



SERVIEW

CONSULTING.
TRAINING.
DIFFERENT!

SCRUM

MASTER & PRODUCT OWNER

ALLES, WAS MAN WISSEN MUSS

ISBN 978-3-7583-8166-9

© Copyright Michael Kresse & Kathrin Eggerling 2023

1. Auflage, November 2023

IMPRESSUM

Herstellung und Verlag:

BoD – [Books on Demand](#), Norderstedt

Gartenstr. 23, 61352 Bad Homburg v.d. H., Deutschland

Tel.: +49 (0) 6172 / 177 44-0

info@serview.de, www.serview.de

Konzeption und Gestaltung:

UnitedCreation GmbH

Zehntwiesenstr. 35b, 76275 Ettlingen, Deutschland

Tel.: +49 (0) 7243 / 20 711-0

info@unitedcreation.de, www.unitedcreation.de

Autoren:

Kathrin Eggerling



Michael Kresse



Co-Autoren:

Michael Baumann

Tino Besser

Katja Bilsing

Erik Kurths

Thomas Marenich

Alexander Quast

Johannes Uske



IN- HALTS- VER- ZEICH- NIS

10 Vorwort

12 Über SERVIEW

TEIL I — DAS SCRUM-FRAMEWORK

16 Überblick: Scrum und die agile Idee

24 Das Agile Manifest

28 Was ist Scrum?

31 Säulen und Werte

36 Rollen: Das Scrum Team

38 Product Owner

41 Scrum Master

43 Developer

45 Das Scrum Team im Umfeld

48 Events: Der Herzschlag von Scrum

50 Warum Timeboxing?

50 Eventüberblick

52 Sprint Planning

54 Daily Scrum

55 Sprint Review

56 Sprint Retrospective

58 Refinement

64 Artefakte & Commitments: Aus Ideen werden Features

- 66 Commitment: Product Goal
- 67 Artefakt: Product Backlog
- 71 Best Practice: Definition of Ready
- 72 Commitment: Sprint Goal
- 73 Artefakt: Sprint Backlog
- 75 Commitment: Definition of Done
- 77 Artefakt: Product Increment

80 Scrum in Action: Der Scrum-Flow im Überblick

86 Best Practices: Reporting und Visualisierung

- 89 Velocity und Capacity
- 92 *Burn-Charts*
- 93 *Burn-Down-Chart*
- 95 *Burn-Up-Chart*
- 98 Product-Burn-Up
- 102 Nachhaltige Geschwindigkeit und Warnsignale
- 103 Kanban Board
- 108 Impediment Backlog
- 110 Scrum Board

TEIL II — METHODEN UND TECHNIKEN FÜR PRODUCT OWNER

114 Das große Ziel: Product Goal

- 116 Entwicklung des Product Goals
- 120 Product Goal Template
- 122 Product Vision Board

125 Was tun beim Nicht-Erreichen des Sprint Goals?

130 Der Weg dorthin: Product Roadmap und Releases

131 Product Roadmap

134 User Storys

140 Von der User Story zum Release

141 User Story Map

144 Metriken für Product Owner

148 Überblick: Product Backlog Management

152 Was wirklich zählt: Requirements Engineering

155 Phasen des Refinements

158 Priorisieren

159 *Kano-Modell*

162 *MoSCoW-Prinzip*

166 Detaillieren

167 *SPIDeR-Modell*

170 Schätzen

173 *Story Points als Skala*

176 *Schätzmethode: Planning Poker*

179 *Schätzmethode: Magic Estimation*

03

TEIL III — METHODEN UND TECHNIKEN FÜR SCRUM MASTER

184 Aspekte eines Scrum Masters

188 Führung: Den richtigen Weg wählen

- 189 Servant Leader sein
- 192 Situativ führen
- 195 Führen mit dem DISG-Modell

202 Facilitation: Hindernisse ausräumen

- 203 Als Facilitator agieren
- 205 Als Manager lenken
- 208 Als Troubleshooter Hindernisse überwinden

212 Teamentwicklung: Gemeinsam wachsen

- 213 Phasen der Teamentwicklung
- 219 Als Coach unterstützen
- 222 Als Mentor den Weg weisen

224 Organisationsentwicklung: Das große Ganze sehen

- 225 Als Lehrer Wissen vermitteln
- 227 Als Change Agent Wandel gestalten

232 Zeitplanung

236 Retrospective: Der Blick zurück

- 241 Methoden
- 241 *Seestern-Retrospective*
- 243 *Zeitleisten-Retrospective*
- 246 *Glad, Sad, Mad*

248 Best Practice zum Sprint-Set-up

TEIL IV — SCRUM IM KONTEXT

254 Relevante Frameworks & Best Practices

- 255 DevOps
- 257 PRINCE2® Agile
- 258 Kanban
- 260 Design Thinking
- 261 Lean Startup
- 263 OKR

