

Vorwort

Missverständnisse zwischen Softwareentwicklern und Personen aus Fachabteilungen sind ein häufiges Problem. Schlechte Kommunikation ist eine Plage, die Projekte scheitern lässt. Domain Storytelling bietet Abhilfe, denn diese Technik verwandelt Domänenwissen in effektive Unternehmenssoftware. Domain Storytelling bringt Fachexperten, Softwareentwickler, User Experience Designer, Product Owner, Produktmanager und Business-Analysten zusammen. Sie lernen voneinander, indem sie Geschichten erzählen und sie in leicht verständlichen Bildern darstellen.

Wer sollte dieses Buch lesen?

Das Buch richtet sich an alle, die an der Softwareentwicklung beteiligt oder daran interessiert sind, auch an nicht technische Leserinnen und Leser. Es gibt nur wenige Codebeispiele und der Großteil dieses Buches ist ohne Vorkenntnisse in Softwareentwicklung verständlich. Wo es sinnvoll ist, verweisen wir auf empfohlene Lektüre.

Neben Fachleuten aus der Softwareentwicklung ist das Buch auch für Führungskräfte, Abteilungs- und Teamleiter sowie für Fachexperten relevant. Sie können das Buch nutzen, um ihre Geschäftsprozesse besser zu verstehen und die Kommunikation mit ihren (IT-)Kolleginnen und Kollegen zu verbessern.

Dieses Buch in der Softwareentwicklungslandschaft

Dieses Buch ist als praktischer Leitfaden gedacht. Wir wollten es kurz halten und mussten der Versuchung widerstehen, ein Buch über unsere Sicht auf Softwareentwicklung im Allgemeinen zu schreiben. Themen wie Domain-Driven Design, fachliche Modellierung, Anforderungen und Agilität sind für Domain Storytelling relevant und werden in diesem Buch erwähnt. Wir haben versucht, nicht zu tief in diese Themen einzutauchen, weil andere Autoren das viel besser gemacht haben, als wir es könnten. Wer neugierig geworden ist, findet hier eine (unvollständige) Liste von Autorinnen und Autoren, die uns beeinflusst haben:

■ Heinz Züllighoven et al.

Der *Werkzeug & Material-Ansatz* beschrieben im *Object-Oriented Construction Handbook* [Züllighoven 2004] ist wichtig für unsere Sichtweise der Softwareentwicklung. Er fördert die anwenderorientierte Softwareentwicklung und basiert auf den starken Metaphern *Werkzeug* und *Material*. Das Buch wurde erstmals in den späten 1990er-Jahren unter dem Originaltitel *Das objektorientierte Konstruktionshandbuch nach dem Werkzeug & Material-Ansatz* veröffentlicht und behandelt Themen wie iterativ-inkrementelle Entwicklung, Modellierung, Szenarien und Muster für die Softwarekonstruktion.

■ Eric Evans

Domain-Driven Design (DDD) [Evans 2004] teilt viele Merkmale mit dem *Werkzeug & Material-Ansatz*, obwohl die beiden unabhängig voneinander entwickelt wurden. Die Konzepte *Bounded Contexts* und *Ubiquitous Language* haben uns sofort überzeugt. DDD eröffnete einen neuen Anwendungsbereich für Domain Storytelling und beeinflusste die Richtung, in die wir die Methode in den letzten Jahren vorangetrieben haben.

■ Kent Beck et al.

Das *Agile Manifest* [Beck et al. 2001] schätzt »Individuen und Interaktionen mehr als Prozesse und Werkzeuge« und »funktionierende Software mehr als umfassende Dokumentation«. Bei Domain Storytelling geht es um Individuen und ihre Interaktionen auf zwei Ebenen: (1) Domain Stories stellen Menschen und ihre Zusammenarbeit vor, und (2) der Workshop, in dem diese Domain Stories erzählt werden, bringt Menschen zusammen, um miteinander zu interagieren. *Extreme Programming (XP)* brachte das Konzept der Geschichten in die Softwareentwicklung und führte *User Stories* in die agile Welt ein [Beck 2000], [Beck & Andres 2005] und [Cohn 2004].

■ Alistair Cockburn

Writing Effective Use Cases [Cockburn 2001] ist eines unserer Lieblingsbücher über Anforderungen. Selbst wenn man keine Use Cases anwendet, ist das Buch sehr lesenswert. Es zeigt, wie man einen agilen Pfad von der Domäne bis zur Software beschreitet. Wir haben die Idee der *Goal Levels* entlehnt und sie auf Domain Storytelling angewendet.

