

Einführung

Wenn eine Gruppe von Menschen zusammenleben muss, kann einiges schiefgehen – sei es in einer Partnerschaft, in der Familie oder einer größeren Zweckgemeinschaft eines Wohnblocks. Für ein friedliches Zusammenleben braucht es eine Ordnung, die je nach Situation expliziter oder impliziter definiert ist. Ein frisch verliebtes Paar braucht nicht unbedingt eine Hausordnung, viele Regeln ergeben sich durch die gegenseitige Rücksichtnahme. In einem Mehrfamilienhaus hingegen wird es bereits schwieriger, besonders wenn es gemeinsame Ressourcen, wie zum Beispiel eine Waschküche, zu teilen und zu organisieren gibt. Nicht selten führen sogenannte Kleinigkeiten zu großen Problemen. Je mehr Menschen mit unterschiedlichen Ansichten und Lebenseinstellungen aufeinandertreffen, desto wichtiger wird eine explizite Ordnung.

Analog gilt dies auch für Software-Systeme. Ein PC zu Hause kann ad-hoc betrieben werden. In einem größeren Unternehmen gibt es eine Vielzahl von Systemen, auf denen die unternehmenskritischen und weniger kritischen Applikationen laufen. Sie müssen miteinander Daten austauschen, sind prozess-technisch voneinander abhängig und die Schlagader eines Unternehmens. Diese Komponenten brauchen für die korrekte Funktionsweise genauso eine Ordnung wie wir Menschen, besonders da diese von Menschen entwickelt und betrieben werden. Mit der Software-Architektur werden solche IT-Systeme beschrieben und mittels Prinzipien deren Funktionieren festgelegt. Deshalb gefällt mir die Definition des Begriffs „Software-Architektur“ von Wilhelm Hasselbring sehr gut:

Die grundlegende Organisation eines Systems, dargestellt durch dessen Komponenten, deren Beziehungen zueinander und zur Umgebung sowie den Prinzipien, die den Entwurf und die Evolution des Systems bestimmen.

Wilhelm Hasselbring

Dieses Buch wird Ihnen dabei helfen, sich in den Architekturen zurechtzufinden und zu bewegen. Es wird Ihnen helfen, gesamtheitlichere Architekturen zu entwerfen.

Architekturen

Diese Ordnung gibt es auf verschiedenen Ebenen, in verschiedenen Architekturen, wie z. B. die Architektur der Software, der Infrastruktur, der Netzwerke, aber auch der Geschäftsprozesse. Diese Architekturen bilden im Unternehmen wiederum eine Zweckgemeinschaft, die das Ziel hat, mit der IT die Erreichung der Geschäftsziele des Unternehmens zu unterstützen. Dafür gibt es die Unternehmensarchitektur oder auch Enterprise-Architektur (siehe folgendes Bild). Wir werden verschiedene Ebenen der Architektur durchlaufen, wobei der Fokus auf der Applikationsarchitektur, der Software, liegt.

