

Ramona Greiner
David Berger
Matthias Böck

Analytics und Artificial Intelligence

Datenprojekte mehrwertorientiert, agil
und nachhaltig planen und umsetzen



Springer Gabler

Analytics und Artificial Intelligence

Ramona Greiner • David Berger
Matthias Böck

Analytics und Artificial Intelligence

Datenprojekte mehrwertorientiert, agil
und nachhaltig planen und umsetzen

Ramona Greiner
FELD M GmbH
München, Deutschland

David Berger
Berylls Mad Media GmbH
München, Deutschland

Matthias Böck
FELD M GmbH
München, Deutschland

ISBN 978-3-658-38158-5 ISBN 978-3-658-38159-2 (eBook)

<https://doi.org/10.1007/978-3-658-38159-2>

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Springer Gabler

© Der/die Herausgeber bzw. der/die Autor(en), exklusiv lizenziert an Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2022

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Verlags. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von allgemein beschreibenden Bezeichnungen, Marken, Unternehmensnamen etc. in diesem Werk bedeutet nicht, dass diese frei durch jedermann benutzt werden dürfen. Die Berechtigung zur Benutzung unterliegt, auch ohne gesonderten Hinweis hierzu, den Regeln des Markenrechts. Die Rechte des jeweiligen Zeicheninhabers sind zu beachten.

Der Verlag, die Autoren und die Herausgeber gehen davon aus, dass die Angaben und Informationen in diesem Werk zum Zeitpunkt der Veröffentlichung vollständig und korrekt sind. Weder der Verlag, noch die Autoren oder die Herausgeber übernehmen, ausdrücklich oder implizit, Gewähr für den Inhalt des Werkes, etwaige Fehler oder Äußerungen. Der Verlag bleibt im Hinblick auf geografische Zuordnungen und Gebietsbezeichnungen in veröffentlichten Karten und Institutionsadressen neutral.

Planung/Lektorat: Rolf-Guenther Hobbeling

Springer Gabler ist ein Imprint der eingetragenen Gesellschaft Springer Fachmedien Wiesbaden GmbH und ist ein Teil von Springer Nature.

Die Anschrift der Gesellschaft ist: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

*Für unsere Familien und Freund:innen,
Lutz Wiechert
und alle, deren Maxime ist:
Be best, not in, but for the world.*

Vorwort

„Computer sind nicht dazu da, uns Antworten zu geben, sondern sie sind Werkzeuge, um Fragen zu stellen“, schreibt James Bridle (2020, S. 15) in seinem Buch über die technologischen Dystopien unserer Zeit. Genau dieses Fragenstellen scheinen Menschen und Unternehmen aber vor allem in Bezug auf Technologie- und Tooleinsatz, Big Data und Artificial Intelligence¹ verlernt zu haben. Die Gründe dafür sind vielfältig. Bereits in der Ausbildung, in Schule und Studium, wird fast ausschließlich das Finden und Geben von Antworten trainiert, selten aber das Stellen von wirklich relevanten Fragen.

Das Tagesgeschäft in Unternehmen lässt häufig keine Zeit, mehrwertstiftende Fragestellungen zu entdecken oder gezielt zu entwickeln. Durch Routine und eingespielte Prozesse ergibt sich gar kein Bedürfnis danach, eine echte Problemstellung zu identifizieren. Man sieht den Wald vor lauter Bäumen nicht, verliert sich im Kleinklein und es ist kaum möglich, einen Schritt zurück zu machen und auf ein Setup, sei es technisch oder organisatorisch, mit unverstelltem Blick zu schauen. Natürlich finden in vielen Firmen Ideenfindungsworkshops statt, doch Kreativität auf Knopfdruck funktioniert nicht immer und für bahnbrechende Ideen braucht es mehr als einen bestimmten Zeitslot und spannende Workshopmethoden. Es ist eine Frage des eigenen, persönlichen Mindsets, aber auch des Arbeitsumfelds, ob die Beschäftigten zu Kreativität und kritischem Denken angeregt werden.

In den häufig noch streng getrennten Abteilungen eines Unternehmens herrscht überwiegend Silodenken vor und es fehlt die Vernetzung. Nur durch Austausch und Durchlässigkeit kann das Entwickeln gemeinsamer Fragestellungen gefördert werden, indem Pilotprojekte, Analysen und Reportings oder schlichtweg Problemstellungen geteilt und polyperspektivisch betrachtet werden.

Gerade im IT-Umfeld ist es schwierig, den Überblick über technische Entwicklungen zu behalten. Das Entdecken neuer Möglichkeiten ist eher Heuristik als Strategie: Oft stößt man zufällig über Branchenkontakte oder abseits des Arbeitsplatzes auf neue Methoden, Tools oder technische Entwicklungen und nicht, weil man für ein spezifisches Problem

¹Warum wir durchgehend von Artificial Intelligence (AI) und nicht dem deutschen Pendant „Künstliche Intelligenz (KI)“ sprechen, haben wir in Kap. 3 für Sie erläutert.

eine Lösung sucht oder einen Prozess etabliert hat, der die relevanten Entwicklungen der Tech-Welt systematisch beobachtet.

Wenn wir den Bereich Analytics und AI betrachten, ist ein – wenn auch umstrittener – Treiber von innovativen Projekten der unbedingte Wille vieler, sich im Unternehmen oder als Unternehmen mit eigenen Leuchtturm-Prestige-Projekten ein Denkmal zu setzen und Erste:r zu sein: Erste:r beim Einsatz neuer Technologien oder beim umfassenden Tracking aller Unternehmensplattformen, Erste:r mit einer spannenden Keynote auf einer Konferenz, Erste:r, der oder die in einschlägigen Fachmagazinen für Innovationskraft und Adaptionsfähigkeit der neuen Technologien gelobt wird. Echte Early-Adopter eben.

Doch der Preis dafür ist oft hoch: Wirtschaftlich, sozial und ökologisch. Viele Firmen sammeln unfassbar viele Daten, die sie niemals verwenden. Das Tracking der unzähligen Websites und Apps ist ressourcenintensiv in der Implementierung, im Betrieb und in der Wartung. Die Daten zu übermitteln und zu speichern kostet Geld und Energie. Meist ist die Data Literacy, also die Kompetenz der Mitarbeitenden, die mit den Daten arbeiten sollen, bei weitem nicht so hoch, als dass die Daten unmittelbar gewinnbringend genutzt werden könnten. Von komplexeren Anwendungen ganz zu schweigen. Soweit die Worst-Case-Szenarien des Status quo.

Web- und App-Analytics sowie AI-Anwendungen sind aus dem heutigen Online-Business-Alltag nicht mehr wegzudenken und werden immer wichtiger: Wer in Sachen datengetriebene Organisation und datengetriebenes Marketing nicht *mit der Zeit* geht, *geht* mit der Zeit. Der Wettbewerb ist hart umkämpft und immer mehr Geschäftsmodelle basieren auf der Verarbeitung von Daten. Wie Sie also Analytics- und AI-Projekte wirtschaftlich und in mehrfacher Hinsicht ressourcenschonend umsetzen, darum geht es in diesem Buch. Wir geben Ihnen Strategien an die Hand, mit welchen Methoden Sie in der Lage sind, die richtigen und wertstiftenden Fragen für Ihr Geschäftsmodell zu stellen, wie Sie die Technologien und Daten so einsetzen, dass sie echten Mehrwert generieren und wie Sie die Projekte effektiv, effizient und gewinnbringend umsetzen.