266 Scrum skalieren

- 268 Scrum of Scrums
- 268 Nexus
- 269 LeSS
- 270 SAFe®

272 Wichtige Organisationen

276 Prüfungsvorbereitung

280 Glossar



VOR- WORT



Willkommen in einer neuen Welt! Einer Welt, in der Kommunikation mehr zählt als feste Prozesse. Wo Teams kontinuierlich nach Verbesserung streben. Wo innovative Produkte in erstaunlich kurzer Zeit die Marktreife erreichen – und wo Menschen durch effektive Zusammenarbeit über sich hinauswachsen. **Willkommen in der Welt von Scrum!**

Dieses Buch soll Ihr Wegbegleiter durch den agilen Kosmos sein. Es enthält alles, was Sie wissen müssen, um Ihre Reise als Scrum Master, Product Owner oder Developer zu beginnen. Für Ihren Erfolg in der Praxis haben wir einen gewaltigen Wissensschatz zusammengetragen: vom Basiswissen über die einzelnen Elemente von Scrum und ihr effektives Zusammenspiel bis hin zu Hintergrundinformationen, Best Practices sowie zahlreichen Tipps und einprägsamen Beispielen.

Es war mein persönlicher Herzenswunsch, einen umfangreichen Leitfaden für die Welt von Scrum zu verfassen. Dieses Buch soll Ihnen kompetent und unterhaltsam den Weg weisen, während Sie Scrum für sich entdecken. Gleichzeitig dient es als Nachschlagewerk, wenn Sie Inspiration suchen oder ihre Kenntnisse auffrischen möchten. Und natürlich können Sie es ebenso nutzen, um sich auf eine offizielle Scrum-Zertifizierung vorzubereiten.

Wie auch immer Sie das Wissen in diesem Buch für sich verwenden, an welchem Punkt Ihrer Scrum-Reise Sie sich auch befinden: Ich wünsche Ihnen viel Freude und Erfolg auf Ihrem Weg – und dass Sie ganz im Sinne des agilen Gedankens nie aufhören zu lernen.

Ihr Michael Kresse
Geschäftsführer der SERVIEW GmbH



ÜBER SERVIEW



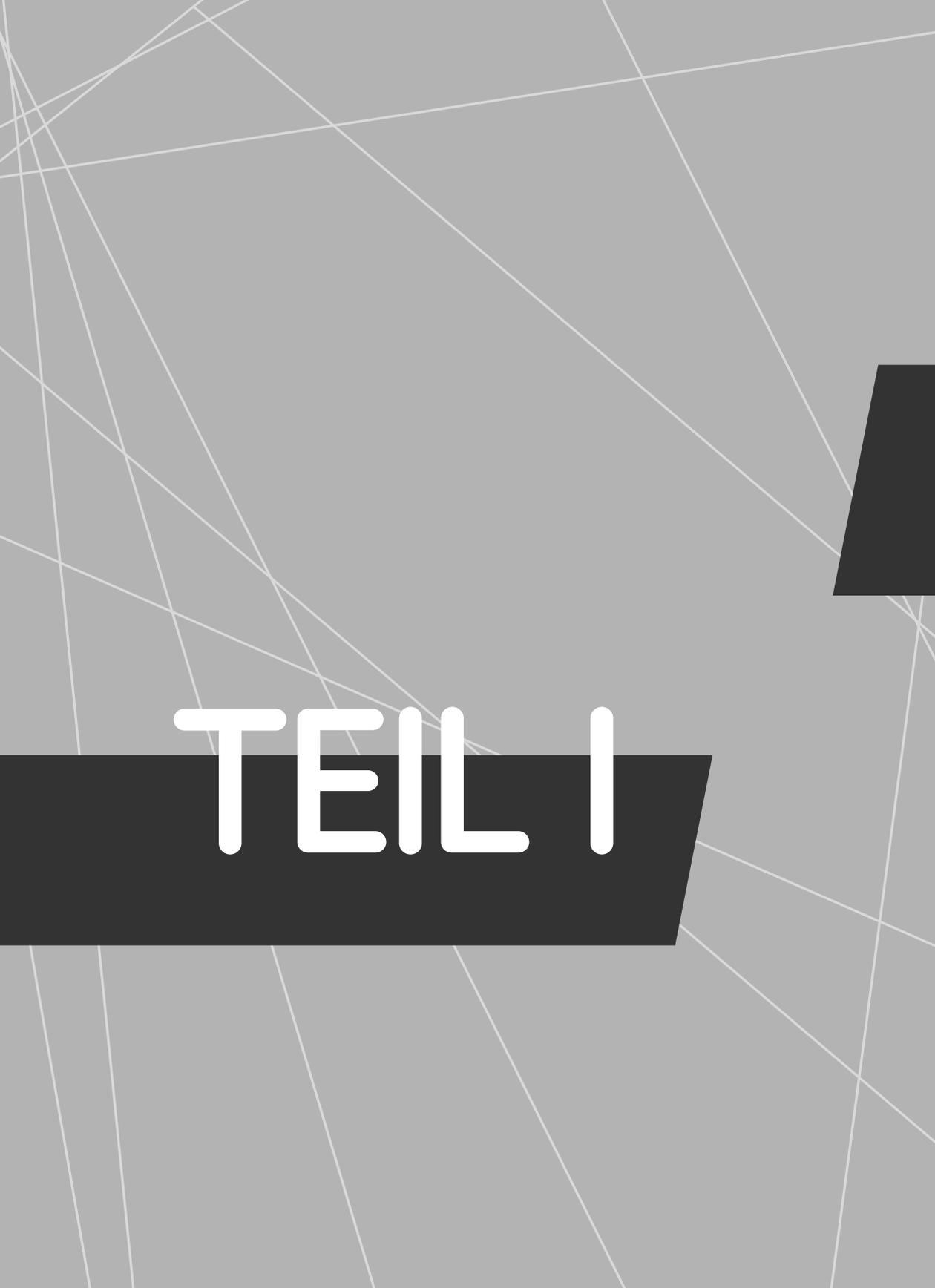
SERVIEW – das ist gelebte Leidenschaft, außergewöhnliche Kompetenz und furchtloses Streben nach immer besserer Leistung. Als einzigartige, voll auf etablierte Managementmethoden fokussierte Unternehmung kämpfen wir für herausragende IT-Organisationen, deren Geschäftsbereiche wir zu Höchstleistungen befähigen.

