

Effektive Softwarearchitekturen

Ein praktischer Leitfaden

» Hier geht's
direkt
zum Buch

DAS VORWORT

1

Einleitung

*Wir bauen Software wie Kathedralen:
Zuerst bauen wir – dann beten wir.*

Gerhard Chroust

Bitte erlauben Sie mir, Sie mit einer etwas bössartigen kleinen Geschichte zur weiteren Lektüre dieses Buchs zu motivieren.

Eine erfolgreiche Unternehmerin möchte sich ein Domizil errichten lassen. Enge Freunde raten ihr, ein Architekturbüro mit dem Entwurf zu betrauen und die Erstellung begleiten zu lassen. Nur so ließen sich die legendären Probleme beim Hausbau (ungeeignete Entwürfe, mangelnde Koordination, schlechte Ausführung, Pfusch bei Details, Kostenexplosion und Terminüberschreitung) vermeiden.

Um die für ihr Vorhaben geeigneten Architekten zu finden, beschließt sie, einigen namhaften Büros kleinere Testaufträge für Einfamilienhäuser zu erteilen. Natürlich verrät sie keinem der Kandidaten, dass diese Aufträge eigentlich Tests für das endgültige Unterfangen sind.

Nach einer entsprechenden Ausschreibung in einigen überregionalen Tageszeitungen trifft unsere Bauherrin folgende Vorauswahl:

- Wasserfall-Architektur KG, Spezialisten für Gebäude und Unterfangen aller Art,
- V&V Architektur GmbH & Co. KG, Spezialisten für Regierungs-, Prunk- und Profanbauten,
- Extremarchitekten AG.

Alle Büros erhalten identische Vorgaben: Ihre Aufgabe besteht in Entwurf und Erstellung eines Einfamilienhauses (EFH). Weil unsere Unternehmerin jedoch sehr häufig, manchmal fast sprunghaft, ihre Wünsche und Anforderungen ändert, beschließt sie, die Flexibilität der Kandidaten auch in dieser Hinsicht zu testen.

Wasserfall-Architektur KG

Die Firma residiert im 35. Stock eines noblen Bürogebäudes. Dicke Teppiche und holzvertäfelte Wände zeugen vom veritablen Wohlstand der Firmeneigner.



Bild 1.1 Foto von Wolfgang Korn

„Wir entwerfen komplexe technische Systeme“, erklärt ein graumeliertes Mittfünfziger der Bauherrin bei ihrem ersten Treffen. Sein Titel „Bürovorsteher“ prädestiniert ihn wohl für den Erstkontakt zu dem vermeintlich kleinen Fisch. Von ihm und einer deutlich jüngeren Assistentin wurde sie ausgiebig nach ihren Wünschen hinsichtlich des geplanten Hauses befragt.

Als sie die Frage nach den Türgriffen des Badezimmer-schranks im Obergeschoss nicht spontan beantworten kann, händigt man ihr Formblätter aus, die ausführlich den Change-Management Prozess beschreiben.

Das Team der Wasserfall-Architektur KG legte nach wenigen Wochen einen detaillierten Projektplan vor. Gantt-Charts, Work-Breakdown-Struktur, Meilensteine, alles dabei. Die nächsten Monate verbrachte das Team mit der ausführlichen Dokumentation der Anforderungen sowie dem initialen Entwurf.

Pünktlich zum Ende dieser Phase erhielt die Unternehmerin einen Ordner mit gut 400 Seiten Beschreibung eines Hauses. Nicht ganz das von ihr Gewünschte, weil das Entwicklungsteam aus Zeitgründen einige (der Bauherrin nur wenig zu-

sagende) Annahmen über die Größe mancher Räume und die Farbe einiger Tapeten getroffen hatte. Man habe zuerst überall groben Sand als Bodenbelag geplant, könne das aber später erweitern. Mit etwas Zement und Wasser vermischt, stünden später alle Möglichkeiten offen. Im Rahmen der hierbei erwarteten Änderungen habe das Team vorsorglich die Treppen als Rampe ohne Stufen geplant, um Schubkarren den Weg in die oberen Etagen zu erleichtern. Das Begehren unserer Unternehmerin, doch eine normale Treppe einzubauen, wurde dem Change-Management übergeben.

