

Die Werte

Nachdem sie im vorigen Kapitel bereits kurz vorgestellt wurden, werden wir die Werte, auf denen Scrum basiert, hier nun ausführlich beschreiben. Erst durch das Zusammenspiel von Werten, Prinzipien und Praktiken entfaltet Scrum sein volles Potenzial. Deshalb haben wir den Werten ein eigenes Kapitel gewidmet. Wir definieren zunächst den Begriff »Wert«, bevor wir die fünf Scrum-Werte vorstellen und deren praktische Bedeutung anhand von Beispielen illustrieren. Abschließend zeigen wir, wie die Werte, Prinzipien und Praktiken ineinander greifen, um mit Scrum erfolgreich arbeiten zu können.

Was sind Werte?

Der Kern aller agilen Vorgehensmodelle ist ein Wertesystem. Diese Werte geben den Prinzipien und Praktiken von Scrum eine Bedeutung und der Mechanik von Scrum einen Sinn.

Wenn wir hier den Begriff »Werte« verwenden, meinen wir immaterielle, moralische (oder innere) Werte, die auf Werterfahrungen beruhen. Für den deutschen Soziologen und Sozialphilosophen Hans Joas sind Werte attraktiv-motivierend. Im Gegensatz dazu stehen die Normen, die restriktiv-obligatorisch sind. Während Normen bestimmte Handlungen ausschließen, erschließen Werte den Menschen neue Handlungsmöglichkeiten.

Die Werte in Scrum

Scrum gründet auf einem Fundament aus fünf Werten (siehe Scrum Guide):

- **Selbstverpflichtung (Commitment):** »[Alle Beteiligten] verpflichten sich persönlich dazu, die Ziele des Scrum-Teams zu erreichen.« – Seien Sie willens, sich einem Ziel zu verpflichten. Scrum gibt Ihnen die nötigen Befugnisse, um diese Verpflichtung zu erfüllen.
- **Mut (Courage):** »Die Mitglieder des Scrum-Teams haben den Mut, das Richtige zu tun und an schwierigen Problemen zu arbeiten.« – Haben Sie den Mut, sich einem Ziel zu verpflichten, zu handeln, Offenheit zu zeigen und für gegenseitigen Respekt einzutreten.
- **Fokus (Focus):** »Jeder fokussiert sich auf die Arbeit im Sprint und die Ziele des Scrum-Teams.« – Erledigen Sie Ihre Arbeit. Verwenden Sie all Ihre Energie und Erfahrung auf die Aufgabe, zu der Sie sich verpflichtet haben. Machen Sie sich um alles andere keine Sorgen.
- **Offenheit (Openness):** »Das Scrum-Team und seine Stakeholder sind sich einig, offen mit allen Belangen ihrer Arbeit und den damit verbundenen Herausforderungen umzugehen.« – Machen Sie alle Informationen über das Projekt für alle sichtbar.
- **Respekt (Respect):** »Mitglieder von Scrum-Teams respektieren sich gegenseitig als fähige, eigenverantwortliche Individuen.« – Akzeptieren Sie die Prägung jedes Individuums durch seinen persönlichen Hintergrund und seine Erfahrungen. Respektieren Sie die verschiedenen Menschen, die sich zu einem Team zusammengeschlossen haben.

Auch eXtreme Programming [Beck 2004], eine leichtgewichtige und agile Softwareentwicklungsmethode, basiert auf fünf Werten. Der Wertekatalog ist bewusst klein gewählt, damit er von allen, die in agilen Projekten arbeiten, gleichermaßen akzeptiert und gelebt wird. Wer einmal ernsthaft versucht hat, sein Handeln in einem Scrum-Projekt konsequent an den fünf Werten auszurichten, wird

festgestellt haben, wie anspruchsvoll diese wertorientierte Arbeitsweise ist. Wenn es dem Scrum-Team jedoch gelingt, diese fünf Werte zu verkörpern und zu leben, erweckt es die Scrum-Säulen Transparenz, Überprüfung (Inspection) und Anpassung (Adaptation) zum Leben und schafft unter allen Beteiligten Vertrauen und Zuversicht. Betrachten wir deshalb die Scrum-Werte etwas genauer.