Das Fundament bilden agile Methoden und Design Thinking. Aus der Software- und Produktentwicklung stammend, haben wir das agile Vorgehen in unsere alltäglichen Analytics- und Daten-Projekte überführt und – gemäß dem agilen Ansatz – nachjustiert, erweitert und iterativ auf die spezifischen Anforderungen angepasst. Wie beim Design Thinking auch, sind unsere Ansätze nutzer:innenzentriert, zielorientiert und flexibel einsetzbar. Design-Thinking-Methoden liefern dabei die Anhaltspunkte, um Probleme und Fragestellungen zu identifizieren und dafür maßgeschneiderte Lösungen zu finden, die nicht dem Selbstzweck dienen. Für eine US-amerikanische Perspektive aus der Praxis haben wir im Dezember 2021 mit den beiden Product Evangelists und Autoren John Cutler und Adam Greco gesprochen. Deren Zitate ergänzen unsere Ausführungen an mehreren Stellen des Buches.

Doch Big Data, Analytics und AI stellen uns nicht nur vor methodische und technologische Herausforderungen, sondern auch vor rechtliche, moralische und ökologische. Spätestens seit Inkrafttreten der DSGVO im Jahr 2018 ist Datenschutz in aller Munde und branchenübergreifend stellen wir ein Bedürfnis der Kund:innen und Endanwender:innen

nach nachhaltigen Produkten und vertrauenswürdigen Unternehmen fest. Datenprojekte sollten heute nicht mehr ohne die ethischen, rechtlichen und ökologischen Implikationen gedacht werden, weshalb wir auch hierzu die relevantesten Themen für Sie in einem Kapitel behandeln.

In diesem Buch geben wir Ihnen Hilfestellungen an die Hand, wie Sie die für Sie und Ihr Unternehmen zentralen Fragestellungen finden und beantworten können. Nach der Beschreibung der übergeordneten Frameworks, Funktionsweisen und Methoden widmen wir uns in einem Deep Dive dem „Data Value Loop“ und seinen Technologien. Am Ende stellen wir Ihnen die konkrete Umsetzung von Analytics- und AI-Projekten vor, damit Sie einen Hands-on-Einblick in die tägliche Praxis gewinnen und unsere Erkenntnisse für Ihre eigenen Vorhaben nutzen können.

Literatur

Bridle, J. (2020). New Dark Age: Der Sieg der Technologie und das Ende der Zukunft. München: Verlag C. H. Beck.

München, Deutschland

Ramona Greiner
David Berger
Matthias Böck

Inhaltsverzeichnis

1 Agile Basics	1
1.1 Agil ist überall	1
1.2 Wie alles begann: Eine kurze Geschichte von <i>Agile</i>	2
1.3 Agile Prinzipien als Erfolgsfaktoren	4
1.4 Agile = Scrum, oder etwa nicht?	14
1.5 (Fr)Agil? Kritik an agilen Prozessen	22
1.6 Warum Sie Ihre Analytics- und AI-Projekte agil angehen und umsetzen sollten.	27
Literatur.	33
2 Design Thinking und Data Thinking	37
2.1 Schlaflose Nächte und echter Mehrwert – Warum Design Thinking und für wen?	37
2.2 Was ist Design Thinking?	46
2.3 Data Thinking und der Double Diamond.	59
2.4 Verbindung von agilen Methoden und Design Thinking.	63
Literatur.	66
3 Artificial Intelligence	67
3.1 Der Data-Handshake: AI und ihre Verwandten	69
3.2 Deep Dive ML: Algorithmen, Modelle und ihre Anwendung	75
3.3 Starke und schwache AI.	96
3.4 Gibt es AI ohne Analytics?	98
3.5 Künstliche Intelligenz und ihre Herausforderungen in der Praxis. Gastbeitrag von Matthis Eicher	100
Literatur.	107
4 Ethische, rechtliche und ökologische Implikationen	109
4.1 Schreckgespenst Data Analytics?!	111
4.2 Schreckgespenst AI?!	115
4.3 Nachhaltige Daten und bessere Raten	126

4.4	Daten für den guten Zweck – ein Überblick. Gastbeitrag von Alexander Merdian-Tarko	129
	Literatur.	137
5	Der Data Value Loop.	139
5.1	Die Data Value Chain.	139
5.2	Technologie-Stack und Daten	155
5.3	Data-Driven Culture.	163
5.4	Data meets Agile: Die Datenwertschöpfung als Iteration	172
	Literatur.	184
6	Analytics in der Praxis	185
6.1	Konzeption – Wert definieren und übersetzen	186
6.2	Tracking – Daten erheben und bereitstellen	197
6.3	Reporting und Analyse.	205
6.4	Optimierung & Aktivierung.	213
6.5	Prozesse & Artefakte	215
	Literatur.	222
7	AI in der Praxis	223
7.1	Data Science und Agile – geht das überhaupt zusammen?	224
7.2	Exemplarische Projektdurchführung I – In drei Monaten zum Feedback-Analyse-Prototypen.	226
7.3	Exemplarische Projektdurchführung II – In sechs Monaten zum Pricing-Prototypen.	235
7.4	AI-Praxis: Quo vadis?	247
7.5	Die Bedeutung von Participatory Design für Anwendungen mit Künstlicher Intelligenz. Gastbeitrag von Prof. Dr. Doris Aschenbrenner	252
	Literatur.	259
8	</buch>: In the future, everyone will be a data scientist for 15 minutes.	263
	Glossar	265

Abkürzungsverzeichnis

AI	Artificial Intelligence
API	Application Programming Interface
ARIMA	Auto Regressive Integrated Moving Average
AWS	Amazon Web Services
CDP	Customer Data Platform
CDR	Corporate Digital Responsibility
CLV	Customer Lifetime Value
DL	Deep Learning
DMP	Data Management Platform
DSGVO	Datenschutzgrundverordnung
DWH	Data Warehouse
IoT	Internet of Things
JEDUF	Just Enough Design Upfront
KPI	Key Performance Indicator
ML	Machine Learning
NLP	Natural Language Processing
NPS	Net Promoter Score
PaaS	Platform as a Service
PDP	Product Detail Page
ROI	Return on Investment
SaaS	Software as a Service
SQL	Structured Query Language
SSOT	Single Source of Truth oder Golden Source
TTDSG	Telekommunikation-Telemedien-Datenschutz-Gesetz
UX/UI	User Experience/User Interface
WIP	Work in Progress

Über die Autoren



Dr. Ramona Greiner studierte Philosophie und Kunstgeschichte. Seit 2017 arbeitet sie als Digital Analytics und Data Ethics Consultant bei der Münchner Unternehmensberatung FELDM. Dort leitet sie internationale agile Kund:innenprojekte im Bereich Data & Analytics, hält Vorträge und Design-Thinking-Workshops und berät Kund:innen bei der ethischen wie rechtskonformen Datenerhebung und -nutzung. Mit Digitaler Ethik und der sozialen Gestaltung der Digitalisierung beschäftigt sie sich außerdem auf politischer Ebene.