■ Gojko Adzic

Specification by Example [Adzic 2011] hebt die Bedeutung von Gesprächen in der Softwareentwicklung hervor. Wir betrachten Softwareentwicklung als eine Reihe von Gesprächen über die Domäne und die Anforderungen, unterstützt durch verschiedene »Gesprächsmethoden« wie Domain Storytelling. Wir teilen Adzics Ansichten über Anforderungen und die Vorteile der Verwendung von Beispielen.

■ Christiane Floyd

Wir hatten das Vergnügen, von Professorin Floyd an der Universität Hamburg zu lernen. Sie forschte und lehrte (unter anderem) zu Modellierungstheorie, den Grenzen der Modellierung, soziotechnischen Aspekten der Modellierung und Softwareent-

wicklung und partizipativem Design. Wer neugierig ist, sollte sich z.B. »Software Development as Reality Construction« [Floyd 1992] ansehen und nach Abhandlungen über ihren STEPS-Ansatz suchen, ein Vorreiter für das, was später als agile Softwareentwicklung bekannt wurde.

Was dieses Buch behandelt

Dieses Buch ist in zwei Teile gegliedert: Teil I erklärt die Methode, und Teil II beschreibt, wie Domain Storytelling für bestimmte Einsatzzwecke genutzt und angepasst wird.

Teil I, »Domain Storytelling – die Methode«, vermittelt alles, was Sie wissen müssen, um mit Domain Storytelling etwas über eine Domäne zu lernen.

Kapitel 1, »Einführung«, erklärt, was Domain Storytelling eigentlich ist. Außerdem stellen wir eine Fallstudie vor, die einen ersten Eindruck von der Methode bietet und darlegt, warum sie nützlich ist.

Kapitel 2, »Die Bildsprache«, erläutert die grafische Notation. Um Domain Stories visuell festzuhalten, sind mehrere Icons und auch Regeln erforderlich, um diese zu kombinieren. Wir zeigen, was wir für guten Sprachstil halten und welche Fallstricke zu vermeiden sind. Wer noch nie Domain Storytelling ausprobiert hat, sollte das nach Kapitel 2 tun. Dazu nimmt man ein Blatt Papier und einen Stift zur Hand und modelliert einen Arbeitsablauf, mit dem man vertraut ist. Bevor Sie Domain Storytelling jedoch in einer Workshop-Situation ausprobieren, sollten Sie den Rest von Teil I lesen.

Kapitel 3, »Szenariobasiertes Modellieren«, behandelt einen der Hauptunterschiede zwischen Domain Stories und anderen Sprachen zur Geschäftsprozessmodellierung: In jeder Domain Story geht es um *einen* Fall. Wir zeigen, welche Fälle man modellieren sollte und wie man den Überblick behält.

Kapitel 4, »Scope«, zeigt, wie detailliert die Domain Stories sind, ob sie beschreibend oder explorativ sind und wie viele technische Informationen sie enthalten. All diese Faktoren müssen Sie mit jeder neuen Domain Story berücksichtigen. Dieses Kapitel hilft dabei, den richtigen Scope zu wählen.

Kapitel 5, »Modellierungswerkzeuge«, fasst unsere Erfahrungen mit verschiedenen Modellierungswerkzeugen zusammen. Wir empfehlen, welche Werkzeuge in welchen Situationen nützlich sind, beschreiben ihre Stärken und Schwächen und geben praktische Tipps für müheloses Modellieren.

Kapitel 6, »Das Workshop-Format«, zeigt, dass Domain Storytelling am besten funktioniert, wenn es für Collaborative Modeling (also gemeinschaftliches Modellieren von Fachleuten und Entwicklungsteam) eingesetzt wird. Hier lernen Sie, wie Sie einen Workshop vorbereiten, durchführen und nachbereiten. Dieses Kapitel hilft Ihnen dabei, eine gute Moderatorin oder ein guter Moderator zu werden.

Kapitel 7, »Das Verhältnis zu anderen Modellierungsmethoden«, behandelt andere Modellierungsmethoden und Workshop-Formate und wie Domain Storytelling mit ihnen kombiniert werden kann. Dieses Kapitel hilft Ihnen, das richtige Werkzeug für die richtige Aufgabe auszuwählen.

Teil II, »Domain Storytelling für verschiedene Einsatzzwecke nutzen und anpassen«, beschäftigt sich mit den verschiedenen Problembereichen, für die Domain Storytelling genutzt werden kann. Auch wenn wir in allen Kapiteln von Teil II dasselbe Beispiel verwenden, müssen die Kapitel nicht der Reihe nach gelesen werden. Suchen Sie sich die für Sie relevanten Einsatzzwecke aus.

Kapitel 8, »Fallstudie – Alphorn Auto Leasing GmbH«, stellt eine zweite, umfangreichere Fallstudie vor.