Selbstverpflichtung (Commitment)

»[Alle Beteiligten] verpflichten sich persönlich dazu, die Ziele des Scrum-Teams zu erreichen.«

Den ersten Scrum-Wert »Commitment« ins Deutsche zu übersetzen, ist nicht leicht, da dieser Begriff sehr vielschichtig und die Bedeutung im Scrum-Kontext sehr speziell ist. Hier bezeichnet »Commitment« insbesondere die Zusage des Entwicklungsteams, das gemeinsam festgelegte Sprint-Ziel zu erreichen. Dazu gibt das Entwicklungsteam im Sprint Planning eine Prognose (Forecast) ab. Diese nach bestem Wissen und Gewissen ermittelte Liste von Backlog Items (das Selected Product Backlog) geht dann als Sprint Backlog in den nächsten Sprint ein. Wurde das Sprint-Ziel am Ende des Sprints trotz kontinuierlicher Überprüfung und Anpassung des Forecast nicht erreicht, kann das ganz unterschiedliche Gründe haben, die es im Rahmen einer Retrospektive zu untersuchen gilt.

Ein gutes Entwicklungsteam steht zu seinem Produkt. Es will gemeinsam mit dem Product Owner ein Produkt mit größtmöglichen Nutzen schaffen, das den gestellten Anforderungen gerecht wird. Diese Anforderungen werden entsprechend ihrer Priorisierung der Reihe nach umgesetzt – Sprint für Sprint. Und weil ein Entwicklungsteam weiß, wie wichtig es ist (und wie gut es tut), wenn am Ende das Sprint-Ziel erreicht ist, wird es bei der Sprint-Planung sorgfältig zu Werke gehen. In diesem Sinne weitet die unmittelbar unter der Überschrift dieses Abschnitts kursiv wiedergegebene Fassung aus dem Scrum Guide 2017 den Begriff Commitment auf die gesamte Zusammenarbeit innerhalb des Scrum-Teams aus.

Mut (Courage)

»Die Mitglieder des Scrum-Teams haben den Mut, das Richtige zu tun und an schwierigen Problemen zu arbeiten.«

Mut im beruflichen Kontext? Was auf den ersten Blick unpassend erscheinen mag, entpuppt sich bei näherer Betrachtung als wichtiger Wert in einem kollaborativen Umfeld. Je intensiver Menschen zusammenarbeiten, desto größer ist der Einfluss zwischenmenschlicher Befindlichkeiten. Persönliche oder fachliche Differenzen, ungenaue Kommunikation, übertriebener oder fehlender Ehrgeiz – es gibt viele Gründe dafür, dass die Stimmung im Team kippen kann. Solche Situationen aufzulösen und zu neuer Gemeinsamkeit zu finden, erfordert nicht nur Erfahrung und Fingerspitzengefühl, sondern vor allem den Mut, die Probleme offen anzusprechen.

Man darf Mut nicht mit Übermut verwechseln. Mut ist kalkuliertes Risiko, basierend auf Erfahrung und Urteilsvermögen. Mutige Handlungen haben oft positive Auswirkungen. Übermut hingegen tut, wie schon der Volksmund sagt, selten gut.

Mut bedarf es aber nicht nur zur Auflösung von Konflikten. Wenn ein Teammitglied seine eigene Meinung z.B. zur Softwarearchitektur oder einer Prozessverbesserung konsequent vertritt, auch gegen den Widerstand anderer Teammitglieder, dann ist das mutig. Ein Scrum Master muss oft allen Mut zusammennehmen, um sein Team vor störenden Einflüssen von außen zu schützen. Und ein Product Owner beweist Mut, wenn er beispielsweise eine fachliche Anforderung eines hochrangigen Stakeholders ablehnt, weil diese nicht auf die Produktvision einzahlt.