David Berger studierte Sportmanagement und Strategic Management mit dem Schwerpunkt Marketing, Innovation Management und Sustainable Development. Er ist zertifizierter Product Owner und hat mehrjährige Erfahrung in der Leitung von Data & Analytics Projekten für global agierende Kunden, wodurch er fundierte Expertise in der Konzeption und Umsetzung agiler Datenprojekte gesammelt hat. Er arbeitet in München als Berater für internationale Projekte im Bereich Business Strategy, Data-Driven Marketing und Business Transformation.



Dr. Matthias Böck promovierte in Bioinformatik und Machine Learning und arbeitet seit 2013 als Data Scientist im Bereich Data Product bei der Münchner Unternehmensberatung FELD M. Dort ist er technischer Leiter für Projekte aus den Bereichen Advanced Analytics und Maschinelles Lernen. Dies beinhaltet u. a. Personalisierung, Natural Language Processing oder Vorhersagen sowie die zugrundeliegenden Datenarchitekturen und -strategien. Er hält Design-Thinking-Workshops und arbeitet mit Universitäten an Forschungsprojekten. Neben diesen Feldern beschäftigt er sich mit dem Thema Data for Good und dessen Einsatz in der Praxis.

Abbildungsverzeichnis

Abb. 1.1	Agile vs. Wasserfall (basierend auf TCW o. J.). Reduced Time-to-Market durch iteratives Vorgehen und kontinuierliche Produktverbesserung führt zu früherer Marktreife und erhöhtem Kund:innennutzen.	7
Abb. 1.2	Beispielhafte Kurve der kumulativen Wertschöpfung bei der agilen im Vergleich zur herkömmlichen Produktentwicklung. Dass die Kurve bei der agilen Produktentwicklung kurzzeitig ins Negative laufen kann, ist dem Umstand geschuldet, dass beim Einholen von Feedback auch Rückschläge passieren können, die dazu führen, dass neu konzipiert werden muss oder einzelne Features ganz verworfen werden.	9
Abb. 1.3	Die Kernelemente von Scrum	16
Abb. 1.4	Kanban-Workflow. Exemplarisches und stark vereinfachtes Kanban-Board. Es kann weitere Spalten geben, aber auch Gruppierungen in der Vertikalen, durch sogenannte Swimlanes. Einzelne Elemente können auch anderen Elementen zu-, über- oder untergeordnet werden. WIP-Limits pro Spalte oder Bereich lassen sich mit den entsprechenden Zahlen direkt auf dem Board angeben	20
Abb. 1.5	Stacey-Matrix mit geeigneten Methoden und Frameworks, je nach Grad der Komplexität.	30
Abb. 2.1	Darstellung einer Stakeholder Map als Stakeholder-Matrix	39
Abb. 2.2	Der Double Diamond als Visualisierung der grundsätzlichen Herangehensweise	49
Abb. 2.3	Impact-Effort-Matrix zur Priorisierung Ihrer Use Cases, Daten und Features. Die Achsen können auch in T-Shirt-Größen (S, M, L, XL) ausgedrückt werden	54
Abb. 2.4	Der Double Diamond für Data Thinking. Auch unser Data-Thinking-Prozess setzt auf die vier Phasen des übergeordneten divergierenden und konvergierenden Problem- und Lösungsraums	60

Abb. 2.5	Verbindung von Design Thinking und agilen Methoden. Design Thinking sorgt für das nutzer:innenzentrierte Fokussieren auf tatsächliche Bedürfnisse. Ein agiles Vorgehen in der Umsetzung sorgt für ein iteratives Bauen von (Daten-)Produkten und somit schnelles Schaffen von Mehrwert. Durch die frühe und regelmäßige Nutzung der Produkte werden Erkenntnisse bezüglich tatsächlicher Bedürfnisse gewonnen, die wiederum in den ersten Phasen des Design Thinkings berücksichtigt werden können	65
Abb. 3.1	Eckdaten der AI-Geschichte	68
Abb. 3.2	Schnittmengen (der Teildisziplinen) von AI und Data Science.	70
Abb. 3.3	Beispielstrukturen zweier Neuronaler Netzwerke für die Erkennung von Handschrift. Ein Neuronales Netz besteht dabei aus einer Eingabeschicht (Input Layer), die z. B. für Bilder jeweils Pixel oder bestimmte Teilausschnitte des Bildes aufnimmt, einem Hidden Layer, in dem die verschiedenen Teile analysiert und Zusammenhänge identifiziert werden, und letztlich einem Output Layer, der das Ergebnis zurückgibt. Hier ist es die Zahl 4, die aus dem handschriftlichen Input erkannt werden sollte. Potenziell ist jeder Knoten eines Layers mit allen Knoten des folgenden Layers verbunden. Die Stärke dieser Verbindungen wird beim Lernen des Netzwerks festgelegt. Deep Neural Networks (rechtes Netz), wie man sie für Deep Learning verwendet, besitzen eine Vielzahl an Hidden Layers und können hierdurch komplexe Muster erlernen	75
Abb. 3.4	Vorhersagebeispiel mit einem ARIMA-Modell (Auto Regressive Integrated Moving Average). Hierbei handelt es sich um einen einfachen, aber häufig verwendeten Ansatz, um Trends und saisonale Effekte fortzuschreiben	78
Abb. 3.5	Confusion-Matrix. Diese Matrix beschreibt das Verhältnis der vorhergesagten Klassen mit der wahren/tatsächlichen Klasse und wie entsprechende Evaluationsmetriken daraus berechnet werden können.	81
Abb. 3.6	Überblick über die wichtigsten Machine-Learning-Grundkonzepte. Die meisten Methoden lassen sich in Supervised, Unsupervised und Reinforcement Learning einteilen. Dazwischen gibt es aber auch Mischformen, wie z. B. Semi-Supervised Learning, bei dem sowohl gelabelte als auch ungelabelte Daten verwendet werden	85
Abb. 3.7	Beispiel eines einfachen Entscheidungsbaums. Hier wird anhand von verschiedenen Features (Vorhersage, Luftfeuchtigkeit und Wind) entschieden, ob es ein guter Tag zum Tennis spielen ist oder nicht. Die Features (in Rechtecken) werden auch als Knoten bezeichnet. Die unteren Endknoten (in Kreisen) werden Blätter genannt und	

