
Vorwort

Warum macht das Bauen von Modellen so viel Spaß und warum ist es so eine befriedigende Tätigkeit? Schon als Kind habe ich es geliebt, Modelle zu bauen. Damals habe ich hauptsächlich Modelle von Autos und Flugzeugen angefertigt. Ich frage mich, wo Lego damals war. Im Leben meines Sohnes hat Lego jedenfalls eine große Rolle gespielt, schon seit er sehr klein war. Mit diesen kleinen Bausteinen sich Modelle auszudenken und zu bauen ist sehr faszinierend. Man kann mit kleinen Modellen anfangen, und es scheint, als könnte man seine Ideen nahezu unendlich erweitern.

Sie haben sicherlich auch das eine oder andere Modell in Ihrer Jugend gebaut.

Modelle kommen in unserem Leben in sehr vielen Situationen vor. Wenn Sie Spaß an Brettspielen haben, benutzen Sie Modelle. Wir haben Modelle von Immobilien und Grundbesitzern oder Modelle von Inseln und Schiffbrüchigen oder Modelle von Territorien und Baumaßnahmen oder von wer weiß was allem. Auch Videospiele sind Modelle. Vielleicht modellieren sie eine Fantasiewelt mit bunten Charakteren, die fantastische Rollen spielen. Kartenspiele und deren Varianten bilden Machtspiele als Modell ab. Wir benutzen die ganze Zeit Modelle und wahrscheinlich so oft, dass wir den meisten Modellen nicht die wohlverdiente Anerkennung geben. Modelle sind einfach Teil unseres Lebens.

Aber warum? Jeder Mensch hat einen Lernstil. Es gibt eine Vielzahl von Lernstilen, aber die drei am häufigsten diskutierten sind auditives, visuelles und taktiles Lernen. Das auditive Lernen basiert auf Hören und Zuhören. Die visuellen Lerner verstehen durch Lesen und Sehen von Bildern. Die taktilen Lerner erfassen Neues dadurch, dass sie während des Lernprozesses Dinge berühren. Es ist interessant, dass die verschiedenen Individuen jeweils einen bestimmten Lernstil bevorzugen, und zwar so stark, dass der Lernende Probleme mit den anderen Arten des Lernens hat. Zum Beispiel merken sich taktile Lerner wahrscheinlich, was sie getan haben, haben aber vielleicht Probleme, sich zu erinnern, was während des Prozesses gesagt wurde. Man würde annehmen, dass beim Modellbilden visuelle und taktile Lerner einen riesigen Vorteil gegenüber den auditiven Lernern hätten, weil das Bauen von Modellen vor allem visuelle und taktile Stimulationen zu enthalten scheint. Wenn ein Team von Modellentwicklern allerdings in seinem

Prozess der Modellbildung mündliche Kommunikation einsetzt, sieht es gleich ganz anders aus. Mit anderen Worten bietet das Bilden von Modellen die Möglichkeit, die Lernstile der meisten Individuen zu berücksichtigen.

Software beeinflusst und unterstützt unser Leben immer stärker. Wenn wir schon so eine angeborene Neigung dazu haben, durch das Bilden von Modellen zu lernen, wäre es da nicht nur natürlich, auch Software auf Basis eines Modells zu entwickeln? Tatsächlich scheint Software aus Modellen zu bauen irgendwie, nun ja, menschlich zu sein. Also sollten wir Modelle als Basis von Software verwenden. Mir scheint, dass Menschen von Natur aus ein Talent zum Bilden von Softwaremodellen haben.

Ich möchte Ihnen unbedingt helfen, beim Modellieren von Software diese angeborenen Talente so gut wie möglich zu nutzen, indem Sie einige der besten verfügbaren Softwaremodellierungstechniken verwenden können. Diese Techniken sind unter dem Namen »Domain-Driven Design« oder kurz DDD als ein Werkzeugkasten zusammengefasst. Dieser Werkzeugkasten, eigentlich eine Menge von Mustern, wurde zuerst von Eric Evans in seinem Buch *Domain-Driven Design: Tackling Complexity in the Heart of Software* [Evans 2004] aufgeschrieben. Meine Vision ist es, möglichst jedem DDD zugänglich zu machen. Um es zuge-spitzt zu formulieren: Ich will DDD den breiten Massen bekannt machen. DDD verdient es, überall bekannt zu sein. Es ist der Werkzeugkasten, den modellorientierte Menschen für ihre Arbeit verdient haben, um ihre fortschrittlichsten Softwaremodelle zu erschaffen. Mein Ziel ist es, mit diesem Buch das Lernen und Anwenden von DDD so einfach wie möglich zu machen und DDD dem größtmöglichen Fachpublikum näherzubringen.

Für auditive Lerner hält DDD die Aussicht bereit, durch die Kommunikation im Team das Entwickeln eines Modells auf Basis einer *Ubiquitous Language* (dt.: *allgegenwärtige Sprache*) zu lernen. Für visuelle und taktile Lerner können die DDD-Techniken sehr anschaulich und anfassbar eingesetzt werden, wenn das Team sowohl strategisch als auch taktisch modelliert. Das gilt insbesondere für das Zeichnen von *Context Maps* (dt.: *Kontextlandkarten*) und das Modellieren des Geschäftsprozesses mit *Event Storming*. Deshalb glaube ich, dass DDD jedem helfen kann, der durch das Bilden von Modellen lernen und Großartiges erreichen will.

Für wen ist dieses Buch?