Die nun folgende Erstellungsphase (die Firma verwendete hierfür den Begriff „Implementierungsphase“) beendete das Team in 13 statt der geplanten acht Monate. Die fünf Monate Zeitverzug seien durch widrige Umstände hervorgerufen, wie ein Firmensprecher auf Nachfrage erklärte. In Wirklichkeit hatte ein Junior-Planning-Consultant es versäumt, einen Zufahrtsweg für Baufahrzeuge zu planen – das bereits fertiggestellte Gartenhaus musste wieder abgerissen werden, um eine passende Baustraße anlegen zu können.

Ansonsten hatte das Implementierungsteam einige kleine Schwächen des Entwurfs optimiert. So hatte das Haus statt Treppe nun einen Lastenaufzug, weil sich die ursprünglich geplante Rampe für Schubkarren als zu steil erwies. Das Change-Management verkündete stolz, man habe bereits erste Schritte zur Anpassung des Sandbodens unternommen: Im ganzen Haus seien auf den Sand Teppiche gelegt worden. Leider hatte ein Mitglied des Wartungsteams über den Teppich dann, in sklavischer Befolgung der Planungsvorgaben, Zement und Wasser aufgebracht und mithilfe ausgeklügelte brachialer Methoden zu einer rotgrauen zähen Paste vermischt. Man werde das in der Wartungsphase optimieren, hieß es seitens der Firma.

Die zu diesem Zeitpunkt von den Wasserfall-Architekten ausgestellte Vorabrechnung belief sich auf das Doppelte der ursprünglich angebotenen Bausumme. Diese Kostensteigerung habe die Bauherrin durch ihre verspätet artikulierten Zusatzwünsche selbst zu verantworten.

V&V Architektur GmbH & Co. KG

Die V&V Architektur GmbH & Co. KG (nachfolgend kurz V&V) hatte sich in den vergangenen Jahren auf Regierungs-, Prunk- und Profanbauten spezialisiert. Mit dem unternehmenseigenen Verfahren, so wird versichert, könne man garantiert jedes Projekt abwickeln. Der von V&V ernannte Projektleiter überraschte unsere Unternehmerin in den ersten Projektwochen mit langen Fragebögen – ohne jeglichen Bezug zum geplanten Haus. Man müsse unbedingt zuerst das Tailoring des Vorgehensmodells durchführen, das Modell exakt dem geplanten Projekt anpassen. Am Ende dieser Phase erhielt sie, in zweifacher Ausfertigung, mehrere Hundert Seiten Dokumentation des geplanten Vorgehens.

Dass ihr Einfamilienhaus darin nicht erwähnt wird, sei völlig normal, unterrichtete sie die Projektleitung. Erst jetzt, in der zweiten Phase, würde das konkrete Objekt geplant, spezifiziert, realisiert, qualitätsgesichert und konfigurationsverwaltet.



Der Auftraggeberin wurde zu diesem Zeitpunkt auch

Bild 1.2 Foto von Ralf Harder

das „Direktorat EDV“ der Firma V&V vorgestellt. Entgegen ihrer Annahme befasst sich diese Abteilung nicht mit Datenverarbeitung, sondern mit der „*Einhaltung Des Vorgehensmodells*“.

Nach einigen Monaten Projektlaufzeit stellte unsere Bauherrin im bereits teilweise fertiggestellten Haus störende signalrote Inschriften auf sämtlichen verbauten Teilen fest. Das sei urkundenechte Spezialtinte, die sich garantiert nicht durch Farbe oder Tapete verdecken ließe, erklärte V&V stolz. Für die Qualitätssicherung und das Konfigurationsmanagement seien diese Kennzeichen unbedingt notwendig. Ästhetische Einwände, solche auffälligen Markierungen nicht in Augenhöhe auf Fenstern, Türen und Wänden anzubringen, verwarf die Projektleitung mit Hinweis auf Seite 354, Paragraph 9 Absatz 2 des Vorgehensmodells, in dem Größe, Format, Schrifttyp und Layout dieser Kennzeichen verbindlich definiert seien. Die Bauherrin hätte bereits beim Tailoring widersprechen müssen, nun sei es wirklich zu spät.