	geben jeweils mit „Ja“ oder „Nein“ das Ergebnis an. Der oberste Knoten (hier Vorhersage) gibt das Feature an, das die Daten am besten aufsplittet und wird als Wurzel (Root) bezeichnet	87
Abb. 3.8	Schematische Darstellung des Reinforcement Learnings. Ein Agent (Algorithmus, der verschiedene Aktionen ausführen kann) kann innerhalb eines gegebenen Environments (künstliche oder auch reale Umgebung) eine Aktion ausführen. Diese Aktion führt zu einem Ergebnis, das in eine Belohnung (Reward, positiv oder negativ) übersetzt und an den Agenten zurückgespielt wird. Dieser entscheidet, welche Aktion er als nächstes ausführen möchte, basierend auf dem jetzigen Zustand (State) des Environments und dem bisher Erlernten	91
Abb. 3.9	Beispiele für eine GAN-Anwendung (Detail, Zakharov et al. 2019). Hier wurden aus einem gegebenen Bild verschiedene Gesichtsausdrücke oder Posen erzeugt. Bei den Resultaten handelt es sich nicht nur um statische, sondern tatsächlich bewegte Bilder (kurze Videos), die aus wenigen gegebenen statischen Bildern generiert wurden	93
Abb. 3.10	Netzwerkstruktur eines Autoencoders. Auch hier gibt es einen Input Layer, der allerdings die empfangenen Informationen auf einem Hidden Layer mit weniger Knoten abbilden muss. Die Informationen müssen hierfür komprimiert oder kodiert werden. Deswegen wird der Input Layer auch als Encoder bezeichnet und der Hidden Layer als Bottleneck (Flaschenhals). Der Output Layer muss dann in der Lage sein, die komprimierten Informationen wieder zu dekodieren (Decoder)	94
Abb. 3.11	Iterativ-inkrementeller Prozess in MLOps (basierend auf MLOps o. J.) . . .	96
Abb. 3.12	Reifegraddiagramm der verschiedenen Entwicklungsstufen von Analysen in einem Unternehmen – angefangen bei deskriptiven Reports, die rückwirkend beschreiben, was passiert ist, bis hin zu präskriptiven Modellen, die Vorhersagen erstellen und basierend auf diesen Handlungsempfehlungen geben. Dem jeweiligen Reifegrad entsprechend, steigt auf der x-Achse die Komplexität und damit auch die verbundenen Kosten. Diese steigen in der Realität jedoch nicht linear und sind abhängig von den jeweiligen Anforderungen. Auf der y-Achse erhöht sich der potenzielle Mehrwert. Parallel dazu steigt in der Regel mit der Komplexität der Modelle auch der Anspruch an die Menge und Granularität der Daten, was wiederum zu datenschutzrechtlichen wie technischen Fragestellungen führt. Richtig ist aber auch, dass nicht jedes Unternehmen und jede Abteilung den höchsten Reifegrad erreichen muss und in vielen Fällen bereits die diagnostische Ebene absolut ausreichend sein kann, um die eigenen Ziele zu erreichen	99

Abb. 4.1	Screenshot des Tweets, der die gefährliche Challenge zeigt, die „Alexa“ vorschlägt.	116
Abb. 4.2	Die drei Dimensionen (oder Säulen) der Nachhaltigkeit.	130
Abb. 4.3	Die drei Ebenen der Wirkung	131
Abb. 5.1	Die Datenwertschöpfungskette (Data Value Chain) mit ihren vier Teilen Datenerhebung, Datenverarbeitung, Datenanalyse und Datenmehrwert.	141
Abb. 5.2	Gesamtübersicht der verschiedenen Komponenten einer modernen Dateninfrastruktur (basierend auf Bornstein et al. 2020). Ausgeschlossen sind OLTP-, Log-Analyse- und SaaS-Analyse-Anwendungen	157
Abb. 5.3	Mindmap mit exemplarischer Daten- und Tool-Landschaft	160
Abb. 5.4	Beispiel für eine einfache Datenarchitektur mit wenigen Tools. Bereits dieses rudimentäre Setup reicht für Anwendungsfälle rund um die Datenintegration, Generierung einer SSOT, die Erstellung von Berichten und Dashboards sowie ad-hoc-Analysen innerhalb der Datenbank. Daten, die nicht durch die Connector-Tools abgedeckt werden können, könnten mit Python-Skripten und einem Upload über einen SFTP-Server verarbeitet werden. Die Kosten für diesen Tool-Stack ergeben sich größtenteils aus den Lizenzkosten des Connector-Tools. Python, PostgreSQL und Metabase sind Open-Source-Technologien.	161
Abb. 5.5	Data-to-AI-Reifegradstufen von Unternehmen oder Abteilungen (basierend auf Bratton und Glynn 2020). Diese Übersicht hilft, sich den Status quo zu verdeutlichen und zu überlegen, wie man die nächste Stufe erreichen kann. Genauso hilft sie, neue Projekte zu bewerten und in die richtige Reihenfolge zu bringen. Befinde ich mich gerade noch auf Stufe 2 und versuche, das Reporting sauber aufzusetzen, ist es vermutlich nicht der nächste Schritt, ein komplett automatisiertes AI-System zu bauen. Hierfür fehlen in der Regel die Daten in ausreichender Qualität und Menge oder das grundlegende Verständnis der eigenen Daten.	166
Abb. 5.6	Übersicht über die verschiedenen Bestandteile von Data Governance. Aufbauend auf den Menschen, Prozessen und Technologien unterstützt Data Governance über vier wesentliche Säulen die datengetriebenen Entscheidungen eines Unternehmens. Die vier Säulen sind: Policies und Standards (Wer darf mit welchen Daten welche Entscheidungen treffen), Informations-/Datenqualität (Stewardship), Privacy/Compliance und Security (Assessment der Risiken und wer welchen Zugriff auf die	

	Daten erhalten darf) und Architektur/Integration (Einheitliche Definitionen der Daten, Übersicht der gesamten Datenarchitektur und des Datenflusses).	170
Abb. 5.7	Data Value Loop: Reduzierte Time-to-Data im großen Loop. Durch ein iteratives Vorgehen sollen die einzelnen Schritte der Datenwertschöpfung in kürzeren Abständen und öfters durchlaufen werden, um bereitgestellte Daten und andere Use Cases früh und regelmäßig in Kontakt mit Endnutzer:innen zu bringen	174
Abb. 5.8	Data Value Loop: Reduzierte Time-to-Insight im kleinen Loop. Bei der Analyse von Daten zum Erkenntnisgewinn wird die Zeit zwischen Fragestellung und gewonnener Erkenntnis reduziert, um gewonnene Insights früh in Optimierungen oder weitere vertiefende Analysen einfließen zu lassen.	176
Abb. 5.9	Data Value Loop: Fokus & Priorisierung innerhalb des Loops. Angestrebte Use Cases und bestehende Anforderungen werden anhand des zu erwartenden Mehrwerts priorisiert. Das Team fokussiert sich nur auf die für den Moment relevantesten Use Cases in der nächsten Iteration. Bei der fortlaufenden Arbeit mit Datenanalysen werden bestehende Fragestellungen ebenfalls nach Relevanz priorisiert und es wird immer nur an einer bestimmten Anzahl dieser Fragestellungen gleichzeitig gearbeitet (= WIP-Limit). Bei Bedarf (insb. bei der Entwicklung und Verwaltung unternehmensweiter Analytics-Systeme) können innerhalb der einzelnen Phasen der Datenwertschöpfung zusätzlich WIP-Limits definiert werden, um sicherzustellen, dass es zu keinen Engpässen beim Durchlaufen des Data Value Loops kommt	178
Abb. 5.10	Data Value Loop: Inspect & Adapt. Durch die aktive Arbeit mit Daten (z. B. Analysen und Reportings) wird fortlaufend Feedback über Qualität und Umfang der bestehenden Daten und Prozesse in den Data Value Loop zurückgespielt. Durch die Evaluation des gelieferten Datenmehrwerts einzelner Use Cases und Liefergegenstände am Ende einer Iteration wird Feedback generiert, das in die nächste Iteration einfließen kann	180
Abb. 5.11	Data Value Loop: Transparenz, Autonomie und Selbstorganisation als Multiplikatoren des Datenmehrwerts und Feedback Loops. Durch das Bereitstellen der durch den Data Value Loop generierten Daten und Lösungen an die gesamte Organisation bzw. das aktive und autonome Nutzen der Lösungen durch alle Personen und Teams, wird der Mehrwert in das Unternehmen getragen und somit vervielfacht. Zugleich vermehrt sich das Feedback an den Data Value Loop	182
Abb. 5.12	Der vollständige Data Value Loop.	183

