

Vorwort

»Alle glücklichen Familien gleichen einander, jede unglückliche Familie ist auf ihre eigene Weise unglücklich.« Dieses Anna-Karenina-Prinzip gilt nicht nur für Familien, sondern auch für Entwicklungsteams von Legacy-Software. Verschiedene Softwaresysteme leiden unter unterschiedlichen »Krankheiten« und wir brauchen eine Vielzahl von Mitteln, um sie alle zu heilen. Die folgenden Krankheiten haben wir in unterschiedlichen Kombinationen in den letzten Jahren bei Legacy-Systemen – egal ob Monolithen oder Microservices – und ihren Organisationen beobachten können:

- Das Legacy-System ist ein **Big Ball of Mud** geworden, weil niemand die Abhängigkeiten kontrolliert hat und nun alles mit allem verbunden ist (s. Kap. 4).
- Die Fachlichkeit ist in einem **großen Domänenmodell** miteinander verwoben, dessen Teile nur begrenzt zusammenpassen oder sich sogar widersprechen (s. Abschnitt 11.4).
- Der **fachliche und technische Sourcecode** ist vermengt, und so wird der Austausch von veralteter Technik oder eine fachliche Erweiterung zu einer Herkulesaufgabe (s. Kap. 7).
- Die Menschen sind in einer **Teamstruktur** organisiert, die für schnelles Vorankommen ungeeignet ist (s. Kap. 8).

In diesem Buch stellen wir die Heilmittel gegen diese Krankheiten vor, die wir in den letzten 20 Jahren bei unserer Arbeit mit **Domain-Driven Design (DDD)** und **Legacy-Software** kennenlernen durften. Dazu gehören: **Refactorings**, **Domain Storytelling**, **Event Storming**, **Team Topologies** und der **Modularity Maturity Index (MMI)**. Wenn wir in unseren Projekten und mit unseren Kunden die Heilung angehen, dann erreichen wir, dass das jeweilige Entwicklungsteam deutlich positiver in die Zukunft schaut und die Motivation steigt. Je weiter die Heilung voranschreitet, desto glücklicher werden auch unsere Fachanwender, Projektleiter und Manager sein, denn die schwergängige und teure Legacy-Software wird stabiler, lässt sich schneller anpassen und erlaubt schließlich sogar (wieder) innovative, zukunftsweisende Erweiterungen.

Unser Ziel mit diesem Buch ist es, Ihnen Mittel an die Hand zu geben, die Ihnen helfen, Ihre Softwaresysteme so aufzuteilen, dass Änderungen möglichst lokale Auswirkungen haben. Dadurch werden die einzelnen Softwaremodule für die Entwicklungsteams wieder beherrschbar und verstehbar und können von entsprechend klei-

nen schlagkräftigen Teams bearbeitet werden. Deshalb finden Sie in diesem Buch eine ganze Reihe Sourcecode-Beispiele. Die Legacy-Systeme, die uns begegnen, sind in ganz unterschiedlichen Programmiersprachen¹ gebaut: Java, C#, PHP, C++ und viele andere. Wir haben uns entschieden, die Beispiele in der Sprache zu schreiben, die im Vergleich am häufigsten auftaucht, nämlich Java. Die Beispiele sind kurz genug, dass sie auch Programmierern anderer Sprachen verständlich sein sollten, und das Vorgehen, das wir beschreiben, ist unabhängig von der Programmiersprache.

Einige Leserinnen und Leser haben möglicherweise unsere anderen Bücher *Langlebige Softwarearchitekturen* [Lilienthal 2019] und *Domain Storytelling* [Hofer & Schwentner 2023] gelesen. Das vor Ihnen liegende Buch ist für uns die Vereinigung und logische Folge aus den beiden anderen Büchern und unserer täglichen Arbeit in unseren Projekten. Die Leser dieser beiden Bücher werden die Beispieldomänen Kino und Autoleasing und unsere Sicht auf Architektur wiedererkennen. Darüber hinaus gibt es viel Neues zu entdecken :-). Wir hoffen sehr, dass Ihnen diese Synthese gefällt, und freuen uns auf Ihr Feedback!

Ganz besonders möchten wir uns an dieser Stelle bei allen bedanken, die uns in den letzten Jahren unterstützt, uns Ideen für dieses Buch gegeben und mit uns diskutiert haben. Die sehr aktive und engagierte **DDD-Community** und unsere Firma, die **WPS** – Workplace Solutions, seien hier als Erstes genannt. Ohne ein gutes Umfeld wäre es uns nicht möglich, die Energie aufzubringen, die es für das Schreiben eines Buches braucht. Viele der hier präsentierten Ideen haben ihren Ursprung bei Christiane Floyd und Heinz Züllighoven. Danke, dass wir von euch lernen durften!

Unsere beiden Kollegen Stefan Hofer und Sönke Magnussen haben unsere Texte gereviewt und uns mit ihren kritischen, konstruktiven Anmerkungen zu einem deutlich besseren Ergebnis verholfen. Unsere Lektorin Christa Preisendanz vom dpunkt.verlag hat uns wieder einmal hervorragend betreut und motiviert. Danke, Christa!

Unsere Familien haben unsere Fokussierung auf ein weiteres Buch voller Verständnis begleitet und uns Freiraum und Unterstützung gegeben. Henning dankt seiner Frau Julia und seinen Kindern Lennart, Joris, Friedo, Tara, Aurelia und Viola für ihre großartige Unterstützung. Carola dankt ihrem Mann, ihrem Sohn und ihren Freunden für einfach alles. Herzlichen Dank an euch alle!

Was die geschlechtergerechte Sprache angeht, haben wir uns für dieses Buch entschieden, abwechselnd pro Kapitel die weibliche und männliche Form zu verwenden. Dann bleibt das Lesen flüssig und jede und jeder kann sich hoffentlich inkludiert fühlen.

Carola Lilienthal und Henning Schwentner
Hamburg, April 2023
www.domain-driven-transformation.com

1. Programmierer lieben es ja, darüber fachzusimpeln, welche Programmiersprache die beste ist. Wenn es um das Thema »Bauen von monolithischer Legacy-Software« geht, gibt es einen klaren Gleichstand: Das kann man mit allen einfach schaffen.