Mit zielgerichteten Impulsen und kühnen Ideen beraten und trainieren wir seit 20 Jahren innovationshungrige Unternehmen beim Einsatz von Service Management, agilen Arbeitsweisen und effektivem Projektmanagement. Jährlich erleben rund 15.000 Teilnehmerinnen und Teilnehmer unsere bahnbrechenden Schulungen, deren Lerneffekt inspiriert und beflügelt. Ergänzt wird das Angebot um energiegeladenes Consulting und dynamisches Coaching on the Job für die zahlreichen Rollen der vermittelten Best Practice Frameworks wie ITIL®, PRINCE2® oder Scrum. Dadurch vermitteln wir nicht nur Wissen, sondern erwecken es direkt in Kundenorganisationen zum Leben, um deren Herausforderungen zu lösen – auf der persönlichen Coaching-Ebene sowie unternehmensweit in der Beratung. So verankern wir genau die Methoden und Techniken in einer Organisation, die den individuell größten Mehrwert erzielen.

Seit unserer Gründung 2002 erfinden wir uns immer wieder neu. Angetrieben von unbändiger Innovationskraft entwickeln wir unsere Schulungs-, Beratungs- und Eventformate konsequent weiter. Persönliche Weiterentwicklung als Erlebnis, offener Austausch durch alle Branchen und mitreißende Veranstaltungen, die man nie vergisst: So verstehen wir unseren Anspruch an eine Businesswelt, in der alle nach den für ihre Organisation besten Lösungen streben. Lassen Sie uns gemeinsam nach den Sternen greifen – angefangen mit Ihrem Pfad zum Scrum-Profi.



www.serview.de



TEIL I

DAS SCRUM- FRAME- WORK

ÜBERBLICK:

SCRUM

UND DIE
AGILE
IDEE

Die agile Arbeitsweise ist in aller Munde. Mit ihrer Hilfe lassen sich Produkte und Services kundenorientierter, ressourcenschonender und fokussierter entwickeln. Doch warum ist das eigentlich so? Was macht die agile Arbeitswelt grundlegend anders? Um dieser Frage auf den Grund zu gehen, stellen wir zwei Vorgehensweisen gegenüber: das klassische Wasserfallmodell auf der einen Seite, den agilen Ansatz auf der anderen.



Da sich in der Praxis viele englische Bezeichnungen durchgesetzt haben, finden Sie in diesem Buch hinter den deutschen Ausdrücken zahlreiche englische Fachbegriffe in Klammern und kursiv. Verinnerlichen Sie am besten beide Varianten, um sowohl für die Prüfung als auch für die praktische Arbeit mit Scrum gerüstet zu sein. Für eine bessere Lesbarkeit verwenden wir durchgängig das generische Maskulinum. Darin möchten wir unabhängig von Gender und Geschlecht alle Menschen einschließen, die sich für Scrum interessieren oder damit arbeiten – ob Scrum Masterin oder Scrum Master, Developerin oder Developer, Product Ownerin oder Product Owner, Stakeholderin oder Stakeholder.

Wir beginnen unsere Reise wie angekündigt mit einer Frage: Was unterscheidet agile Vorgehensweisen von der klassischen Art zu arbeiten? Betrachten wir zunächst eine Entwicklung nach dem Wasserfallmodell. Dieses klassische Modell beschreibt einen linearen und sequenziellen Entwicklungsansatz mit klaren Zielen für jede Phase der Entwicklung. Das bedeutet: Sobald eine Phase abgeschlossen ist, geht die Entwicklung in die nächste Phase über (*sequenziell*). Abgeschlossene Phasen werden nicht weiter betrachtet (*linear*). Daher stammt auch der Name „Wasserfallmodell“: Das Wasser, das einen Berg herunterfließt, kann nicht mehr zurückfließen.