Abb. 7.1	Empathy Map. Die Empathy Map ist eine Möglichkeit, Erkenntnisse über einzelne Nutzer:innen oder auch Gruppen zu sammeln. Sie soll helfen, sich in die Nutzer:innen und ihren jeweiligen Kontext hineinzusetzen. Dies wiederum hilft später bei der Ableitung und Priorisierung von Anforderungen an eine mögliche Lösung sowie ggf. auch bei der Erstellung von Personas für Marketingzwecke . . .	232
Abb. 7.2	Exemplarische Story Map	234
Abb. 7.3	Use-Case-Steckbrief. Der Steckbrief soll kompakt die wichtigsten Informationen zusammenfassen. Wichtig ist ein sprechender Titel sowie eine Beschreibung des Use Cases. Diese Beschreibung kann z. B. in Form einer User Story verfasst sein. Danach müssen die jeweiligen Zeilen gemeinsam befüllt werden, soweit das im Rahmen des Workshops möglich ist. Das Ziel beschreibt, worauf der Use Case einzahlt und die Erfolgsmessung hält fest, wie dies am Ende quantifiziert werden kann. Daneben sollten die wichtigsten Stakeholder gelistet werden sowie die Datenquellen, die voraussichtlich benötigt werden. Die Zeilen in Grau sind unter Umständen gesondert mit der IT oder auch einem Dienstleister zu klären und können nachgetragen werden. Sollte es wichtige Hürden geben, ist es gut, diese für alle sichtbar zu machen, um sie bei späteren Entscheidungen zu berücksichtigen. Nicht ausgelassen werden darf der Owner, also die Person, die verantwortlich für die Umsetzung des Use Cases ist.	239
Abb 7.4	Now-Next-Later-Template. Diese Vorlage wird verwendet, um Use Cases gemeinsam in eine zeitliche Reihenfolge zu bringen.	240
Abb 7.5	Steampunk-Teddybären. Mit DALL-E 2 generiertes Bild basierend auf dem Text: „Teddy Bears mix sparkling chemicals as mad scientists in a steampunk style“ (OpenAI 2022)	249

Tabellenverzeichnis

Tab. 2.1	Vergleich zwischen Prototyp und Minimum Viable Product.	50
Tab. 3.1	Methoden des Supervised Learnings	86
Tab. 3.2	Übersicht über die Methoden des Unsupervised Learnings	89
Tab. 3.3	Typische Anwendungsfelder von RNN, CNN und GAN	92
Tab. 6.1	Exemplarischer Auszug aus einem Tagging-Plan für das Tracking von Produktinformationen	196
Tab. 7.1	Steckbrief der für unser Projekt relevanten Unternehmen.	230
Tab. 7.2	Exemplarische Agenda eines Kickoff-Workshops.	236



Zusammenfassung

Nicht nur die großen Buzzwords Big Data und AI haben sich in ihrer inflationären Verwendung in den letzten Jahren teilweise verselbstständigt. Auch *Agile* entwickelt im Bereich des Produkt- und Projektmanagements zunehmend ein Eigenleben. In diesem Kapitel erläutern wir, was *Agile* eigentlich ist, was es nicht ist sowie wann und warum sich ein agiles Vorgehen auch für Ihre künftigen Analytics- und AI-Projekte eignet. Ein grundlegendes Vorwissen um agile Prozesse erleichtert Ihnen das Verständnis dieses Kapitels, ist aber nicht zwingend notwendig. Wenn Sie noch nie von agilen Methoden und Denkansätzen gehört haben, empfehlen wir aber eine vertiefende Beschäftigung damit.

1.1 Agil ist überall

Betrachtet man anhand von Google Trends das Suchaufkommen für Begriffe wie „Scrum“ oder „*Agile*“, erkennt man eine kontinuierliche und noch immer anhaltende Steigerung seit den frühen 2000er-Jahren. Für das Wort „Agilität“ im Managementkontext ist das Interesse insbesondere seit 2017 im Vergleich zu den Jahren davor nahezu sprunghaft gestiegen.

Studien und Befragungen bestätigen die weitreichende Verbreitung von *Agile* bzw. agilen Methoden. So verwenden laut einer 2020 erhobenen „Agile Adoption“-Statistik (Djurovic 2020) bereits 71 % der befragten Unternehmen agile Methoden. Auch wenn sich innerhalb dieser Unternehmen die Nutzung noch unterscheidet – manche haben agile Methoden und Prozesse dauerhaft im Einsatz, andere nur sporadisch –, so zeigt es doch, dass

in immer mehr Unternehmen eine agile Adaption stattfindet und das Thema zum prozesualen Breitensport wird.

Dieser Trend wurde auch von einer global angelegten KPMG-Studie aus dem Jahr 2019 (KPMG Advisory N.V. 2019) bestätigt. Die Befragung zeigt, dass 70 % der Unternehmen Ambitionen hatten, innerhalb der drei darauffolgenden Jahre ihre unternehmensweite „Agile Transformation“ voranzutreiben. Der Fokus sollte dabei nicht nur auf IT und Software-Entwicklung liegen, sondern sich auf alle Unternehmensbereiche erstrecken.

Der jährlich erscheinende „State of Agile Report“ (Digital.ai 2021, S. 13) beschäftigt sich unter anderem mit den agilen Methoden, die im Bereich der Software-Entwicklung genutzt werden. Dabei wird ersichtlich, dass Scrum (siehe Abschn. 1.4) mit 66 % die am meisten genutzte Methode ist. 15 % der Befragten haben zudem eine Abwandlung von Scrum bzw. hybride Methoden (z. B. ScrumBan oder Scrum/XP) im Einsatz. 6 % der Umfrageteilnehmer:innen nannten Kanban (siehe Abschn. 1.4.4) als deren vorrangig verwendete Methode.

Die genannten Entwicklungen erlauben vor allem drei zentrale Einsichten:

1. *Agile* ist aus dem Bereich IT und Software-Entwicklung nicht mehr wegzudenken.
2. *Agile* ist auch im allgemeinen Management von Unternehmen, Projekten und Prozessen, die über die IT und Software-Entwicklung hinausgehen, angekommen.
3. Scrum liegt als Vorgehensmodell einem Großteil der umgesetzten agilen Prozesse zugrunde, wenn auch teilweise in abgewandelter bzw. hybrider Form.

Bevor wir uns jedoch im Detail damit beschäftigen, *Agile* für Bereiche außerhalb der Software-Entwicklung – in unserem Fall insbesondere für den Bereich Data & Analytics – anzuwenden, wollen wir erst darlegen, was *Agile* eigentlich ist.