Nicht alle Menschen sind von Natur aus mutig, und die meisten können oder wollen es auch gar nicht sein. Unter gewissen Rahmenbedingungen können aber auch solche Personen über ihren Schatten springen und Wahrheiten offen aussprechen. Zu diesen Rahmenbedingungen zählt der weiter unten beschriebene Respekt als Grundwert, aber auch ein offener Umgang mit Fehlern. Es gibt Organisationen, die es sich zum Ziel gesetzt haben, aus den eigenen Fehlern zu lernen und diese Erkenntnis allen Mitgliedern der Organisation

zur Verfügung zu stellen. Dazu muss jeder bereit sein, eigene Fehler zuzugeben und gemeinsam mit den Kolleginnen und Kollegen an der Fehlerbehebung zu arbeiten. Wer eine solche Lernkultur leben darf, für den ist es selbstverständlich, mutig über Fehler zu sprechen. In diesem Umfeld fällt es leichter, auch andere Missstände anzusprechen, die keine Fehler sind, aber beispielsweise Verschwendungen bewirken oder ein großes Risiko für die Organisation darstellen. Mit dieser Grundeinstellung wird sich eine Organisation positiv weiterentwickeln – die Welt gehört den Mutigen!

Fokus (Focus)

»Jeder fokussiert sich auf die Arbeit im Sprint und die Ziele des Scrum-Teams.«

Der Sprint ist gestartet. Nun gilt es, die in das Sprint Backlog übernommenen Product Backlog Items umzusetzen. Das ist die Aufgabe des Entwicklungsteams – nicht mehr und nicht weniger. Deshalb sollen sich alle Teammitglieder auf diese Aufgabe konzentrieren. Das ist leichter gesagt als getan. Tagesgeschäft, Tätigkeiten für andere Projekte und »Kannste mal eben«-Aufgaben machen dem um Fokus bemühten Entwickler das Leben schwer.

Mehrere wissenschaftliche Untersuchungen (z.B. [Baethge 2010]) haben gezeigt, dass sowohl das gleichzeitige Bearbeiten mehrerer Aufgaben (Multitasking) als auch eine hohe Frequenz von Arbeitsunterbrechungen negative Auswirkungen haben. Die Qualität der Arbeitsleistung nimmt ab, unter anderem weil schneller gearbeitet wird, um die durch Unterbrechungen verlorene Zeit zu kompensieren. Trotzdem ist die Gesamtbearbeitungsdauer für jede Einzelaufgabe beim Multitasking höher als bei einer unterbrechungsfreien sequenziellen Abarbeitung der Aufgaben. Eine häufige Folge von Multitasking und Unterbrechungen ist Stress. Eine fokussierte Arbeitsweise sorgt dafür, dass die Zahl der Unterbrechungen abnimmt.

Scrum unterstützt eine fokussierte Arbeitsweise. Ein ideales Scrum-Team arbeitet für genau ein Projekt und ist personell stabil. Die Aufgaben sind klar und ausreichend konkret benannt, sodass einer

konzentrierten Bearbeitung nichts im Weg steht. Die Definition of Done beschreibt, wann eine Aufgabe als erledigt betrachtet werden kann. Die Backlog Items werden der Reihe nach abgearbeitet. Idealerweise befindet sich zu jedem Zeitpunkt im Sprint genau ein Backlog Item in Bearbeitung. Ein regelmäßiger Austausch ist durch das Daily Scrum garantiert und fest in den Tagesablauf integriert, hat also nicht den Charakter einer (ungeplanten) Unterbrechung.

Nicht nur das Entwicklungsteam arbeitet fokussiert, auch der Product Owner hat das Produkt im Fokus. Er konzentriert sich auf die Beschreibung der Ideen, Konzepte und Eigenschaften des Produkts. Grundsätzlich wird er sich möglichst viel Funktionalität wünschen. Demgegenüber muss der Scrum Master das Entwicklungsteam darin unterstützen, dass die Funktionen des Produkts in der geforderten Qualität und innerhalb der durch die Sprint-Länge vorgegebenen Zeit realisiert werden. Im Zweifelsfall wird er seinem Team empfehlen, sich gegen die Aufnahme eines weiteren Backlog Item in das Selected Product Backlog auszusprechen, wenn dadurch die Erledigung der bereits vom Team zugesagten Backlog Items gefährdet wäre.

Der Fokus des Scrum Masters liegt auf der Einhaltung des Scrum-Rahmenwerks und der Teamregeln. Er tut alles, um die Produktivität des Teams zu gewährleisten und zu verbessern. Dazu berät er auch alle Personen außerhalb des Teams, die Einfluss auf die Arbeit des Teams (oder Interesse an dessen Arbeitsweise) haben.