Extrem-Architekten AG

Die Extrem-Architekten laden unsere Unternehmerin zu Projektbeginn zu einem Planungsspiel ein. Jeden Raum ihres geplanten EFH soll sie dabei der Wichtigkeit nach mit Gummibärchen bewerten. Die immer nur paarweise auftretenden Architekten versprechen ihr eine erste funktionsfähige Version des Hauses nach nur sechs Wochen. Auf Planungsunterlagen würde man im Zuge der schnellen Entwicklung verzichten.



Bild 1.3 „Gummibär-Tango“
von Klaus Terjung

Zu Beginn der Arbeiten wurde das Team in einer Art Ritual auf die gemeinsame Vision des Hauses eingeschworen. Wie ein Mantra murmelten alle Teammitglieder ständig mit seltsam gutturaler Betonung die Silben „Einfamilien-Haus“, was sich nach einiger Zeit zu „Ei-Mi-Ha“ abschloß. Mehrere Außenstehende wollen gehört haben, das Team baue einen bewohnbaren Eimer. Sie stellten eine überdimensionale Tafel am Rande des Baugeländes auf. Jeder durfte darauf Verbesserungsvorschläge oder Änderungen eintragen. Dies gehöre zu einem Grundprinzip der Firma: „Kollektives geistiges Eigentum: Planung und Entwurf gehören allen.“

Nach exakt sechs Wochen laden die Extrem-Architekten die Unternehmerin zur Besichtigung der ersten funktionsfähigen Version ein. Wieder treten ihr zwei Architekten entgegen, jedoch erkennt sie nur einen davon aus dem Planungsspiel wieder.

Der andere arbeitet jetzt bei den Gärtnern. Der ursprüngliche

andere Gärtner hilft dem Elektriker, ein Heizungsbauer entwickelt dafür die Statik mit. Auf diese Weise verbreite sich das Projektwissen im Team, erläutern beide Architekten eifrig.

Man präsentiert ihr einen Wohnwagen. Ihren Hinweis auf fehlende Küche, Keller und Dachgeschoss nehmen die Extrem-Architekten mit großem Interesse auf (ohne ihn jedoch schriftlich zu fixieren).

Weitere sechs Wochen später hat das Team eine riesige Grube als Keller ausgehoben und den Wohnwagen auf Holzbohlen provisorisch darüber befestigt. Das Kellerfundament haben ein Zimmermann und ein Statiker gegossen. Leider blieb der Beton zu flüssig. Geeignete Tests seien aber bereits entwickelt, dieser Fehler käme garantiert nie wieder vor.

Mehrere weitere 6-Wochen-Zyklen gehen ins Land. Bevor unsere Unternehmerin das Projekt (vorzeitig) für beendet erklärt, findet sie zwar die von ihr gewünschte Küche, leider jedoch im Keller. Ein Refactoring dieses Problems sei nicht effektiv, erklärte man ihr. Dafür habe man im Dach einen Teil der Wohnwagenküche verbaut, sodass insgesamt die Zahl der Küchen-Gummibären erreicht worden sei.

Das immer noch flüssige Kellerfundament hat eines der Teams bewegen, auf die Seitenwände des Hauses auf Dauer zu verzichten, um die Lüftung des Kellers sicherzustellen. Im Übrigen besitzt das Haus nur ein Geschoss, das aktuelle Statik-Team (bestehend aus Zimmermann und Gärtner) hat dafür die Garage in drei Kinderzimmer unterteilt.

Weil das Team nach eigenen Aussagen auf die lästige und schwergewichtige Dokumentation verzichtet hatte, waren auch keine Aufzeichnungen der ursprünglichen Planung mehr erhalten.

Im Nachhinein beriefen sich alle Projektteams auf ihren Erfolg. Niemand hatte bemerkt, dass die Bauherrin keines der „implementierten“ Häuser wirklich akzeptierte.

Chaos nur am Bau?