1.2 Wie alles begann: Eine kurze Geschichte von *Agile*

Am Anfang steht die Produktentwicklung und vor allem ihre Veränderung ab etwa Mitte des 20. Jahrhunderts. Produkte waren bis dahin vor allem gefertigte Waren, die zu einem bestimmten Zeitpunkt und nach festgelegtem Ablauf entworfen, geplant, zusammengebaut, verpackt und verkauft wurden. Als Konrad Ernst Otto Zuse 1941 den ersten funktionsfähigen, software-gesteuerten und frei programmierbaren Computer „Z3“ entwickelte, begann die Software-Entwicklung und plötzlich fielen nach und nach – ja, es hat noch ein bisschen gedauert, bis auch Lochstreifen und Lochkarten nicht mehr hergestellt werden mussten – die handwerklichen und physischen Produktionsschritte des eigentlichen Produkts weg. Software wird zwar auch heute noch vereinzelt auf physischen Datenträgern verkauft, aber spätestens seit Einzug des Internets sind Produktverbesserungen via Download und Update der Normalfall.

Was hatte sich also geändert? Ein Produkt war nun nicht mehr in einem finalen, unänderlichen Zustand, sondern konnte angepasst und verbessert werden. Natürlich wurden

auch herkömmliche Produkte wie Staubsauger immer wieder verbessert oder neue Versionen entwickelt, doch mit der Programmierung war es möglich, kleinere und größere Anpassungen unmittelbar vorzunehmen, auszuprobieren und wieder zurückzunehmen. Damit wurden Konzepte wie „inkrementell“ und „iterativ“ in der Software-Entwicklung die logische Konsequenz: Das „inkrementelle Vorgehensmodell“ nahm bereits Mitte der 1950er-Jahre entscheidende Elemente des Agilen vorweg, auch wenn damals noch niemand von Agilität sprach (Larman und Basili 2003). Neu war vor allem das Vorgehen in kleinen oder kleinsten Schritten, im Gegensatz zum vorher vorherrschenden Ansatz der „Big Bang Integration“ (= alles wird am Ende auf einmal veröffentlicht). Das endgültige Resultat musste neuerdings also noch nicht am Anfang feststehen, sondern wurde nach und nach entwickelt.

Inkrementell vs. Iterativ

Im Alltag werden die Begriffe „inkrementell“ und „iterativ“ fälschlicherweise oft synonym verwendet. Damit Ihnen das nicht passiert, hier der Unterschied:

- Inkrementell bedeutet, dass verschiedene Teilbereiche eines Projekts zu unterschiedlichen Zeitpunkten fertiggestellt sein sollen. Unmittelbar nach Fertigstellung eines Teilbereichs wird dieser in das Gesamtsystem integriert.
- Iterativ bedeutet, dass (teil-)integrierte Bereiche weiterhin überarbeitet, getestet und verbessert werden. Wichtig ist, dass die Erkenntnisse und Erfahrungen aus vorherigen Schritten in die nächsten Zyklen übernommen werden können.

Inkrementelles und iteratives Vorgehen ergänzen sich gut, bedingen sich aber nicht. Wenn inkrementell gearbeitet wird, bedeutet es nicht, dass zwangsläufig eine Revision und Überarbeitung stattfinden muss. Ebenso wenig muss ein Produkt in Teilbereichen fertiggestellt werden, um im Nachhinein ausgiebig getestet und verbessert zu werden. Eine Trennung der beiden wäre jedoch eher künstlich und ist nicht die Norm. Den agilen Prozessen liegt zugrunde, dass das Lernen aus gelungenen wie gescheiterten Teilprojekten die wichtigste Voraussetzung ist, um ein gutes Produkt besser zu entwickeln.

In den 1970er-Jahren kamen neue Vorgehensweisen wie das „evolutionäre Projektmanagement“ und die „adaptive Software-Entwicklung“ hinzu. Ersteres ist Vorläufer der „Selbstorganisation“, bei dem die Rahmenbedingungen für Projekte (definierte Ziele, Strukturen, Arbeitsabläufe, Verantwortlichkeiten) nicht vom Management oder einer höheren Ebene vorgegeben, sondern in den Teams selbst entwickelt werden sollten. Das „*Adaptive Software Development (ASD)*“, entwickelt von Jim Highsmith, setzte den Fokus noch einmal mehr auf die Adaption, die als Normalzustand in immer komplexer werden-

den Umfeldern angesehen werden sollte. Wichtigstes Element ist dabei eine regelmäßige Überprüfung der neuesten Entwicklungsfortschritte, die auf eine Reihe von vordefinierten Phasen („Spekulieren“, „Zusammenarbeiten“, „Lernen“) folgt. Wer sich hier jetzt bereits an „Sprints“ erinnert fühlt, weiß erstens offensichtlich schon einiges über *Agile* und Scrum und hat zweitens auch Recht.

Von einer agilen Software-Entwicklung im heutigen Sinn kann man etwa ab den 1990er-Jahren sprechen. Ihren Durchbruch erlangte sie vor allem durch das 1999 von Kent Beck verfasste Buch „Extreme Programming“ (siehe Abschn. 1.4.4) zur gleichnamigen, damals vorherrschenden agilen Methode. Extreme Programming (XP) wurde als gängigste agile Methode spätestens Mitte der 2000er-Jahre von Scrum abgelöst.

Bis 2001 nannte man die Vorläufer der agilen Methoden jedoch noch nicht „agil“, sondern „lightweight“, also leichtgewichtig. Der Begriff „Agile“ wurde 2001 vom Physiker und Software-Entwickler Mike Beedle bei einem Treffen der damals einschlägigen Community in Utah ausgewählt. Bei dieser Gelegenheit entstand auch das noch heute immer wieder zitierte und als agiler Standard herangezogene „Agile Manifesto“, in welchem die vier agilen Grundwerte niedergeschrieben sind (Beck et al. 2001):

- „Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan.“

Diese vier Werte zeigen, dass die Kernidee der agilen Bewegung von den heute prädominanten Methoden unabhängig ist und sich auf wenige grundlegende Prinzipien stützt: Fokus auf Zusammenarbeit von Personen, Funktionalität der Software/des Produkts, kurze Entwicklungszyklen, Kund:innenzentrierung und erhöhte Flexibilität bzw. Adaptivität. Wir stellen Ihnen diese Grundprinzipien im nächsten Kapitel als Erfolgsfaktoren für Ihre künftigen Projekte genauer vor.

1.3 Agile Prinzipien als Erfolgsfaktoren

Am Anfang jedes agilen Prozesses oder Projekts sollten die grundlegenden Fragen nach dem „Was?“ und dem „Wie?“ beantwortet werden. Das „Was“ ist die Vision, die Strategie, die Anforderung, der Use Case. Das „Wie“ ist der Prozess.

- Was ist es, das unsere Kunden:innen bzw. Endnutzer:innen beschäftigt? Was stiftet ihnen Nutzen? Was brauchen sie, um ihre Probleme zu lösen bzw. ihre Bedürfnisse zu befriedigen? Was müssen wir entwickeln, umsetzen oder liefern, um genau diese Bedürfnisse zu treffen? Und was davon idealerweise als erstes und in welchem Umfang?
- Wie setzen wir das um? Wie kommen wir vom Verständnis der Bedürfnisse und Anforderungen zu einem ersten Produkt? Wie schaffen wir es, schnellstmöglich aber auch

regelmäßig und ständig zusätzlichen Nutzen zu liefern? Wie stellen wir sicher, dass wir nicht nur das Produkt oder die Qualität unserer Liefergegenstände, sondern auch unsere Arbeitsabläufe weiterentwickeln und optimieren?