Am Anfang steht der Kunde, der sich ein Produkt wünscht – etwa eine Software oder eine Website. Die Erwartungen, die er an das Produkt stellt, werden in der sogenannten Anforderungsphase genauestens spezifiziert. Das Ergebnis wird häufig als Lastenheft bezeichnet. Ist diese Phase abgeschlossen, beginnt die Planungs- oder auch Designphase. In dieser überlegt das Team, wie es die aufgenommenen Anforderungen liefern kann. Das Ergebnis dieser Phase ist ein Zeit-Budget-Plan, meistens in Form von Gantt-Charts. Noch bevor also die eigentliche Umsetzung startet, stehen die Eckpfeiler der Entwicklung fest:

- Was genau wird entwickelt?
- Mit welchem Budget?
- Bis wann?

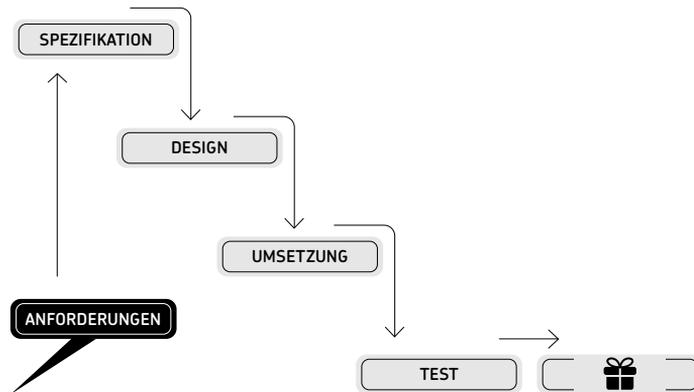
Dabei ergibt sich das sogenannte „magische Dreieck“, an dessen Spitze der Umfang der gewünschten Funktionen steht. Er ist vom Kunden fest vorgegeben und gibt den Ausschlag für die beiden Basispunkte des Dreiecks, nämlich Kosten und benötigte Zeit, die vom Entwicklungsteam geschätzt werden. Der Zweck des magischen Dreiecks ist, dem Team zu verdeutlichen, was gesetzt und was geschätzt ist. Auf diese Weise kann das Team Entscheidungen treffen und bei Bedarf Kompromisse mit dem Kunden verhandeln. Für den Kunden bedeutet das meist, dass mit Lieferverzögerungen zu rechnen ist – oder das Produkt mehr kostet als erwartet, da für eine rechtzeitige Lieferung zusätzliche Entwickler eingestellt werden müssen. Sicherheit besteht nur bei dem Endprodukt, was am Ende herauskommen soll. Alles andere muss sich flexibel anpassen.

Erst nach Abschluss der Planungsphase beginnt die Umsetzung. Durchschnittlich 75% der Entwicklungszeit werden in diese Phase investiert. Im Anschluss wird nur noch getestet, ob die anfangs festgelegte Qualität erreicht wurde. Das Qualitätsmanagement prüft somit erst zum Schluss, ob das Produkt fehlerfrei funktioniert und den Anforderungen entspricht.

Am Ende der Entwicklung hat das Produkt alle Stufen des Wasserfalls durchlaufen. Die Software, Website oder App ist fertiggestellt und wird dem Kunden ausgeliefert. Häufig kommt der Kunde erst jetzt mit seinem Wunschprodukt in Kontakt. Doch was ein Grund zur Freude sein sollte, führt nicht selten zu Unzufriedenheit. Oft entspricht die fertige Software nämlich nicht (mehr) den Erwartungen des Kunden. Über die lange Entwicklungszeit können sich Anforderungen geändert haben, die Prioritäten liegen mittlerweile woanders oder die Software hat den finalen Test nicht bestanden. Im schlimmsten Fall ist das ganze Produkt obsolet geworden. Zudem dauert die Entwicklung bis zu diesem Zeitpunkt meist länger und kostet dadurch mehr als veranschlagt. Das führt zu weiterer Frustration.

Aus diesem Grund zeigt sich das Wasserfallvorgehen in vielen Fällen problembehaftet. Man spricht vom sogenannten U-Boot-Effekt: Anfangs wird der Kunde stark miteinbezogen. Kunde und Entwicklungsteam stehen im Dialog und erarbeiten gemeinsam die Anforderungen des Produkts. Während der folgenden Entwicklungsphasen kommt es jedoch zum „Abtauchen“. Der Kontakt zwischen Entwicklern und Kunde geht verloren. Dadurch nimmt die eine Seite Änderungen auf der anderen Seite kaum oder gar nicht wahr. Beim U-Boot-Effekt handelt es sich damit um ein Kommunikationsproblem. In diesem klassischen Modell wird der Kunde insgesamt zu wenig in die Produktentwicklung einbezogen, sodass man nie sicher sein kann, ihn am Ende zufriedenzustellen. Das erkannten viele agile Vertreter, so auch die Scrum-Väter Jeff Sutherland und Ken Schwaber. Aus diesem Grund beteiligt Scrum den Kunden regelmäßig und eng am gesamten Entwicklungsprozess.

Auch eine weitere Problematik des Wasserfallmodells wird von Scrum gelöst: Selbst wenn sich neue Anforderungen während der Entwicklungszeit ergeben und kommuniziert werden, können diese im Wasserfallmodell nicht umgesetzt werden. Schließlich wurden die entsprechenden Anforderungen in der Planungsphase nicht berücksichtigt. Es sind also weder Zeit noch Budget dafür vorhanden.