Offenheit (Openness)

»Das Scrum-Team und seine Stakeholder sind sich einig, offen mit allen Belangen ihrer Arbeit und den damit verbundenen Herausforderungen umzugehen.«

Offenheit hat mehrere Facetten. Eine davon ist die Offenheit der Menschen gegenüber ihrer Umwelt. Man braucht z.B. einen offenen Blick auf andere beteiligte Personen und auf das, was im Umfeld geschieht. Nur so kann man Probleme, Risiken, Chancen und sonstige Veränderungen erkennen und danach gegebenenfalls sein Handeln ausrichten.

Ein anderer, aktiver Aspekt von Offenheit ist die Informationstransparenz. Der Forderung nach Offenheit im Umgang mit Informationen liegt die Erkenntnis zugrunde, dass ein Mensch nur dann in der Lage ist, eine fundierte Entscheidung zu treffen, wenn er Zugang zu allen relevanten Informationen hat. Je einfacher diese Informationen zugänglich sind und je offener sie präsentiert werden, desto intensiver werden sie genutzt. Ein Beispiel: Anstatt den Projektstatus in einem elektronischen Planungswerkzeug zu pflegen, findet man in vielen Scrum-Projekträumen ein Taskboard, an dem die Backlog Items und Tasks des laufenden Sprints als Pappkarten oder Haftnotizzettel in einer der drei Spalten *To Do*, *In Progress* und *Done* hängen. So lässt sich auch für einen Außenstehenden der Status des Projekts (bezogen auf den laufenden Sprint) auf einen Blick erkennen. Ein anderes Beispiel: Das Impediment Backlog ist eine offen geführte Liste der Hindernisse, die das Entwicklungsteam in seiner Arbeit beeinträchtigen. Es ist Aufgabe des Scrum Masters, für die Beseitigung dieser Hindernisse zu sorgen. Da aber jeder die Liste einsehen kann, ist es möglich, dass ein Hindernis von einer anderen Person aus dem Weg geräumt wird. Man kann schließlich nur die Probleme lösen, von deren Existenz man weiß.

Die Offenheit im Umgang mit Informationen hat eine Kehrseite, die wir jedoch für eine Stärke halten: Transparenz wirft Fragen auf. Wenn der Projektstatus für alle sichtbar in einem Burndown Chart entlang einer Ideallinie dokumentiert wird, sind Abweichungen – nach oben wie auch nach unten – sofort erkennbar. In einem Unternehmen, das nur gut laufende Projekte kennt (weil es die weniger gut laufenden entsprechend »frisiert«), kann ein Scrum-Projekt unangenehm auffallen. Es wird sich von traditionellen Projektleitern die Frage gefallen lassen müssen, warum es im Vergleich zu den traditionellen Projekten so schlecht läuft. Und das Management könnte sich beschweren, dass dieses Projekt das erste in der Geschichte des Unternehmens sei, das nicht ständig im grünen Bereich ist. Dabei wissen die Projektleiter, dass auch in ihren Projekten nicht alles glänzt. Sie haben aber immer bis zur nächsten Lenkungsausschusssitzung (oder einem vergleichbaren Projekt-

statusmeeting) Zeit, zwischenzeitliche Abweichungen wieder einzufangen, weil bis dahin niemand den aktuellen Projektstatus erfragt. Anders das Scrum-Projekt: Es ist tagesgenau auskunftsähig – nicht etwa um dem Management zu gefallen, sondern in erster Linie weil das Scrum-Team selbst daran interessiert ist, zu wissen, wie es um das eigene Projekt steht. Je früher es eine Fehlentwicklung erkennt, desto effektiver kann das Scrum-Team Gegenmaßnahmen ergreifen. Das funktioniert am besten in einer Unternehmenskultur, die Änderungen willkommen heißt und Fehler als Chance zum Lernen begreift. Wer nur positive Statusberichte bekommen möchte, muss akzeptieren, dass er betrogen wird. Und wer nicht am tatsächlichen Zustand seiner Projekte interessiert ist, der wird sich wundern, warum trotz vermeintlich erfolgreicher Projekte der Erfolg auf der Strecke bleibt. Transparenz wirft Fragen auf – aber Transparenz macht vor allem Probleme früh sichtbar und somit behebbar. Das ist die wahre Stärke von Scrum und anderen agilen Methoden.