Keineswegs! Ähnlichkeiten mit bekannten Vorgehensweisen bei der Softwareentwicklung sind ausdrücklich gewollt, denn nicht nur beim Hausbau herrscht Chaos.¹ Auch andere In-

¹ Viele Grüße nach Berlin. Ähnlichkeiten mit gescheiterten Flughafenprojekten sind rein zufällig.

genieurdisziplinen erleben *turbulente Situationen*, obwohl der Maschinenbau über mehr als 200 Jahre Erfahrung verfügt. In der Softwarebranche geht es mindestens ebenso schlimm zu.

Der regelmäßige Chaos-Report der Standish-Group zeigt eine seit Jahren gleichbleibende Tendenz: Über 30 % aller Softwareprojekte werden (erfolglos) vorzeitig beendet, in über 50 % aller Softwareprojekte kommt es zu drastischen Kosten- oder Terminüberschreitungen.²

Softwarearchitektur allein kann diese Probleme nicht lösen. Stakeholder mit klaren Zielvorstellungen, ein motiviertes Entwicklungsteam und ein effektives, flexibles und (hoffentlich) agiles Management bilden wichtige Voraussetzungen für erfolgreiche Softwareentwicklung³.

■ 1.1 Softwarearchitekt:innen



Liebe Softwarearchitektinnen, ich meine überall auch Sie, obwohl ich im weiteren Buch meist die maskuline (weil kürzere) Form „Softwarearchitekt“ verwende. Ich meine grundsätzlich Softwarearchitektinnen und Softwarearchitekten –verwende aber nur sporadisch die längere Form Softwarearchitekt:innen. Das Gleiche gilt für Auftraggeber:innen, Anwender:innen und andere Rollen.

Danke für Ihr Verständnis!

Der Rolle „Softwarearchitektur“ kommt in IT- oder Softwareprojekten besondere Bedeutung zu:



Softwarearchitekten bilden die Schnittstelle zwischen Analyse, Entwurf, Implementierung, Management und Betrieb von Software.

Diese verantwortungsvolle Schlüsselrolle bleibt in vielen Projekten oft unbesetzt oder wird nicht angemessen ausgefüllt. Architekt:innen sollten sicherstellen, dass Anforderungen der Stakeholder einerseits umsetzbar sind und andererseits auch umgesetzt werden.

Softwarearchitekt:innen denken langfristig – auf die gesamte Lebensdauer von IT-Systemen bezogen. Sie ermöglichen kurzfristige Änderungen, sichern gleichzeitig die Langlebigkeit und Nachhaltigkeit von Software. Sie verfolgen konzeptionelle Integrität (auch genannt *Konsistenz*): Die gesamte Konstruktion von Software sollte einem einheitlichen Stil folgen. Insbesondere sollten *ähnliche Aufgabenstellungen in Systemen ähnlich gelöst* werden. Dies erleichtert das Verständnis und die langfristige Weiterentwicklung.

Langfristigkeit

² Quelle: The Standish Group Chaos Report. Erhältlich unter <https://standishgroup.com>

³ Eine gute Voraussetzung für Projekterfolg ist es, im Team die Eigenschaften **Kompetenz, Energie** und **Verantwortung** zu bündeln – danke an Dierk König für diese Formulierung.

Konzeptionelle Integrität

Das Buch gibt aktiven und angehenden Softwarearchitekt:innen praktische Ratschläge und Hilfsmittel, diese komplexen Aufgaben effektiver zu erfüllen. Es unterbreitet konkrete Vorschläge, wie Sie bei Softwarearchitektur in der Praxis vorgehen können. Auch wenn Sie in anderen Funktionen in Softwareprojekten arbeiten, kann dieses Buch Ihnen helfen. Sie werden verstehen, welche Bedeutung Architekturen besitzen und wo die Probleme bei Entwurf und Kommunikation von Architekturen liegen.

■ 1.2 Effektiv, agil und pragmatisch

Effektivität, Agilität und Pragmatismus prägen die Grundhaltung erfolgreicher Entwicklungsteams, insbesondere der Softwarearchitektur.