Das klassische Wasserfallmodell

Oder, um in der Analogie zu bleiben: Wasser fließt nicht zurück. So wird die sequenzielle, lineare Aufteilung des Wasserfalls, die eine genaue Planung ermöglichen sollte, zur Falle – und zur Sackgasse für jegliche Innovation.

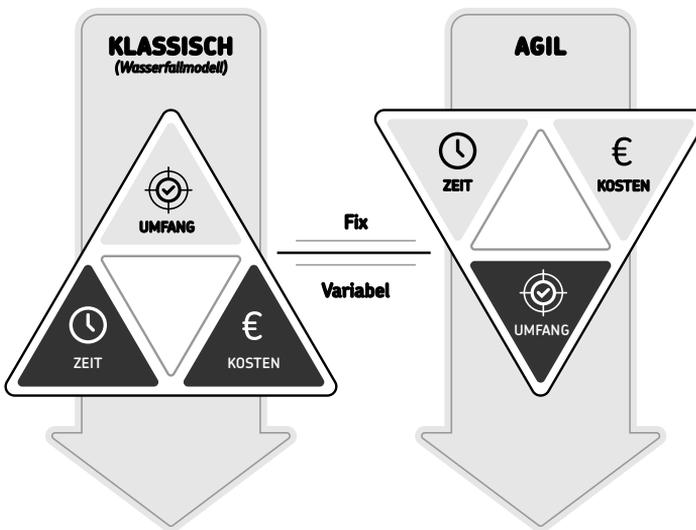
Um die Möglichkeit zu schaffen, veränderte Wünsche des Kunden zeitnah umzusetzen, geht der agile Ansatz *iterativ* und *inkrementell* vor. Das heißt, man arbeitet in zeitlich begrenzten Schleifen, die in Scrum *Sprints* genannt werden. Damit entfallen die klassischen Kaskaden, in denen das „Wasser“ nicht zurückfließen kann. An ihre Stelle treten viele kleine Strudel, die den Mehrwert für den Kunden zum Zentrum haben.

Sprint >
S. 50

Kommunikation und Zusammenarbeit spielen eine tragende Rolle bei agilen Ansätzen wie Scrum. Doch wie gestaltet sich das in der Praxis?

Beim agilen Vorgehen brechen wir das große Produkt, das am Ende herauskommen soll, in kleine Teilprodukte herunter. In jedem Sprint, angefangen mit dem ersten Sprint, stellt das Entwicklungsteam einen kleinen Umfang an Funktionalitäten her. Dabei wird

innerhalb eines Sprints jede Phase aus dem Wasserfallmodell abgebildet: Die Anforderungen werden geplant, umgesetzt und getestet, sodass am Ende des Sprints ein kleines Produkt verfügbar ist. Für das Produkt als Ganzes existieren eine Leitidee sowie ein gewünschtes Enddatum und Budget. Ein besonderes Augenmerk liegt dabei auf der fix vorgegebenen Gesamtzeit. Alle anderen Eckdaten beziehen sich allein auf den laufenden Sprint und das entstehende kleine Teilprodukt. Bei der agilen Lieferung wird das magische Dreieck also einmal umgedreht: Aus den vorgegebenen Kosten und der veranschlagten Zeit, die der Kunde zu gewähren bereit ist, ergibt sich der endgültige Umfang des Produkts.



Das magische Dreieck der klassischen und der agilen Entwicklung im Vergleich

Das im Sprint angefertigte kleine Produkt wird dem Kunden direkt am Sprintende präsentiert. Auf diese Weise wird der Kunde mit einbezogen, kann das Produkt betrachten und sich aktiv an der Entwicklung beteiligen. An diesem Punkt fällt dem Kunden häufig

auf, dass er sich an der ein oder anderen Stelle missverständlich ausgedrückt hatte – oder ihm selbst gewisse Anforderungen noch nicht bewusst waren. Auch kommt es vor, dass Anforderungen vom Team falsch aufgefasst wurden. Das merkt der Kunde nun und kann sein Feedback dazu geben.

Product Backlog >

S. 67

Dieses Feedback ist elementar für das agile Vorgehen, denn so bekommt der U-Boot-Effekt keine Chance; das fertige Teilprodukt wird schließlich regelmäßig dem Kunden gezeigt und in folgenden Sprints mit seinen Wünschen in Einklang gebracht. Dazu wird die Liste der Anforderungen, die in Scrum *Product Backlog* heißt, in jedem Sprint gemäß Kundenfeedback aktualisiert und neu priorisiert. So bleibt der Kontakt zum Kunden permanent bestehen und das Entwicklungsteam kann nicht versehentlich an den Bedürfnissen des Kunden vorbei entwickeln. Der Kunde steht klar im Fokus. Deswegen sind agile Ansätze wie Scrum darauf ausgelegt, ständig Feedback zu generieren, um Mehrwert für den Kunden zu produzieren. Durch den fortlaufenden Austausch steigen zudem das Vertrauen und die Zufriedenheit des Kunden bezüglich der Zusammenarbeit.