Die folgenden agilen Erfolgsfaktoren, die wir aus den Werten des Agilen Manifests abgeleitet haben, helfen Ihnen dabei, diese Fragen zu beantworten und das Projekt gemäß der agilen Prinzipien umzusetzen. Denn „agil sein“ heißt nicht, spontan zu sein, Planungen zu meiden oder dass Teams einfach rumprobieren. Vielmehr stecken dahinter klare Annahmen und Sinneshaltungen. *Agile* gibt Struktur und Flexibilität, braucht dafür aber auch entsprechend Disziplin und neue Kompetenzen.

1.3.1 Reduzierte Time-to-Market & Iteration

Ein zentrales Merkmal von agiler Arbeit ist, dass sie die Zeit zwischen Anforderung und Ergebnis durch iteratives Vorgehen so kurz wie möglich hält. Iterativ bedeutet konkret, dass Produktbestandteile, Aufgaben oder auch nur Teile von Aufgaben in kurzen, sich wiederholenden Zeitabschnitten bewertet, geplant und verbessert werden. Dadurch wird die Wahrscheinlichkeit erhöht, dass Sie mit Ihrem Produkt die Anforderungen und den erhofften Nutzen für Endnutzer:innen besser und schneller treffen. Feedback und Anpassungswünsche können ebenfalls früher und effizienter eingearbeitet werden. Iteratives Vorgehen reduziert das Risiko, dass Sie am tatsächlichen Bedarf vorbeientwickeln, steigert Ihre Flexibilität und sorgt dafür, dass Sie zu jeder Zeit am aktuell Wichtigsten und Wertschöpfendsten arbeiten. Sven Günther (2020, S. 61) beschreibt den dadurch entstehenden Nutzen wie folgt:

„Um flexibel zu werden, dürfen wir uns nicht zu früh festlegen. Daher betrachten wir alle Aufgaben, die noch nicht angefangen wurden, als optional. Optional bedeutet, dass noch nicht entschieden ist, wann die Aufgabe fertig wird oder ob sie überhaupt erledigt wird. Diese Entscheidung wollen wir zum spätest möglichen Zeitpunkt treffen, um jederzeit auf neue, dringendere oder wichtigere Anfragen reagieren zu können.“

Betrachtet man die Motive, weshalb Unternehmen ihre Prozesse und Arbeitsweisen auf *Agile* umgestellt haben, sieht man, dass genau die Fähigkeit, auf sich ändernde Prioritäten gut und schnell reagieren zu können, mittlerweile einer der Hauptgründe ist: So haben im „State of Agile Report“ (Digital.ai 2021, S. 8) 64 % der befragten Unternehmen im Bereich Software-Entwicklung angegeben, dass „Enhance ability to manage changing priorities“ einer der wichtigsten Gründe der agilen Adaption waren. Ebenfalls 64 % gaben „Accelerate software delivery“ als einen Hauptgrund an, also das schnelle und iterative Liefern von Ergebnissen.

Bei der besonders in der Software-Entwicklung weit verbreiteten Scrum-Methode wird Iteration dadurch erreicht, dass die zu bearbeitenden Aufgaben für fixe Zeitperioden – sogenannte *Sprints* – definiert und geplant werden. Ziel ist es dabei, alle geplanten Aufgaben

innerhalb eines Sprints zu bearbeiten und sie am Ende des Sprints zu bewerten, um dann für den nächsten Sprint, sprich: die nächste Iteration, den Prozess wieder von vorne zu starten. Definieren, planen, bearbeiten, bewerten. Geplant werden dabei immer nur die Aufgaben, die aktuell die höchste Wichtigkeit haben.

Aber auch außerhalb von Scrum kann und sollte iteratives Vorgehen angestrebt werden. Alleine durch ein regelmäßiges (wöchentliches, zweiwöchentliches, monatliches) Bewerten aller abgeschlossenen, noch laufenden sowie zukünftig geplanten Aufgaben wird sichergestellt, dass ein ständiger Abgleich mit den sich möglicherweise ändernden Anforderungen stattfindet. Dadurch können Inhalte von bereits gestarteten Aufgaben angepasst sowie neue Aufgaben definiert werden. Das wiederum erhöht die Effektivität der eingesetzten Ressourcen.

Das iterative Mantra sollte lauten

Wir tun das, wovon wir aktuell ausgehen, dass es den besten Beitrag zum Erreichen definierter Ziele liefert. Merken wir, dass eine Kursänderung notwendig ist, können wir schneller reagieren. Ist eine Aufgabe oder ein Liefergegenstand obsolet geworden, erkennen wir das früh und können damit aufhören. Ressourcen werden dadurch nicht verschwendet und können wieder produktiv eingesetzt werden.

Dass dieses Denken bei vielen Unternehmen bereits ins Bewusstsein gerückt ist, zeigt erneut der „State of Agile Report“ (Digital.ai 2021, S. 13). Zur Verwendung von agilen Praktiken gaben 83 % der befragten Unternehmen an, dass sie Planung in Sprints bzw. Iterationen betreiben.

In Abb. 1.1 sehen Sie den Vergleich zwischen der herkömmlichen Wasserfall-Methode und dem agilen Ansatz. Die reduzierte Time-to-Market lässt sich vor allem dadurch erklären, dass die Erprobungs- und Lernphase bereits in die Sprints eingeflossen ist und nicht erst nach dem Release, also der Markteinführung, wie bei der Wasserfall-Methode zu sehen, stattfindet. Ein agiler Ausdruck, den man in diesem Zusammenhang findet, ist JEDUF (= Just Enough Design Upfront). Das bedeutet, dass Design und Entwicklung Hand in Hand gehen und das Konzept (oder Design) erstmal nur so weit reicht, dass es die grundlegendsten Aspekte abdeckt, sodass diese früh getestet und angepasst werden können (Lisefski 2019).

1.3.2 Fokus

Ein weiterer entscheidender Bestandteil agilen Vorgehens ist das gezielte Fokussieren auf wenige (Teil-)Aufgaben mit dem Ziel, diese schnellstmöglich und effizient zu beenden, bevor wieder neue Aufgaben begonnen werden. Dies bedeutet, dass man versucht, die Anzahl der sich in Bearbeitung befindlichen Aufgaben so gering wie möglich zu halten und in eine Iteration nur aufzunehmen, was notwendig ist und im Rahmen einer Iteration