Agilität ist notwendig

Software wird in vielen Projekten immer noch als starres, unveränderliches Produkt betrachtet, obwohl Anwender und Auftraggeber laut nach hochgradig flexiblen Lösungen rufen. In der Praxis ähnelt die Softwareentwicklung leider oftmals eher dem Brückenbau: Eine Rheinbrücke bleibt auch in den kommenden Jahren eine Rheinbrücke. Weder verändert sich der Flusslauf noch wird aus einer Eisenbahnbrücke eine Startbahn für Passagierflugzeuge. Für Software stellt sich die Lage ganz anders dar: Hier kann aus einem abteilungsinternen Informationssystem schnell eine global genutzte Internet-E-Business-Lösung entstehen.

Langjährige Untersuchungen ergeben, dass sich 10 bis 25 % der Anforderungen an Software pro Jahr ändern (Quelle: Peter Hruschka). Management, Architektur und Entwicklung müssen sich in Softwareprojekten durch flexible und bedarfsgerechte Vorgehensweisen darauf einstellen. Das Schlüsselwort lautet „Agilität“.

Agilität und flexibles Vorgehen wird in Softwareprojekten an vielen Stellen dringend benötigt:

- Schon das Requirements-Engineering muss flexibel mit Änderungen von Anforderungen umgehen.
- Softwarearchitekturen müssen stabile Grundgerüste bereitstellen, die die Umsetzung neuer und geänderter Anforderungen ermöglichen. In der heutigen Marktsituation müssen solche Änderungen schnell und effektiv erfolgen – oder sie sind wirkungslos.
- Projekt- und Produktmanagement müssen in der Lage sein, während der Erstellung eines Systems flexibel auf neue Anforderungen, neue Technologien oder aktualisierte Produkte zu reagieren. Hier bietet agiles Vorgehen und risikobasiertes Projektmanagement viele Vorteile gegenüber den strikt am Vorgehensmodell orientierten konventionellen Methoden.
- Dokumentation muss sich an spezifischen Projektbedürfnissen orientieren statt an fix vorgegebenen Ergebnistypen. Inhalt ist wichtiger als Form.
- Agilität erfordert allerdings auch hohe Qualifikation und Professionalität der Beteiligten. Wenn Sie in einem agilen Projekt arbeiten, müssen Sie in allen Belangen mitdenken und Verantwortung übernehmen.

Insgesamt zählt in einem Projekt nur das Ergebnis. Selten kümmern sich Anwender- oder Auftraggeber:innen im Nachhinein um die Einhaltung starrer Vorgehensmodelle.

Softwarearchitektur muss in ihrer Funktion als Schnittstelle zwischen den Projektbeteiligten diesen Ansatz der Agilität aufnehmen und in der Praxis umsetzen.



Handeln Sie agil!

Von Peter Hruschka



Agil heißt, beweglich und flexibel sein. Mitdenken statt „Dienst nach Vorschrift“ und Dogma.

Kein Vorgehensmodell passt für alle Arten von Projekten. Eine agile Vorgehensweise beurteilt das jeweilige Risiko der unterschiedlichen Aufgaben bei der Softwareentwicklung und wählt dann die geeigneten Maßnahmen. Folgende Schwerpunkte werden dabei gesetzt:

- offen für Änderungen statt Festhalten an alten Plänen,
- eher ergebnisorientiert als prozessorientiert,
- „miteinander darüber reden“ statt „gegeneinander schreiben“,
- eher Vertrauen als Kontrolle,
- Bottom-up „Best Practices“ austauschen und etablieren statt Top-down-Vorgaben diktieren.

Trotzdem heißt „Agilität“ nicht, Anarchie zuzulassen:

- Agile Vorgehensmodelle haben Ergebnisse, nur unterscheiden sich diese für unterschiedliche Projekte in Anzahl, Tiefgang und Formalismus.
- Agile Entwicklung kennt Prozesse, nur lassen diese mehr Spielraum für Alternativen und Kreativität.
- Agile Methoden setzen auf Verantwortung; es werden nur notwendige Rollen besetzt.
- Agile Methoden basieren auf Feedback und Iterationen. Fordern und geben (*push und pull*) Sie Rückmeldung zu Ergebnissen – nur so können Teams und Ergebnisse besser werden.