Bei der agilen Entwicklung wird der Zeit bis zur Markteinführung (*time to market*) besonderer Wert beigemessen. Das ist die Zeit, bis das *Minimum Viable Product (MVP)* auf dem Markt platziert wird. Ein MVP ist keinesfalls die perfekte und vollkommene Summe aller Wunscheigenschaften. Es handelt sich vielmehr um ein Produkt mit minimal möglichem Funktionsumfang, das zwar nicht alle Vorgaben erfüllt, aber bereits vom Kunden genutzt werden kann. Aufbauend auf dem Feedback, das das Team vom Kunden bzw. vom Markt erhält, kann sich dann die Umsetzung weiterer Features anschließen. Weil die Ressourcen und der Zeitrahmen festgelegt sind, kann das Team leichter auf Veränderungen auf dem Markt eingehen. Der Kunde hingegen bekommt sein Produkt innerhalb der veranschlagten Zeit und des berechneten Budgets mit maximalem Mehrwert.

Noch einen weiteren wichtigen Pluspunkt können agile Teams gegenüber klassisch organisierten Organisationen verbuchen: Verschieben sich die Prioritäten des Kunden oder kann er nicht mehr zahlen, existiert trotzdem bereits ein Produkt, das der Kunde nutzen kann. Wir erinnern uns: Beim klassischen Wasserfallmodell fließen etwa 75 % der Zeit in die Entwicklungsphase. Die Chance, dass der Kunde genau in dieser längsten Phase die Entwicklung „auf Eis legen“ möchte, ist hoch. Ebenso hoch ist die Wahrscheinlichkeit, dass es zu diesem Zeitpunkt nichts außer einem Lastenheft und Gantt-Charts gibt. Schließlich muss das Produkt erst am Ende aller Phasen funktionstüchtig sein. Das Geld des Kunden ist dann völlig verschwendet. Liefern wir aber agil, können wir jederzeit ein funktionierendes Teilprodukt vorweisen, das genutzt oder auch später weiterentwickelt werden kann.

Anders als beim Wasserfallmodell steht beim agilen Vorgehen nicht von Anfang an detailgetreu und spekulationsfrei fest, was am Ende geliefert wird. Vielmehr folgen wir als agiles Team einer Idee, einem Leitstern, der sich Product Goal nennt. Das Product Goal gilt als „Big Picture“ der Entwicklung. Es wird vor Beginn der Entwicklung festgelegt, schärft sich aber und bildet sich weiter aus, je mehr Sprints vergehen. Dadurch wird erst mit verstreichender Zeit sichtbar, was genau die Entwicklung liefern wird. Je mehr Zeit vergeht, desto detaillierter und klarer wird der Blick aufs Ergebnis. Das nennt man *emergent*.

◀ Product Goal

S. 66

Agil vorzugehen birgt zahlreiche Vorteile für Organisationen. Dennoch kursieren grundlegende Missverständnisse in Bezug auf das, was „Agile“ bedeutet. Während viele Organisationen häufige und kurzfristige Änderungen in ihrer Organisationsstruktur als agil bezeichnen, ist in Wahrheit das Gegenteil der Fall. Agilität bedeutet möglichst stabile Teams, die fokussiert an einer konkreten Aufgabe arbeiten. Oft entsteht auch fälschlicherweise der Eindruck, Änderungen an einem Produkt seien folgenlos für Kosten, Zeit und Umfang der Entwicklung. Dies steht im starken Kontrast zu dem, was

Agilität eigentlich ist: eine Umgebung, in der Veränderungen nicht nur akzeptiert, sondern sogar willkommen geheißen werden.

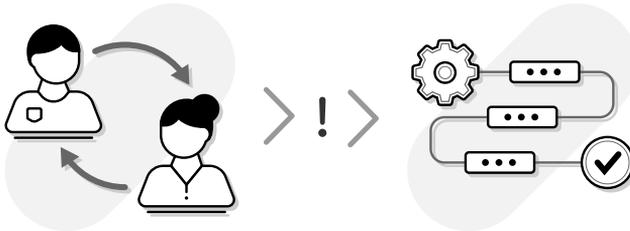
Ob wir klassisch oder agil entwickeln wollen, kommt auf die Organisation, das Entwicklungsteam und den Kunden an, richtet sich aber auch nach dem angestrebten Produkt oder Ziel. So wäre es etwa keine gute Idee, die Statik eines Hauses agil bauen zu wollen. Beim Innenausbau hingegen kommt dies in Betracht. Doch Methoden und Frameworks entwickeln sich ständig weiter. Für Unternehmen lautet die Frage aller Fragen nicht „Liefere wir agil – ja oder nein?“, sondern „Wie viel agil ist für uns möglich?“.

