

1

Einleitung

■ 1.1 Buchmodell

If you want to build a ship, don't drum up people together to collect wood and don't assign them tasks and work, but rather teach them to long for the endless immensity of the sea. – Antoine de Saint Exupéry

Die Sprache als Werkzeug

Softwaresysteme sind komplex und schwierig zu begreifen. Die Fehler unsichtbarer Laufzeitumgebungen geben uns bei der Analyse Rätsel auf. Wenn wir verstehen möchten, was ein System tut, müssen wir sein Umfeld und die Arbeit verstehen, die mit dem System verrichtet wird. Dieses Verständnis ist schwer aufzubauen, wenn uns die gemeinsame Sprache der Konzepte des Systems fehlt. Haben wir jedoch ein gemeinsames, fachliches Verständnis der Domäne und sind Technik und Geschäft kohärent, so können wir unsere natürlichen linguistischen Fähigkeiten für die Analyse und Synthese eines Systems nutzen.

Aus diesem Grund habe ich für dieses Buch ein eigenes Modell entwickelt, eine Art von Allgemeinsprache (5.3), in der jeder verwendete Fachbegriff genau beschrieben ist. Was im Entwurf eines Systems funktioniert, das sollte für ein Buch lange reichen.

Jeweils zu Beginn eines Abschnitts habe ich die im Text folgenden Beschreibungen und Definitionen der besseren Übersichtlichkeit halber als *TL;DR* aufgeführt. Der ganz eilige Leser kann nur diese Textboxen lesen, um die Inhalte dieses Buches zu erfassen.

Für wen ist dieses Buch geschrieben?

Dieses Buch enthält das Wissen um die Architektur von Web- bzw. Geschäftssystemen, also Systemen, die über das Internet funktionieren und für die Arbeit mit Menschen entworfen werden. Das Buch richtet sich an Softwarearchitekten oder jene, die Softwarearchitekt werden möchten. Ebenfalls interessant ist das Buch für Personen, die im Anforderungsmanagement arbeiten. Durch die durchgängige Illustration mit Qualitätsszenarien soll das Buch helfen, Anforderungen besser zu spezifizieren.

Nicht geeignet ist das Buch zum Erlernen spezifischer Technologien oder Programmiersprachen. Zwar sind immer mal wieder konkrete Beispiele für Technologien angegeben, aber dieses Buch konzentriert sich auf Entwurfsmuster und Methoden.

Aufbau des Buches

Im ersten Teil dieses Buches werden verschiedene Architekturstile und Entwurfsmuster für Geschäftssysteme beschrieben: Was bedeutet Architektur und wie wird sie angewendet? Die vorgestellten Möglichkeiten werden in den Teilen danach über die Qualitätsmerkmale beschrieben, die für ein Geschäftssystem heute relevant sind. Jedes Qualitätsmerkmal ist dabei in Submerkmale untergliedert, welche wiederum durch verschiedene Techniken und Methoden im Detail erläutert werden.

Somit eignet sich der Aufbau des Buches auch als Checkliste für Systembewertungen oder als Grundlage für die Planung des Entwurfs. Ich selber nutze in meiner Praxis genau diesen Aufbau, um effizient Expertisen zu schreiben und Beratungsleistungen zu erbringen.

Geschäftssysteme

In diesem Buch geht es um den Entwurf, die Bewertung, Implementierung und Wartung von Geschäftssystemen (1.3), also Systemen, die die Arbeit mit Menschen in Organisationen über das Internet unterstützen. Geschäftssysteme reagieren auf Geschäftsereignisse mit Geschäftsfällen, deren Geschäftstätigkeiten durch Geschäftsregeln bestimmt werden. Das Geschäft steht in diesem Buch für die Arbeit einer Organisation und kann jeden erdenklichen fachlichen Hintergrund haben. Allgegenwärtig ist dabei die *unverzichtbare Komplexität*, die ein Geschäftssystem abbilden muss und die nicht reduziert werden kann. Eine Hauptaufgabe des Architekten ist neben der Möglichkeit, diese Komplexität zu bewältigen, vor allem die Kontrolle und Eliminierung unnötiger Komplexität. Hierfür bietet dieses Buch eine Dokumentation verschiedener Architekturstile, Entwurfsstandards, Qualitätsmodelle sowie Qualitätsmerkmale, die für Geschäftssysteme relevant sind.

Architekturstile

In diesem Buch werden drei Architekturstile vorgestellt, die aufeinander aufbauen und komplementär zueinander stehen. Zum besseren Verständnis gehe ich auf die Geschichte und Hintergründe der Stile ein. Zudem gibt es hier und dort Vergleiche mit dem Architekturstil des Monolithen.



Sammlungen von Entwurfsprinzipien heißen *Architekturstile*. In diesem Buch sind drei verschiedene Stile beschrieben, die zueinander passen und gemischt werden dürfen.