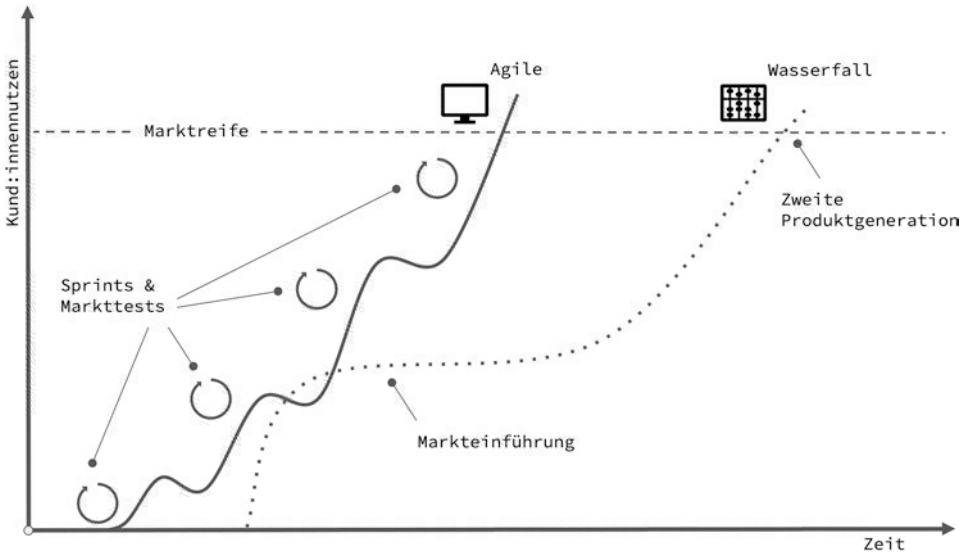


Abb. 1.1 Agile vs. Wasserfall (basierend auf TCW o. J.). Reduced Time-to-Market durch iteratives Vorgehen und kontinuierliche Produktverbesserung führt zu früherer Marktreife und erhöhtem Kund:innennutzen

realistisch geleistet werden kann. So sind alle verfügbaren Ressourcen zwar optimal ausgelastet, die Teammitglieder müssen aber nicht an zu vielen Dingen parallel arbeiten. Sind zu viele Aufgaben gleichzeitig im Workflow, führt das dazu, dass sich die Zeit zwischen Start und Beendigung der einzelnen Aufgabe verlängert.

Das ergänzt das Prinzip des iterativen Vorgehens: Wenn wir eine Aufgabe immer zum spätmöglichen Zeitpunkt starten möchten, um das Risiko zu reduzieren und die Flexibilität zu erhöhen, sollten wir diese auch erst dann beginnen, wenn genug Ressourcen dafür vorhanden sind, um sie in der geringstmöglichen Zeit abzuschließen. Je schneller wir die Aufgabe abschließen, desto früher können wir sie bewerten und wieder Ressourcen für weitere Aufgaben frei machen. Dementsprechend ist „Stop starting and start finishing“ ein weit verbreitetes agiles Credo, das beispielsweise in Scrum und Kanban auf unterschiedliche Art und Weise umgesetzt werden kann.

Wie bereits erwähnt, bedient sich Scrum *Sprints* – fixe, regelmäßig aufeinander folgende Zeitperioden, in denen Aufgaben geplant und bearbeitet werden. Das führt nicht nur zu einem iterativen Vorgehen, sondern schafft zugleich Fokus. Pro Sprint entscheidet man sich für eine Anzahl an Aufgaben, die mit den vorhandenen Ressourcen innerhalb des Sprint-Zeitraums erledigbar sind und die aktuell höchste Priorität haben. Innerhalb des Sprints wird ausschließlich an diesen Aufgaben gearbeitet. Alles, was an neuen Aufgaben und Anforderungen während dieser Zeit aufkommt, wird frühestens für den nächsten Sprint in Betracht gezogen. Ziel ist es, alle Aufgaben eines Sprints auch bis zum Sprint-Ende vollständig abzuschließen. Dadurch stehen im nächsten Sprint alle Ressourcen wieder vollständig für neue Aufgaben bereit und es kann ein neuer Fokus gelegt werden.

Kanban wiederum arbeitet mit sogenannten *WIP*(= Work-In-Progress)-*Limits*. *WIP-Limits* definieren die maximale Anzahl an Aufgaben, die zu einem Zeitpunkt in Bearbeitung sein dürfen. Diese *WIP-Limits* können sowohl pro Arbeitsschritt, Person, Art der Aufgabe oder auch für das gesamte Kanban-System definiert werden. Das Ziel liegt auf der Hand: Man hat nur die Anzahl an Aufgaben im System, die mit den vorhandenen Ressourcen effektiv bearbeitbar sind und denen die aktuell höchste Priorität zugewiesen wird. Neue Aufgaben werden erst gestartet, wenn bestehende beendet und somit Ressourcen frei geworden sind.

Wer es gerne etwas akademischer hat, kann sich bei *WIP-Limits* auf *Little's Law* beziehen. Dieses beschreibt die Beziehung zwischen folgenden drei Variablen (Inthapichai 2020):

- **Work-In-Progress (WIP) Items:** Anzahl der Aufgaben, die sich gleichzeitig in Bearbeitung befinden
- **Throughput:** Anzahl der Aufgaben, die pro Zeiteinheit abgeschlossen werden
- **Lead Time:** Zeit, die eine Aufgabe im Schnitt von Start bis Beendigung braucht

Little's Law bringt diese wie folgt in ein Verhältnis: **WIP = Throughput x Lead Time**

Auch wenn *Little's Law* bei der praktischen Verwendung in Arbeitsprozessen an seine Grenzen stößt und der Komplexität von unterschiedlichen Anforderungen, Größen von Aufgaben sowie Abhängigkeiten innerhalb von Arbeitsabläufen nicht zur Gänze Rechnung tragen kann, zeigt es dennoch, dass die Optimierung der Anzahl sich in Bearbeitung befindlicher Aufgaben durch Fokussierung zu einer kürzeren Durchlaufzeit pro Aufgabe führt.

Ein weiteres Modell, das vor allem die psychologische Komponente von Work-In-Progress beleuchtet, ist der Zeigarnik-Effekt. Dieser besagt, „dass Menschen sich an nicht beendete Aufgaben eher erinnern als an abgeschlossene Aufgaben“ (Gekeler 2019). So banal die Erkenntnis scheint, so relevant ist sie für die Art und Weise, wie wir Arbeitsabläufe gestalten. Gelingt es uns nicht, genügend Fokus – sprich: die Reduktion der Anzahl an Aufgaben pro Person und Arbeitsschritt – zu schaffen, erhöhen wir die Anzahl nicht abgeschlossener Aufgaben. Dadurch können die Bearbeitenden mental nicht mit den noch offenen Aufgaben abschließen, was wiederum zu kognitiver Überlastung und verringerter Fokussierung führen kann. Dies wirkt sich nicht nur negativ auf das Stress-Level aus, sondern kann durch eine aus Überforderung resultierende kurzfristige Inaktivität auch zu ineffizienterem Arbeiten führen (Kodden 2020).

1.3.3 Priorisierung

Fokus bringt nur dann entscheidende Vorteile, wenn man sich auf die richtigen Aufgaben fokussiert. Dafür braucht es Priorisierung. Unter Priorisierung versteht man das Vorgehen, das zu einer Entscheidung für die zum aktuellen Zeitpunkt wichtigsten Aufgaben bzw. angestrebten Liefergegenstände führt.

Um zu dieser Entscheidung zu kommen, braucht es einerseits eine vollständige Übersicht aller aktuell möglichen Optionen, an denen gearbeitet werden könnte. Andererseits Bewertungskriterien, die dabei helfen, mit hoher Wahrscheinlichkeit jene Optionen zu wählen, die den aktuell höchsten Mehrwert in Relation zum Aufwand stiften.