DAS AGILE MANIFEST

Im Februar 2001 erlebten 17 Softwareentwickler in den Bergen Utahs eine vermutlich unvergessliche Skifreizeit. Zweck der Versammlung war, einen gemeinsamen Wertekanon für die Softwareentwicklung zu definieren. Viele der damaligen Teilnehmer arbeiteten bereits zu dieser Zeit mit agilen Methoden und gelten als Vordenker moderner Entwicklungsansätze – so auch Jeff Sutherland und Ken Schwaber, die Gründer von Scrum. Nach anfänglicher Skepsis, ob sich Menschen mit derart unterschiedlichen Denkweisen auf etwas Substantielles einigen könnten, kristallisierten sich vier starke gemeinsame Leitideen sowie zwölf Grundprinzipien heraus. Die Entwickler schrieben ihr Agiles Manifest nieder und veröffentlichten es im Internet, wo es noch heute unter agilemanifesto.org eingesehen werden kann. Die definierten Werte dienen bis heute allen agilen Ansätzen als Basis. So formt das agile Mindset eine Art Dach, unter welchem sich Frameworks und Best Practices wie Scrum, Nexus oder Kanban zusammenfinden.

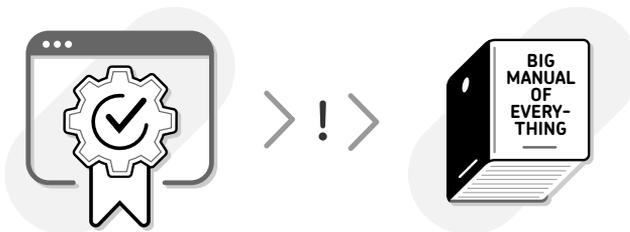
Im Agilen Manifest finden sich vier Kernwerte. Sie wurden ursprünglich für die Softwareentwicklung definiert, gelten im Rahmen agiler Vorgehensweisen aber branchenübergreifend:

1. Individuen und Interaktionen sind wichtiger als Prozesse und Werkzeuge.



Bei der Entwicklung wird Software oder ein anderes Produkt erstellt. Dieses Produkt wird von Menschen für Menschen entwickelt. Das heißt: Die Interaktion zwischen Individuen zählt mehr als Prozesse. Das bedeutet nicht, dass Prozesse und Werkzeuge unwichtig sind. Sie werden immer noch gebraucht, um mit Menschen in Kontakt zu treten – beispielsweise in Form von Kommunikations-tools. Doch die Individuen werden vom agilen Ansatz deutlich in den Vordergrund gerückt.

2. Ein funktionierendes Produkt ist wichtiger als eine umfassende Dokumentation.



Das funktionsfähige Produkt am Ende eines Sprints ist essenziell. Der Kunde soll etwas sehen und es potenziell benutzen können. Eine Dokumentation über die erledigte Arbeit ist zwar notwendig, rückt aber vergleichsweise in den Hintergrund. Ein Augenmerk liegt auch auf der Formulierung „umfassend“: Es wird nur so viel wie nötig und damit so wenig wie möglich dokumentiert. Durch

diesen Leitsatz kommt es in der Praxis häufig dazu, dass die Arbeit an einem Feature in einem Sprint fertiggestellt wird, die Dokumentation jedoch erst im darauffolgenden Sprint erfolgt.

3. Die Zusammenarbeit mit dem Kunden ist wichtiger als Vertragsverhandlungen.



Für eine reibungslose Kooperation werden Vereinbarungen gebraucht. Doch dabei sollte der Kunde stets im Mittelpunkt stehen. Er muss gewillt sein, das Produkt in regelmäßigen Abständen in Augenschein zu nehmen und Feedback zu geben. Erst dadurch erwächst die vertrauensvolle Beziehung zwischen ihm und dem entwickelnden Team.

Es soll auch nicht für jede neue Anforderung ein neuer Vertrag nötig werden. Schließlich kommt es spätestens alle paar Wochen zum Austausch zwischen Kunde und Team, aus dem sich veränderte Entwicklungsideen ergeben. Geht es zum Beispiel nur darum, die Farbgebung oder den Schriftgrad auf einer Webseite anzupassen, brauchen wir keinen neuen Vertrag. Das würde zu erhöhter Frustration zwischen den Parteien führen und von Misstrauen zeugen. Stattdessen geht es um die Beziehung zwischen Team und Kunde, die durch die rege und kontinuierliche Zusammenarbeit gestärkt wird.

4. Das Reagieren auf Veränderung ist wichtiger als das Befolgen eines Plans.



Das agile Tun basiert auf einem Plan: Es existiert eine übergreifende Idee, in welche Richtung sich das Produkt entwickeln soll. Auch gibt es einen Plan für jeden einzelnen Sprint. Doch durch kurze Innovationszyklen und sich ständig ändernde Marktanforderungen müssen wir uns in der Produktentwicklung auf stetigen Wandel einstellen. Veränderungen werden erwartet, entgegengenommen und nötigenfalls zügig umgesetzt. Auf diese Weise stellt die agile Idee Reaktionsfähigkeit auf den Markt und den Kunden über strikte Pläne und Vorgaben.

Wenn man das Agile Manifest betrachtet, sieht man also vor allem eine Gewichtung. Prozesse, Verträge und Pläne sollen nicht völlig aus dem Arbeitsumfeld verschwinden. Die Aspekte auf der einen Seite der Waage werden lediglich als wichtiger erachtet als die auf der anderen Seite. Diese Sichtweise erfordert häufig einen Paradigmenwechsel: Zum einen muss das Team Agilität verstehen und leben, zum anderen muss der Kunde Teil der Produktentwicklung sein wollen. Wenn beide Seiten im Sinne der agilen Werte zusammenarbeiten, entsteht der größte Mehrwert in kürzester Zeit.