Dieses Buch ist für jeden, der daran interessiert ist, die wichtigsten DDD-Aspekte und -Techniken möglichst schnell zu lernen. Die typischen Leser sind Softwarearchitekten und Softwareentwickler, die DDD in Projekten in der Praxis einsetzen wollen. Softwareentwickler erkennen die Schönheit von DDD in der Regel sehr schnell und werden von seinen machtvollen Techniken angezogen. Aber ich habe das Thema auch schon Vorständen, Fachexperten, Managern, Business-Analysten, Informationsarchitekten und Testern vermittelt. Tatsächlich ist die Gruppe derer, die vom Lesen des Buches profitieren können, nicht auf die IT-Industrie und auf Forschungs- und Entwicklungsabteilungen beschränkt.

Wenn Sie Berater sind und mit einem Kunden arbeiten, dem Sie die Verwendung von DDD empfohlen haben, bietet dieses Buch eine Möglichkeit, die wichtigsten Beteiligten schnell mit Wissen zu versorgen. Wenn Sie Entwickler in Ihrem Projekt haben – vielleicht von Junior-, mittlerem oder sogar Seniorlevel –, die nicht vertraut mit DDD sind, aber es sehr bald verwenden müssen, stellen Sie sicher, dass sie dieses Buch lesen. Durch das Lesen dieses Buches werden alle Projektbeteiligten und Entwickler auf jeden Fall das Vokabular lernen und verstehen, wie man die grundsätzlichen DDD-Techniken einsetzt. Dies ermöglicht allen Beteiligten, sinnvoll Wissen auszutauschen, während sie das Projekt vorantreiben.

Was immer Ihre Erfahrung und Rolle ist, lesen Sie dieses Buch und verwenden Sie dann DDD in einem Projekt. Danach lesen Sie das Buch noch einmal und schauen, was Sie aus Ihren Erfahrungen lernen können und wo Sie sich in der Zukunft verbessern können.

Was dieses Buch enthält

Das erste Kapitel, »DDD für mich«, erklärt, was DDD für Sie und Ihre Organisation tun kann, und bietet einen detaillierteren Überblick darüber, was Sie lernen werden und warum das wichtig ist.

Kapitel 2, »Strategisches Design mit Bounded Contexts und der Ubiquitous Language«, führt das strategische Design von DDD ein und lehrt die Eckpfeiler von DDD, *Bounded Contexts* (dt.: *begrenzte Kontexte*) und die *Ubiquitous Language*. Kapitel 3, »Strategisches Design mit Subdomains«, erklärt *Subdomains* (dt.: *Teildomänen, Subdomänen*) und wie man sie verwenden kann, um mit der Komplexität bei der Integration von existierenden Altsystemen umzugehen, während man seine neuen Anwendungen modelliert. Kapitel 4, »Strategisches Design mit Context Mapping«, stellt die verschiedenen Wege vor, mit denen Teams strategisch zusammenarbeiten können, und vermittelt Wege, wie die Software verschiedener Teams miteinander integriert werden kann. Das nennt man *Context Mapping* (dt.: *Abilden von Kontexten*).

Kapitel 5, »Taktisches Design mit Aggregates«, lenkt Ihre Aufmerksamkeit auf das taktische Modellieren mit *Aggregates* (dt.: *Aggregate*). Ein wichtiges und mächtiges Werkzeug für taktisches Modellieren sind *Domain Events* (dt.: *Domänenereignisse, fachliche Ereignisse*), die das Thema von Kapitel 6, »Taktisches Design mit Domain Events«, sind.

In Kapitel 7, »Beschleunigungs- und Managementtechniken«, hebt das Buch schließlich einige Techniken für die Beschleunigung und das Management von Projekten hervor. Sie können den Teams helfen, eine hohe Schlagzahl (engl.: *cadence*) zu erreichen und zu erhalten. Diese beiden Themen sind in anderen DDD-Quellen bisher selten, wenn überhaupt, diskutiert und werden dringend benötigt, wenn man beschlossen hat, DDD in der Praxis umzusetzen.

Konventionen

Es gibt nur wenige Konventionen, die man beim Lesen im Gedächtnis behalten sollte. Alle DDD-Begriffe, die ich verwende, sind kursiv gesetzt. Zum Beispiel werden Sie von *Bounded Contexts* und *Domain Events* lesen¹. Eine weitere Konvention ist, dass sämtlicher Quellcode in nichtproportionaler Schrift gesetzt ist.

Was in diesem Buch besonders stark betont wird und was Ihrem Gehirn hoffentlich gefällt, ist visuelles Lernen durch zahlreiche Diagramme und Abbildungen. Sie werden bemerken, dass es keine Abbildungsnummern im Buch gibt, weil ich Sie nicht verwirren wollte. Die Abbildungen und Diagramme stehen immer vor dem Text, der sie erklärt, was bedeutet, dass die Grafiken die Gedanken einführen, während man sich durch das Buch arbeitet. Das heißt, immer wenn Sie Text lesen, können Sie die vorangehende Abbildung zur visuellen Unterstützung verwenden.

1. Anm. d. Übersetzer: Wie in der Vorbemerkung beschrieben, verwenden wir im deutschen Text – wie in der Praxis üblich – die englischen DDD-Begriffe. Jeweils beim ersten Auftreten eines Begriffes geben wir eine Übersetzung auf Deutsch, um den Begriff verständlich zu machen. Zum schnellen Nachschlagen gibt es das Glossar auf den Umschlaginnenseiten.