- Serviceorientierte Architektur (1.3)
- Microservice-Architektur (1.4)
- Domänengetriebener Entwurf (1.5)

Entwurfsstandards

Für Microservices, sonstige Services, Monolithen, Service-Bus-Systeme und Öffentliche APIs haben sich in den letzten Jahren Entwurfsmuster gebildet, die ich in diesem Buch festhalten möchte.



Der Entwurfsstandard ist eine Vorschrift der Governance für den Systementwurf. In diesem Buch sind drei Standards zur Vorgabe enthalten.

- Entwurfsstandard des Service (2.3)
- Entwurfsstandard des Open Host (2.4)
- Entwurfsstandard des Service Bus (2.5)

Qualitätsmodelle für Geschäftssysteme

Ich habe es oft erlebt, dass die für den Entwurf eines Systems notwendigen Grundlagen in Form von Qualitätsanforderungen nicht vorhanden waren. Zwar gibt es immer den einen oder anderen Satz, dass das System superschnell und sehr sicher sein soll, aber diese Beschreibungen sind meistens unzulänglich. Ich denke, dass der Grund hierfür in mangelndem Wissen um die möglichen Qualitäten und ihre Beschreibung ist. Es ist mir also ein Anliegen, die für ein Geschäftssystem notwendigen Qualitäten zu beschreiben, damit diese als Bewertungsgrundlage dienen kann: Welche Eigenschaften muss ein Service haben, damit er im Kontext erfolgreich sein kann?

Ein Qualitätsmodell beschreibt, bewertet und sagt Qualität voraus. Das Modell bildet damit die Basis des Entwurfs, der bewertet werden soll. In diesem Buch sind drei Qualitätsmodelle für Geschäftssysteme enthalten.

- Cloud-Native Systeme (3.1)
- Reaktive Systeme (3.1)
- Geschäftssysteme (3.1)

Es gibt heute viele verschiedene Methoden und Techniken in der Softwareentwicklung wie Eskalationsmanagement, Open Auth Connect, Mehrfaktorauthentifizierung oder Sandboxing. Aber es ist nicht in jedem Fall klar, welche Qualitäten des Produkts diese Methoden beeinflussen, und dies erschwert die Investitionsplanung. Wird ein wartbarer Service benötigt, aber nicht in Methoden investiert, die Wartbarkeit schaffen, so ist das ein Problem. Ein System kann funktional fehlerfrei sein und doch maßlos enttäuschen, weil stillschweigende Erwartungen nicht erfüllt wurden. Dem Architekten kommt die schwierige Rolle zu, diese Erwartungen im Rahmen des vorhandenen Budgets über die Qualitätsmerkmale zu verhandeln.

Qualitätsmerkmale von Geschäftssystemen



Qualitätsmerkmale sind Eigenschaften eines Systems. In diesem Buch sind 19 Qualitätsmerkmale beschrieben. Zu jedem Qualitätsmerkmal gibt es eine Herleitung, Qualitätsszenarien, Entwurfsmuster und Methoden für den praktischen Einsatz.

- Wartbarkeit (II)
 - Konzeptionelle Integrität (5)
 - Konsistenz (6)
 - Testbarkeit (7)
 - Analysierbarkeit (8)
 - Änderbarkeit (9)
- Performance (III)
 - Latenz (11)
 - Service-Performance (12)
 - Kapazität (13)
 - Skalierbarkeit (14)
- Zuverlässigkeit (IV)
 - Verfügbarkeit (16)
 - Herstellbarkeit (17)
 - Prüfbarkeit (18)
 - Resilienz (19)
- Informationssicherheit (V)
 - Identifizierung (21)
 - Autorisierung (23)

Danksagung

Ohne die Hilfe meiner hochgeschätzten Kollegen wäre dieses Buch niemals entstanden. Besonderen Dank möchte ich jenen aussprechen, die mit mir gemeinsam die Diskussion gesucht haben. Viele der Inhalte und Aussagen dieses Buches sind aus dem gemeinsamen Schreiben von Artikeln für Fachmagazine entstanden. Mein ausdrücklicher und großer Dank gilt Christoph Huber, Olaf Otto, Nicolas Bär, Christian Wittwer, Thomas Jaggi, Gion Manetsch, Verena Linder, Urs Siegenthaler, Carlo Bonati, Christoph Camenisch, Marcel Wiedemeier, Karsten Petersen, Stefan Zörner und Thomas Walser. Besonderen Dank gebührt Daniel Rey, ohne dessen exakte und fundierte Kritik mir die abschließende Korrektur des Werks sehr viel schwerer gefallen wäre.

Ich möchte mich außerdem bei all denjenigen Architekten und Autoren bedanken, die das Wissen um die Serviceentwicklung im Web vorantreiben und aktiv teilen. Dazu gehören insbesondere auch Organisationen, die ihr Wissen durch das Führen von Engineering Blogs und die Veröffentlichung von Open Source teilen.