Nun haben wir uns grundlegend mit den Ideen und Vorteilen agiler Ansätze vertraut gemacht. Sie bilden die Basis für Scrum – denn Scrum ist zwar agil, doch nicht jeder agile Ansatz entspricht automatisch Scrum. Im Folgenden geht es daher ganz um die Eigenarten von Scrum.

WAS IST SCRUM?

Bei Scrum handelt es sich um ein leichtgewichtiges Rahmenwerk, das Menschen, Teams und Organisationen hilft, Wert zu generieren. Es ist ein iterativer, zeitgesteuerter Ansatz zur Produktlieferung. Im Rahmen von Scrum können Menschen komplexe Probleme mit adaptiven Lösungen angehen. Dabei liefern sie effizient und kreativ Produkte mit dem höchstmöglichen Kundennutzen. Als Grundlage und „heilige Schrift“ im übertragenen Sinne gilt dabei der Scrum Guide, der von den beiden Scrum-Begründern fortlaufend aktualisiert und im Internet kostenfrei zur Verfügung gestellt wird.

Weil Scrum kein starres Regelwerk, sondern eine Art Rahmen darstellt, spricht man von einem *Framework*. Der Unterschied zu einer *Methode* besteht darin, dass diese das „Wie“ erläutert. Ein Rahmenwerk wie Scrum hingegen beschreibt allein das „Was“. Scrum ist also ein Framework zur Produktentwicklung und keine Projektmanagementmethode. Das lässt sich leicht daran erkennen, dass es die Rolle des Projektmanagers in Scrum gar nicht gibt. Doch dazu später mehr.

**Gibt es einen
Projektmanager in
Scrum? >**
S. 38

Dass Scrum als Framework fungiert, bedeutet: Alle Aspekte aus dem Scrum Guide müssen erfüllt werden, um von Scrum sprechen zu können. Sie bilden den Rahmen und sind essenziell für den Erfolg von Scrum. Dennoch lassen die einzelnen Elemente den Teams die Freiheit, das Rahmenwerk mit eigenen Erfahrungen, Best Practices und Methoden zu füllen. Man könnte auch sagen: Die Teams malen ihr eigenes Bild in den Rahmen hinein. Wird allerdings eines der Rahmenelemente entnommen, fällt das ganze Bild in sich zusammen. Dann kann nicht mehr von Scrum gesprochen werden, sondern lediglich von einem Framework in Anlehnung an Scrum.

Da Scrum als Framework vor allem das „Was?“, nicht aber das „Wie?“ beleuchtet, fällt es vielen Teams schwer, Scrum umzusetzen. Zwar ist Scrum an sich verständlich und der Scrum Guide mit nur wenigen Seiten sehr übersichtlich. Doch in der Praxis ist Scrum

schwer zu meistern. Das liegt nicht zuletzt an den Werten, den Grundprinzipien und der Denkweise, die Scrum zugrunde liegen.

Scrum zu implementieren, ist ein langer Weg. Es von heute auf morgen zu schaffen, ist praktisch unmöglich. Ob die Philosophie, Theorie und Struktur von Scrum der eigenen Organisation oder dem eigenen Team dabei helfen, Ziele zu erreichen und Wert zu schaffen, muss durch Experimentieren herausgefunden werden.

Eine Organisation benötigt zur Einführung von Scrum bereits eine gewisse Offenheit und positive Fehlerkultur, da das Framework durch seine kontinuierlichen Rückkopplungsschleifen viel eher Herausforderungen und Probleme aufdeckt als das Wasserfallmodell. Zudem ist es ein erklärtes Ziel, schnell zu scheitern: Wenn überhaupt ein Misserfolg eintritt, sollte das frühzeitig passieren, da das Team aus der Erfahrung lernt und danach erfolgreicher voranschreiten kann.

Das Herzstück von Scrum bilden das Scrum Team und die dazugehörigen Rollen, Ereignisse, Artefakte und Regeln. Jeder einzelne Aspekt innerhalb des Rahmenwerks verfolgt einen bestimmten Zweck und ist damit wesentlich für den Erfolg der Entwicklung.

◀ Wasserfallmodell vs. agiles Vorgehen

S. 18

Austausch und aktuelle Impulse

Agiles Arbeiten bedeutet, dass man voneinander lernt – nicht nur innerhalb eines Teams oder einer Organisation. Ob in Online-Foren, auf Vorträgen oder bei Events, überall können wir Menschen begegnen, die uns an ihren Erfahrungen teilhaben lassen. Eine hervorragende Anlaufstelle für frische Impulse aus erster Hand bieten auch Websites wie das SERVIEW Infocenter. Von hier gibt es direkten Zugriff auf Artikel, Podcasts, Videos, Downloads und diverse Neuigkeiten aus der agilen Welt.



[www.serview.de/
infocenter](http://www.serview.de/infocenter)