Kapitel 9, »Fachsprache lernen«, zeigt auf, dass die Sprache der Fachexperten zu sprechen der Schlüssel zu effektiven Gesprächen über Geschäftsprozesse und Anforderungen ist. Dieses Kapitel ist interessant, wenn man neu in einer Domäne ist, wenn man in der Software, an der man arbeitet, keine echten Begriffe aus der Domäne verwendet und das ändern will, oder wenn man in einem Unternehmen arbeitet, in dem es keine echte Fachsprache gibt und gewünscht ist, dass sich eine solche entwickelt.

Kapitel 10, »Grenzen finden«, beschäftigt sich damit, dass viele Domänen zu groß sind, um als Ganzes verstanden und modelliert werden zu können. Eine Domäne muss in überschaubare Einheiten aufgeteilt werden. Dieses Kapitel ist interessant, wenn Sie mit einem Monolithen zu kämpfen haben und ihn umstrukturieren oder zerlegen möchten oder Microservices entwerfen. Im Domain-Driven Design hilft dieses Kapitel, Bounded Contexts zu identifizieren. Das Kapitel ist auch nützlich, wenn das Entwicklungsteam zu groß geworden ist, um effizient zu arbeiten, oder wenn Sie bereits mehr als ein Entwicklungsteam haben und die Arbeitsaufteilung für diese Teams besser organisieren möchten.

Kapitel 11, »Mit Anforderungen arbeiten«, zeigt, wie Sie die Kluft zwischen Domänenwissen und Anforderungen überbrücken. Wir erklären, wie Anforderungen aus Domain Stories abgeleitet werden, um über Prioritäten und tragfähige Produkte diskutieren zu können. Dieses Kapitel ist interessant für Rollen wie Product Owner, Produktmanager, Business-Analyst, Requirements Engineer oder Entwickler in einem cross-funktionalen Team, das seine eigenen Anforderungen analysiert.

Kapitel 12, »Modellieren in Code«, beschäftigt sich mit dem Übergang von der Modellierung mit Diagrammen und Klebezetteln zur Modellierung in Programmiersprachen, um Software zu entwickeln. In diesem Kapitel wird vorgestellt, wie man von der visuellen Modellierung zum Code übergeht.

Kapitel 13, »Organisatorische Veränderungen unterstützen«, befasst sich mit dem Ziel eines neuen Softwaresystems, das in der Regel darin besteht, die Arbeit einfacher, schneller und effizienter (oder kurz: besser) zu machen. Dieses Ziel wird nicht dadurch erreicht, dass schlechte manuelle Prozesse digitalisiert werden. Genauso wenig wird aus einem Haufen von Anforderungen auf magische Weise ein reibungslos funktionierender Geschäftsprozess. Um gute Unternehmenssoftware zu entwickeln, ist mehr zu tun, als nur den aktuellen Zustand in Software zu implementieren. Man muss auch die zukünftige Arbeitsweise entwerfen. Domain Stories helfen dabei und veranschaulichen, wie die neue Software die Arbeitsweise der Menschen verändern wird. Dieses Kapitel ist interessant, wenn Sie Geschäftsprozesse optimieren, neue Software einführen oder Veränderungen in Geschäftsprozessen diskutieren und vorantreiben wollen.


Kapitel 14, »Make or Buy-Entscheidungen und Auswahl von Standardsoftware«, weist darauf hin, dass nicht jede Software individuell entwickelt werden muss. Für viele Domänen gibt es Software von der Stange. Domain Stories können bei der Entscheidung helfen, ob ein neues Softwaresystem entwickelt oder gekauft werden soll. Wenn ein Kauf infrage kommt, muss man sich häufig zwischen Produkten mehrerer Anbieter entscheiden. Auch hier können Domain Stories hilfreich sein, um eine Auswahl zu treffen.



Kapitel 15, »Schatten-IT finden«, zeigt auf, dass die Schatten-IT im Weg steht, wenn es darum geht, IT-Landschaften zu konsolidieren oder die Digitalisierung voranzutreiben. Jedes Unternehmen ab einer bestimmten Größe nutzt Software, die der zentralen IT-Abteilung nicht bekannt ist. Diese kleinen Lösungen, die in den Fachabteilungen laufen und wenig beachtet werden, sind manchmal sogar geschäftskritisch. Fachexperten nutzen oft Schatten-IT, ohne sich dessen bewusst zu sein, sodass sie leicht übersehen wird. Domain Stories können der IT-Abteilung und dem Management helfen, diese Schatten-IT zu finden und die gesamte IT-Landschaft zu überblicken.