Respekt (Respect)

»Mitglieder von Scrum-Teams respektieren sich gegenseitig als fähige, eigenverantwortliche Individuen.«

Ein respektvoller Umgang im privaten wie im beruflichen Leben sollte eigentlich selbstverständlich sein. Die Wirklichkeit sieht oft anders aus. Da stehen Programmierhelden im Rampenlicht, die den Rest des Teams im Schatten stehen lassen. Wir erleben paarweises Programmieren, bei dem ein Entwickler seine überlegenen Programmiekünste zum Besten gibt, anstatt seinem Partner zu helfen, besser zu werden. Und natürlich gibt es die vielen kleinen und größeren versteckten und offenen »Nickeligkeiten« – genau wie im Fußball, der diesen Begriff populär gemacht hat.

In Scrum-Teams wird sehr eng und intensiv miteinander gearbeitet. Das funktioniert nur, wenn sich die Teammitglieder gegenseitig respektieren, ihre Verschiedenheit akzeptieren, die Stärken kennen und nutzen, die Schwächen verzeihen oder gemeinsam daran arbeiten und auf ein gemeinsames Ziel hinarbeiten. Daher wird ein

Scrum-Team ohne gegenseitigen Respekt keine Erfolge erzielen können.

Viele Entwickler äußern den Wunsch nach einem homogenen Team, in dem alle ähnlich »ticken«. Dabei steckt oft in der Verschiedenheit der Teammitglieder der Schlüssel zum Erfolg. Scrum-Teams sind interdisziplinär: Sie vereinen alle Kompetenzen, die erforderlich sind, um die anstehenden Aufgaben ohne Unterstützung von außerhalb zu erledigen. In Scrum-Teams treffen leidenschaftliche Programmierer auf begeisterte Tester, kreative Webdesigner auf pfiffige Systemadministratoren, großartige Grafiker auf visionäre Product Owner. Nur gemeinsam können sie ihr Ziel erreichen und ein Produkt entwickeln, das den Kunden gefällt.

Idealerweise bringen die Teammitglieder neben ihrem Expertenwissen ein Interesse an den Spezialgebieten der anderen mit. Sie wollen voneinander lernen und ihren eigenen Horizont erweitern – nicht nur aus Neugierde, sondern auch, um die anderen Teammitglieder unterstützen zu können. Solche Persönlichkeiten werden T-förmig (T-shaped) genannt, weil sie über ein breites Wissen verfügen (waagerechter T-Strich) und zugleich Experte auf (mindestens) einem Gebiet sind (senkrechter T-Strich). Zunehmend finden sich auch Pi-förmige (Pi-shaped) bis Kamm-förmige (Comb-shaped) Persönlichkeiten, die sich durch mehr als nur ein Spezialgebiet bei dennoch breitem Basiswissen auszeichnen.

Ein schönes Beispiel dafür, wie man respektvoll miteinander umgeht, liefert Norman L. Kerth, wenn er Scrum-Teams seine »Retrospective Prime Directive« als Leitlinie für Teamretrospektiven empfiehlt ([Koschek 2014], nach [Kerth 2001]):

»Ganz egal, was wir entdecken werden: Wir verstehen und glauben zutiefst, dass jede(r) nach besten Kräften gearbeitet hat, wenn man den aktuellen Wissensstand, die Fähigkeiten und Fertigkeiten, die verfügbaren Ressourcen und die derzeitige Situation zugrunde legt.«

Teams, deren Mitglieder mit dieser Grundeinstellung durchs (Projekt-)Leben gehen, sind in der Lage, das Beste aus sich herauszuholen und dabei gut miteinander auszukommen. Ein gutes Team ist deshalb mehr als die Summe seiner Mitglieder.

Das Zusammenspiel von Werten, Prinzipien und Praktiken

Aus den Scrum-Werten können Prinzipien abgeleitet werden, in denen wiederum die Scrum-Praktiken ihren Ursprung haben. Werte, Prinzipien und Praktiken spannen somit den Bogen von den Handlungsmöglichkeiten bis hin zu konkreten Handlungsanweisungen.