Das Risiko entscheidet: Alle Beteiligten aus Management, Anforderungsanalyse, Architektur, Entwicklung und Test überprüfen ständig Risiken und entscheiden in ihrem jeweiligen Umfeld über notwendige Maßnahmen, damit aus Risiken keine Probleme werden.

Dr. Peter Hruschka (hruschka@b-agile.de) ist unabhängiger Trainer und Methodenberater. Er ist Prinzipal der Atlantic Systems Guild, eines internationalen Think Tank von Methodengurus, deren Bücher den State of the Art wesentlich mitgestaltet haben (<https://peterhruschka.eu/>).

Effektiv = Ziele erreichen

Weil das Begriffspaar „effektiv und effizient“ immer wieder für Missverständnisse sorgt, möchte ich die Bedeutung beider Wörter hier kurz gegenüberstellen.

Effizient =
hoher Wirkungsgrad

Eine Lexikondefinition des Begriffs „Effizienz“ lautet „Wirkungsgrad“, also das Verhältnis von Aufwand zu Ertrag. Wenn Sie Aufgaben effizient erledigen, dann arbeiten Sie also mit hohem Wirkungsgrad.

Sie investieren für den gewünschten Ertrag einen minimalen Aufwand. Spitzensportler etwa vermeiden in ihren Disziplinen überflüssige Bewegungen oder Aktionen, was in hochgradig effizientem Ausführen der jeweiligen Sportart resultiert.

Prägnant ausgedrückt, bedeutet das:

Effizient = Dinge richtig machen

Effektiv = zielorientiert

„Effektiv“ bedeutet zielorientiert. Sie arbeiten effektiv, wenn Sie Dinge erledigen, die zur Erreichung Ihrer konkreten Ziele notwendig sind. Auch für diesen Begriff wieder eine prägnante Definition:

Effektiv = die richtigen Dinge machen

Es ist viel wichtiger, die richtigen Dinge zu erledigen, als irgendwelche Dinge besonders effizient zu tun.

Softwarearchitekten müssen in hohem Maße effektiv arbeiten. Kunden und Auftraggeber bestimmen Ziele, Architekten müssen sicherstellen, dass diese Ziele auch erreicht werden.

Effektiv = agil und angemessen

Effektiv bedeutet auch, angemessen und bedarfsgerecht zu agieren. Auf die Entwicklung von Software angewandt, heißt das, sich permanent an den Bedürfnissen der Kunden und Auftraggeber zu orientieren (und nicht starr an den Buchstaben eines formalen Vorgehensmodells). Ich plädiere in diesem Buch für Agilität in diesem Sinne.

Effektive Softwarearchitektur bedeutet, das (für Kunden und Auftraggeber) richtige System zu konstruieren und zu bauen – mit technisch und organisatorisch angemessenen Mitteln.

■ 1.3 Wer sollte dieses Buch lesen?

Grundsätzlich können alle Stakeholder⁴ von diesem Buch profitieren und erhalten Antworten auf zentrale Fragen.

- Softwarearchitektur
 - Was sind die Methoden und Werkzeuge unserer Zukunft?
 - Wie gehen Sie beim Entwurf von Architekturen sinnvoll vor?
 - Welche praktisch erprobten Heuristiken und Ratschläge gibt es?
 - Wie meistern Sie externe Einflussfaktoren, die den Entwurf von Architekturen erschweren?
 - Wie können Sie Architekturen pragmatisch und effektiv kommunizieren?
 - Was sind die grundlegenden technischen und querschnittlichen Konzepte?
- Softwareentwickler:innen
 - Wie hängen Architektur und Implementierung zusammen?
 - Wie kann das gesamte Entwicklungsteam bei Entwurf und Pflege der Architektur konstruktiv mitwirken?
 - Welche allgemeinen Prinzipien von Softwarearchitektur und -entwurf sollten Sie bei der Implementierung unbedingt befolgen?
- Alle Personen, die sich auf die Prüfung zum „Certified Professional for Software Architecture – Foundation Level“ (CPSA-F) des iSAQB e. V. vorbereiten
- Technische Manager:innen
 - Warum sollen Sie für Architektur Geld ausgeben? Warum ist Softwarearchitektur wichtig?
- Product-Owner, Scrum-Master, Projektleiter:innen
 - Was genau bewirkt Architektur in IT-Projekten?
 - Welche Aufgaben erfüllt Softwarearchitektur?
 - Welche Bedeutung hat die Dokumentation von Architekturen („Was bedeuten all diese Symbole?“)?
 - Welche grundlegenden Lösungsansätze gibt es für Architekturen?
- Business-Analysts und Requirements-Engineers
 - Was bedeuten all diese Begriffe, die das Entwicklungsteam und die Architekten ständig verwenden?
 - Wie können Sie bei der Architekturentwicklung konstruktiv beitragen?