Kapitel 16, »Ausblick und Fazit«, wirft einen Blick in die Zukunft von Domain Storytelling und fasst seine Essenz zusammen.

Konventionen






Wenn wir einen neuen Begriff definieren, setzen wir ihn fett, z. B. **Akteur**. Begriffe, die von anderen Autoren definiert wurden, werden kursiv gesetzt, wenn wir sie zum ersten Mal verwenden, z. B. *Bounded Contexts*. Scope-Faktoren erscheinen in Kapitälchen, z. B. GROBGRANULAR. Wörter aus Fallstudien werden in Anführungszeichen gesetzt, z. B. »Kinobesucher«. Code wird in fester Breite gesetzt, wie die Klasse `Vorstellung`.

Wichtige Hinweise sind mit einem Glühbirnen-Icon gekennzeichnet: . Wenn wir konkrete Ratschläge geben, sind unsere Tipps mit einem Häkchen-Icon  versehen.

Wir präsentieren zwei Fallstudien und mehrere »Geschichten aus der Praxis«. Abschnitte zur Fallstudie Alphorn Auto Leasing GmbH sind durch ein Icon mit einem Auto gekennzeichnet: . Die Geschichten aus der Praxis werden in Einschüben erzählt, die mit einem Sonnenblumen-Icon  gekennzeichnet sind.

Legende für die »Opening Stories«

Als Modellierer konnten wir der Versuchung nicht widerstehen, Domain Storytelling selbst mit Domain Stories zu beschreiben. Deshalb beginnen mehrere Kapitel in diesem Buch mit einer »Opening Story« – einer kurzen Domain Story, die erklärt, worum es in dem Kapitel geht. Die Legende in Tabelle 1 soll helfen, die Icons zu deuten, die wir in diesen Geschichten verwenden.

Icon	Bedeutung
	Eine Domain Story als Diagramm
	Jede andere Art von Diagramm
	Bausteine einer Domain Story
	Aktivitäten einer Domain Story
	Ein Text wie eine User Story oder eine Quellcodedatei o.Ä.

Tab. 1 In »Opening Stories« verwendete Icons

Ergänzende Materialien

Auf www.domainstorytelling.org/book haben wir eine begleitende Website für dieses Buch erstellt [DomainStorytelling BookWebsite]. Sie enthält die Domain Stories, die wir in diesem Buch vorstellen, auf Deutsch und Englisch. Die Beispiele wurden mit Egon.io erstellt, einem Open-Source-Modellierungstool, das unsere Firma WPS entwickelt hat [Egon.io Website]. Man kann die Quelldateien aus diesem Buch in Egon.io importieren und die Domain Stories »abspielen« (Anweisungen finden sich auf der Website).

Die begleitende Website enthält auch die Bibliografie dieses Buches mit anklickbaren Links zu Online-Ressourcen.

Außerdem finden Sie auf www.domainstorytelling.org Links zu nützlichen Materialien, darunter eine Sammlung von Videos und Artikeln [DomainStorytelling Website].

Über das Cover

Die englische Ausgabe dieses Buches ist in der Addison-Wesley Signature Series von Vaughn Vernon erschienen. Die Titelbilder der Bücher dieser Buchreihe haben ein organisches Thema. Wir haben uns darüber gefreut, denn auch Domain Stories entwickeln sich auf organische Weise. Wie die Sonnenblumen, die auf dem Cover zu sehen sind, fängt eine Domain Story aus einem kleinen Samen an, wächst heran und blüht, wenn alles gut geht, schließlich wunderschön auf. Die Beziehung zwischen Domäne und Domain Story ist ähnlich wie die zwischen Sonne und Sonnenblume:

- Eine Sonnenblume kann ohne Sonne nicht existieren. Genauso wird eine Domain Story ohne den Kontakt zur Domäne verkümmern.
- Die Sonnenblume ähnelt der Sonne, so wie eine Domain Story ein Abbild der Domäne ist.

- Während des Tages dreht die Sonnenblume ihren Kopf, um der Sonne zu folgen. Genauso folgt die Domain Story der Domäne.
- Eine Sonnenblume lebt oft zusammen mit vielen anderen Sonnenblumen auf einem Feld. Genauso steht eine Domain Story normalerweise nicht allein, sondern in Gesellschaft anderer Stories.

All diese Eigenschaften machen die Sonnenblume nicht nur zu einem optisch ansprechenden Titelbild, sondern auch zu einem Symbol für Domain Storytelling.

In diesem Sinn wünschen wir eine anregende Lektüre des *Sonnenblumenbuches* und viel Spaß beim Modellieren!

Stefan Hofer und Henning Schwentner
Hamburg, im März 2023
www.domainstorytelling.org