In der Scrum-Literatur gibt es keine Einigung darüber, welche agilen Prinzipien das Scrum-Framework ausmachen. Eine große Rolle spielen mit Sicherheit die zwölf Prinzipien des Agilen Manifests, die wir in »Das Agile Manifest« auf Seite 6 kurz vorgestellt haben. Tobias Mayer beschreibt in seinem Essay »The Soul of Scrum« (in [Mayer 2018]) die aus seiner Sicht wichtigsten inhärenten Prinzipien von Scrum:

- *Fokus (Focus)*: Dieser agile Wert hilft Teams, sich zu jeder Zeit auf die Erledigung genau einer Aufgabe zu konzentrieren. Ein gut zusammengestelltes interdisziplinäres Team vorausgesetzt, steht alles zur Verfügung, was dazu erforderlich ist. Es gibt keinen Konkurrenzkampf um Zeit, Ressourcen und Personal – alle Energie wird zielgerichtet und nutzbringend auf die Erledigung der Aufgabe verwendet.
- *Ausrichtung (Alignment)*: Der regelmäßige persönliche Austausch des Entwicklungsteams mit den Stakeholdern und Anwendern sorgt dafür, dass Missverständnisse vermieden und Fragen schnell geklärt werden, sodass am Ende das Richtige entsteht.
- *Kunstvolles Schaffen (Artful Making)*: In dem gleichnamigen Buch [Austin 2003] ziehen Rob Austin und Lee Devin Parallelen zwischen dem künstlerischen Bereich und der aus ihrer Sicht kreativen Softwareentwicklung. Sie haben vier Qualitäten identifiziert, die ein Softwareentwicklungsteam aufweisen sollte: Loslassen (*Release*) innerhalb eines klar definierten Rahmens, eine Zusammenarbeit (*Collaboration*), die zu einem großen Ganzen (*Ensemble*) führt, das größer ist als die Summe seiner Teile, und eine Haltung wie im (Schau-)Spiel (*Play*), wo die qualitativ hochwertige Darbietung wichtiger ist als die Leistung des Einzelnen.

- *Selbstorganisation (Self-organization)*: Kleine, interdisziplinäre und selbstorganisierende Teams (drei bis neun Personen) kümmern sich eigenverantwortlich darum, wie sie die vom Product Owner beschriebenen Anforderungen umsetzen. Die Entscheidungsfreiheit hinsichtlich der Umsetzung beschränkt sich nicht nur auf die Wahl der geeigneten Technologien und Architekturen, sondern umfasst auch die Ausgestaltung des Entwicklungsprozesses. Warum sollte diese Aufgabe jemand anderes übernehmen als jene Menschen, die Experten auf diesen Gebieten sind, weil sie täglich damit arbeiten?
- *Rhythmus (Rhythm)*: Je größer eine Aufgabe ist, desto schwieriger lässt sich deren Inhalt und Umfang abschätzen. Deshalb verwendet Scrum kleine Zeiteinheiten (Timeboxen) und fügt diese in immer wiederkehrender Folge zusammen. So entsteht ein Rhythmus, der es Unternehmen erlaubt, schnell voranzuschreiten und schnell zu lernen.

Scrum selbst vertraut laut Mayer auf diese fünf Prinzipien: Ausrichtung und Rhythmus sind die Basis des Empirismus (siehe Abschnitt »Scrum ist angewandter Empirismus« auf Seite 3). Rhythmus entsteht durch feste Zeiteinheiten und regelmäßige Synchronisation. Fokus hilft, ein Backlog zu verwalten und sorgt gemeinsam mit einer klaren Ausrichtung für Nutzen stiftende Sprint-Ziele. Kunstvolle Herstellung und Selbstorganisation schaffen den idealen Rahmen für Teams, um fachliche Anforderungen exzellent zu lösen.

Das Prinzip des Empirismus findet beispielsweise in der Sprint-Retrospektive seine Anwendung. Diese Scrum-Praktik dient dazu, Erkenntnisse aus dem vergangenen Entwicklungszyklus zu gewinnen und in Verbesserungsmaßnahmen für den folgenden Zyklus umzuwandeln. Dazu bedient sich das Team unter anderem historischer Daten über das Projekt. Solche Daten müssen verfügbar sein, erfordern also eine Informationstransparenz, die dem Wert der Offenheit entspricht. Offenheit (Wert) ist also eine Voraussetzung für ein empirisches Vorgehen (Prinzip), beispielsweise in der Sprint-Retrospektive (Praktik) – so funktioniert das Zusammenspiel von Werten, Prinzipien und Praktiken.