⁴ Im Verlauf des Buchs benutze ich den Begriff Stakeholder für Personen oder Organisationen, die bei der Erstellung des Systems mitwirken, es beeinflussen oder am entwickelten System ein fachliches, technisches oder kommerzielles Interesse haben.

■ 1.4 Wegweiser durch das Buch

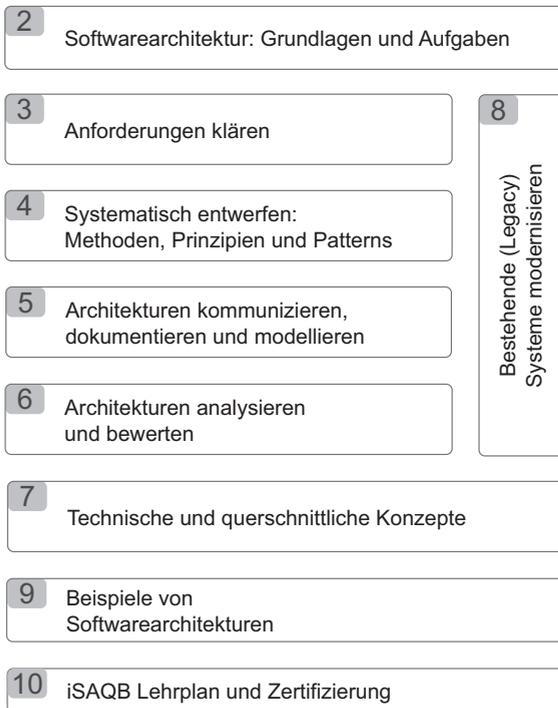


Bild 1.4
Wegweiser

Kapitel 2 klärt die Grundbegriffe rund um Architektur. Es beantwortet die Fragen nach dem Was, Warum und Wer von Softwarearchitekturen. Damit legt es das Fundament für eine systematische und gleichzeitig flexible Vorgehensweise.

Kapitel 3 widmet sich der Aufgabe, Anforderungen systematisch zu klären.

Kapitel 4 gibt Ihnen methodische Werkzeuge für zielgerichtete Entwurfsentscheidungen an die Hand. Sie lernen hier die Grundlagen des Entwurfs, geltende Prinzipien und erprobte Heuristiken. In diesem Teil bekommen Sie eine kompakte Einführung in *Domain-Driven Design*, *Quality-Driven Architecture* sowie in die wesentlichen Architektursichten. Neben diesen Methoden zur Strukturierung Ihrer Systeme und Bausteine lernen Sie hier auch einige wichtige Patterns kennen.

arc42 Template

Kapitel 5 beschreibt, wie Sie Ihre Softwarearchitekturen kommunizieren und dokumentieren können. Außerdem finden Sie hier einige Hinweise für gute Architekturdokumentation sowie das bekannte Architekturtemplate arc42. Zusätzlich lernen Sie einige Grundlagen von UML und dem C4-Ansatz kennen.

Kapitel 6 zeigt, wie Sie methodisch Architekturen analysieren und bewerten können, insbesondere hinsichtlich der geforderten Qualitätseigenschaften.

(technische) Konzepte

Kapitel 7 enthält einen Katalog häufig benötigter technischer Konzepte. Hierzu zählen Persistenz (Datenspeicherung), Integration,

Verteilung, Kommunikation, Sicherheit, grafische Benutzeroberflächen, Ausnahme- und Fehlerbehandlung sowie Management von Geschäftsregeln.

