mit den Konzepten der übrigen Kapitel verbunden.

Nach diesem vielleicht wichtigsten Abschnitt des gesamten Buchs zeige ich, welche Vorgehensmuster die beschriebene zeitgemäße Architekturarbeit ermöglichen. [Abschnitt 2.2](#) gibt einen kompakten Überblick aller Muster des Buchs, inklusive Problem und Kurzbeschreibung. Außerdem werden die Kapitel, in welche die Muster eingegliedert sind, kurz vorgestellt.

Zum Abschluss stelle ich kurz das Fallbeispiel vor, das Sie durch alle Vorgehensmuster begleiten wird ([Abschnitt 2.3](#)).