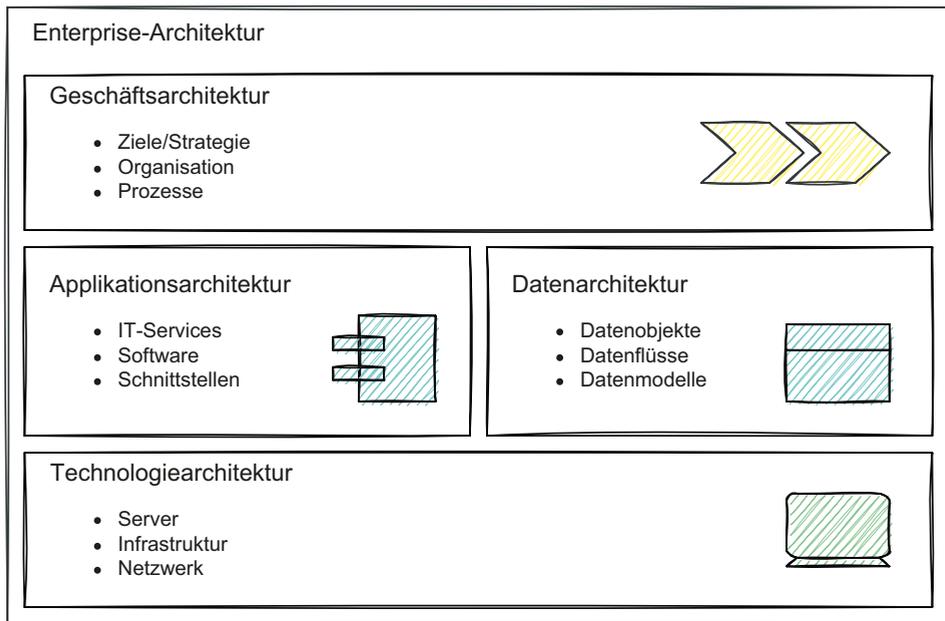


Bild 1 Einordnung der Architekturen

Struktur

Betrachten wir die Definition von Software-Architektur von Wilhelm Hasselbring genauer: Es geht darum, einen Plan, eine Struktur zu haben. Um solch einen Plan zu erhalten, müssen die Systeme in Komponenten zerlegt und beschrieben werden. Wichtig sind deshalb dessen Beziehungen zueinander. Dieser Plan schafft also eine Ordnung und einen Überblick. Vergleichen Sie es mit einem Stadtplan für Touristen (widerspiegelt die Organisation). In diesen Plänen sind die Sehenswürdigkeiten (Komponenten) prominent dargestellt. Je nach Interessengebiet des Besuchers gibt es dann Empfehlungen, in welcher Reihenfolge welche Sehenswürdigkeiten besucht werden könnten. Diese werden also zueinander in Beziehung gestellt und mittels Wegbeschreibungen verknüpft. Für den Tourist überflüssige Informationen, z. B. die Börse, werden aus Platzgründen weggelassen. Das heißt nicht, dass diese unwichtig sind, aber für den Zweck eines Reiseführers sind sie etwas weniger relevant. *Architektur ist also eine Sicht auf Systeme, die sich auf die relevanten Aspekte konzentriert.* Die nicht benötigten Informationen für einen Adressaten werden weggelassen. Der Touristenplan hat das Ziel, dem Tourist die schönsten Orte der Stadt zu zeigen. Ein Plan für öffentliche Verkehrsmittel verfolgt ein anderes Prinzip. Dort wird das Prinzip der übersichtlichen schematischen Darstellung der Transportwege angewendet.

Gernot Starke hat hierzu folgende Punkte in seinem Buch „Effektive Software-Architekturen“ [Sta18] aufgelistet:

- *Architektur enthält Strukturen.*
- *Architektur beschreibt eine Lösung und basiert auf Entscheidungen.*
- *Architektur schafft Ordnung und einen Überblick.*

- *Softwarearchitekturen machen Komplexität von Systemen beherrschbar und verständlich.*
- *Architektur lässt nicht benötigte Informationen gezielt weg.*

Je nach beruflicher Erfahrung dürften die Erläuterungen unterschiedliche Assoziationen hervorrufen. Ein Software-Entwickler wird einen Plan von Klassen und Komponenten, entworfen in UML, vor sich sehen. Ein Applikationsarchitekt sieht einen Plan von ganzen Software-Systemen vor sich, entworfen in ArchiMate. Und Sie?

■ Was erwartet Sie in diesem Buch?

Als Architekt einer Teilarchitektur, einer bestimmten Domäne, dürfen Sie sich als Vertreter an den Unternehmensdiskussionen zur Unternehmensarchitektur beteiligen. Es macht Freude, sich aktiv an Veränderungen beteiligen zu können. Vielleicht sind Sie auf dem Weg dazu oder wurden es kürzlich. Leider ist es nicht ganz einfach, alle Architekturzusammenhänge zu sehen und zu verstehen. Dafür werden tüchtige Architekten mit Unternehmenssicht gebraucht. Mit Weitsicht und der Fähigkeit, über den Tellerrand zu sehen! Es werden pragmatische Architekten gebraucht.

Das Wort „pragmatisch“ kommt aus dem Griechischen und heißt **geschäftskundig, tüchtig**. Es ist eine *Einstellung*, bei der der Pragmatiker auf die sachlichen Gegebenheiten und auf *praktisches Handeln* ausgerichtet ist. In diesem Buch geht es also um das *Tun*. Wie können Sie die sachlichen Gegebenheiten Ihrer Tätigkeiten praxistauglich anwenden? Aber angepasst, mit pragmatisch meine ich nicht waghalsig, also schnelle Entscheidungen ohne Risikoabwägung zu tätigen!

Egal, ob Sie bereits Architekt sind oder einer werden möchten und egal, aus welchem Bereich der IT Sie kommen: Ein paar *Muster* und *Prinzipien* der Software-Architektur helfen Ihnen, unternehmensweite Entscheidungen zu treffen. Für diese Muster müssen Sie nicht programmieren können. Wir lernen vor allem diese Muster zu lesen. Sie zu *analysieren*.

Es geht dann weiter in die *Applikationsarchitektur* und werden das Unternehmen *Scotland Trading* in Ihrer Transformation aktiv begleiten. Dabei stehen die Zusammenhänge im Mittelpunkt. Wir *beschäftigen* uns mit der Architektur und denken uns durch ein Projekt, bevor die Teilarchitekturen die Arbeit weiterführen. Wie in der Architektur üblich, gibt es für eine Problemstellung oder Situation kein „richtig“ oder „falsch“, sondern eher ein mehr oder weniger zutreffend. Deshalb ist mir das Wort *beschäftigen* wichtig. Lernen Sie, welche Best Practises in welcher Situation angebracht sind. Um dies zu lernen, hilft es, mit offenen Augen durch den IT-Dschungel zu gehen. Es gibt immer etwas zu entdecken.

Das Buch ist folglich in zwei Teile gegliedert: Architektur entdecken und aktiv eine Architektur entwerfen.