Kapitel 8 beleuchtet Änderung, Modernisierung und Evolution von Software – womit Entwicklungsteams vermutlich 80 % ihrer beruflichen Zeit verbringen.

Modernisierung und Evolution

Kapitel 9 enthält Beispiele von Softwarearchitekturen, beschrieben nach der Strukturvorlage arc42 (die Sie in Kapitel 4 kennengelernt haben).

Beispiele von Architekturen

In **Kapitel 10** finden Sie einige Hinweise zur Vorbereitung auf die iSAQB Foundation Level Zertifizierung CPSA-F

Vorbereitung auf CPSA-F

■ 1.5 Webseite zum Buch

Auf der Website <https://www.esabuch.de> finden Sie Informationen, die aus Platzgründen ins Internet weichen mussten. Dazu gehören aktuelle Literaturhinweise und Links sowie sonstige Hilfsmittel für Softwarearchitekten zum Download.

Website:
<https://www.esabuch.de>

Sie können unter <https://www.arc42.de> den aktuellen Stand des praxisnahen und umfassend erprobten Architekturtemplates herunterladen – hilfreich für Entwurf, Entwicklung und Dokumentation von Softwarearchitekturen.

<https://www.arc42.de>

■ 1.6 Weiterführende Literatur

[DeMarco+07] beschreiben (äußerst humorvoll und gnadenlos wahr) mehr als 80 hilfreiche „Verhaltensmuster“ aus IT-Projekten – die meiner Meinung nach alle ITler kennen sollten.

Ebenfalls in Form informeller Muster nehmen Peter Hruschka und ich im [Knigge] diverse typische positive und negative Verhaltensweisen rund um Softwarearchitektur ins Visier.

Falls Sie Softwarearchitektur an realen Beispielen „erleben“ möchten (und Ihnen Kapitel 9 nicht genügt) – unter <https://leanpub.com/arc42byexample> finden Sie weitere Beispiele (u. a. eine Schach-Engine und ein Portal für Radsport).

Als englische Literatur lege ich Ihnen [Keeling], [Ford+17] und auch [Bass+21] nahe. Falls Sie verteilte Systeme (etwa: Microservices, Self-Contained-Systems oder generell Client/Server) bauen, diskutieren [Ford+21] viele praxisrelevante Fragestellungen.



■ 1.7 Danksagung

Danke an meine Traumfrau, Cheffe Uli, für Engelsgeduld, Motivation und tolle gemeinsame Zeit auf Bergen, Rädern, Yogamatten und Sofas sowie in Gyms. Du sorgst für mein glückliches Leben!

Seit über 25 Jahren habe ich ungemein von den Diskussionen und Arbeiten mit Peter Hruschka rund um Methodik der Softwareentwicklung sowie arc42 profitiert. Danke, dass ich trotz aller inhaltlicher Differenzen immer noch Dein Freund sein darf!

Danke an Phillip Ghadir, Tobias Hahn, Dr. Simon Harrer, Alexander Heusingfeld, Wolfgang Korn, Stefan Tilkov und Oliver Wolf für ihre Beiträge.

Danke an das großartige Team der INNOQ – ihr seid die Besten!

Lynn und Per, für die wirklichen Prioritäten.

Danke an meine zahlreichen Reviewer – namentlich nenne ich euch auf <https://www.esabuch.de>. Ihr habt mich immer wieder auf Fehler und Ungenauigkeiten hingewiesen und das Buch über die Jahre damit immer besser gemacht.

Danke an Dr. Alexander Lorz für erhellende Diskussionen um die Grundlagen unserer Zunft.

Danke auch an die vielen ehrenamtlichen Mitglieder des iSAQB e. V., insbesondere aus der Foundation-Level Working Group, für konstruktive Diskussionen zu grundlegenden Fragen der Softwarearchitektur.

Zu guter Letzt vielen Dank an meine Kunden, dass ich in Ihren/euren Projekten so viel über Softwarearchitekturen erfahren und erleben durfte.

In Gedenken an den großartigen Stefan Tilkov – der viel zu früh gegangen ist. Du warst Vorbild, Mentor, Moderator und Möglichmacher. Du fehlst mir!