Teil 1: Architektur entdecken

Im ersten Teil entdecken wir die Software-Architektur am Beispiel der Betriebssysteme. Diese sind die Basis der Informatik. Sie regeln zum Beispiel den Zugriff auf gemeinsame Güter und ermöglichen ein Zusammenleben mehrerer Prozesse. Interessant ist für die Architekturanalyse die Evolution – von Unix aus den 60er-Jahren zu mobilen Betriebssystemen, die uns in den letzten Jahren eroberten. Wie kann diese Veränderung mit der Architektur vollzogen werden? Wir wollen Software analysieren und Architekturmerkmale identifizieren. So ergibt sich ein Katalog aus *Architekturprinzipien* und *Architekturmustern*, die wir in anderen Architekturen wieder antreffen.

Teil 2: Entwerfen einer Architektur

Aus den spezifischen Details wachsen wir, wie soft im Leben, heraus und sind plötzlich für größere Zusammenhänge mitverantwortlich, zum Beispiel, wenn der IT-Infrastruktur-Spezialist zum Architekten wird. Er soll dann in wichtigen Unternehmensprojekten mit all den anderen Architekten eine optimale gemeinsame Architektur mitdefinieren.

Im zweiten Teil wachsen wir also als Architekt von der Software-Architektur in die Unternehmensarchitektur hinein. Wir wechseln unsere Rolle vom Analysten zum Architekten. Die Einflussfaktoren und Qualitätsmerkmale prägen dessen Arbeit stark. Um das Thema konkreter anzugehen, durchlaufen wir die Architektur anhand eines Transformationsprojekts: Das Unternehmen *Scotland Trading* will sich neu positionieren und einen dynamischen Preisbilder bauen. Dazu durchlaufen wir gewisse TOGAF-Phasen, entdecken ArchiMate und vertiefen uns in der *Applikations- und Informationsarchitektur*. Dabei lernen wir verschiedene Sichten kennen, die uns helfen, die Architektur zu beschreiben. Mit den konkreten Architekturkonzepten aus dem ersten Teil sind wir gut gewappnet für diese Aufgabe.

Ein wichtiger Teil der Architektur ist die *Integrationsarchitektur*. Dort fügen wir die verschiedenen Bausteine zu einem Kommunikationsnetz zusammen. Wir überlegen uns, wie wir Tausende von Punkt-zu-Punkt-Verbindungen eleganter umsetzen können.

Ich achte hier wieder auf den pragmatischen Weg, der den Übergang der Sichtweise fördert.

Buchaufbau

Zusammengefasst ist das Buch folgendermaßen aufgebaut:

Teil 1: Architektur entdecken	Kapitel 1 bis 3	<i>Thema:</i> Architektur kennenlernen <i>Sicht:</i> Architekt beim Analysieren <i>Beispiel:</i> Betriebssysteme <i>Resultate:</i> Architekturprinzipien und Muster
Teil 2: Entwerfen einer Architektur	Kapitel 4 bis 8	<i>Thema:</i> Mitarbeit in der Unternehmens- und Applikationsarchitektur <i>Sicht:</i> Architekt beim Modellieren <i>Beispiel:</i> Transformationsprojekt der <i>Scotland Trading</i> <i>Resultate:</i> Architekturbauusteine und Tools, in denen die Muster und Prinzipien von Teil 1 angewendet werden können

In jedem Kapitel werden wir Hinweise zu anderen Situationen erhalten und erfahren, wie wir Pragmatiker mit den Informationen umgehen können. Am Ende jedes Kapitels merken wir uns, in Form einer Checkliste, an was wir denken sollten. Oft sind das Fragen, die wir von Zeit zu Zeit wieder hervorholen können und über die wir unsere aktuelle Arbeit hinterfragen können ...

Dieses Buch wendet sich ausdrücklich an alle Geschlechter und Menschen jeglicher Geschlechtsidentität. Für die bessere Lesbarkeit wird die männliche Bezeichnung verwendet.

■ Zusatzmaterial

Unter

<https://plus.hanser-fachbuch.de/>

finden Sie den Quellcode der Programme aus dem ersten Teil und die ArchiMate-Modelle aus dem zweiten Teil. Mit der Eingabe des Codes

plus-12abc-8xyz9

können Sie diese herunterladen.

Unter

<https://arch.xapps.ch>

finden Sie Unterlagen, die ich in meinem Unterricht zu diesem Thema verwende.

■ Danksagung

Ich durfte viele Diskussionen mit Kollegen, Studenten und ehemaligen Kunden zu diesem Thema führen. Vieles ist in dieses Buch auf die eine oder andere Art eingeflossen. Dafür danke ich und hoffe auf weitere spannende Diskussionen.

Einen besonderen Dank möchte ich meinem Architekturkollegen Ferdinand Moosmann geben. Er hat mir viele Tipps und Anregungen während des Entstehens des Buchs gegeben.

Wenn ich nicht für die Schule, an der ich Lehrbeauftragter bin, dieses Modul aufgebaut hätte, wäre dieses Buch wohl nicht entstanden. Dafür möchte ich mich bei Beat Hartmann bedanken.

Diese Leseprobe haben Sie beim
 edv-buchversand.de heruntergeladen.
Das Buch können Sie online in unserem
Shop bestellen.

[Hier zum Shop](#)