

Vertragsrecht und Scrum – ein Überblick

Verfasser

Mari Pissanou, LL.B.

Prof. Dr. Jens M. Schmittmann

FOM Hochschule für Oekonomie & Management Essen

Leimkugelstr. 6, 45141 Essen

jens.schmittmann@fom.de

Klassifizierung

Bürgerliches Recht; Werkvertrag; Beteiligte

Stichworte

Werkvertrag; Dienstvertrag; Pflichten; Scrum; Beteiligte

Abstrakt

Nachdem nun eine zweitinstanzliche Gerichtsentscheidung zur rechtlichen Einordnung von Verträgen, die das Scrum-Verfahren zugrunde legen, ergangen ist, gibt der nachfolgende Beitrag eine kurze rechtliche Einführung und stellt sodann die Grundlagen des Scrum dar.

I. Einführung

Mit einer gewissen zeitlichen Verzögerung werden neuen Techniken und Verfahren auch Gegenstand von Rechtsstreitigkeiten. So hat das OLG Frankfurt am Main (Urteil vom 17. Juli 2017 – 5 U 152/16, K&R 2017, 803 ff.) eine Entscheidung des LG Wiesbaden (Urteil vom 30. November 2016 – 11 O 10/15, CR 2017, 298 = ITRB 2017, 36; vgl. Ernst, CR 2017, 285 ff.) zu Scrum abgeändert und entschieden:

Für die Geltendmachung eines Rücktritts bzw. einer Minderung der Vergütung war zunächst die Setzung einer Frist zur Nacherfüllung erforderlich. Erst wenn diese fruchtlos verstrichen wäre, hätte die Beklagte – wozu jeweils eine weitere Willenserklärung erforderlich gewesen wäre – ggf. ihren Rücktritt von dem Vertrag oder aber eine Minderung des Werklohns erklären können (Leitsatz der Redaktion der Zeitschrift K & R).

Das OLG Frankfurt nimmt daher werkvertragliche Grundsätze an.

II. Scrum

Die aktuelle Rechtsprechung gibt Anlass, sich näher mit „Scrum“ zu befassen.

1. Thematik und Entstehung von Scrum

Bei Scrum handelt es sich um ein agiles Projektmanagement-Verfahren, das oft in der Softwareentwicklung oder auch in anderen Bereichen, z. B. in Bereichen der komplexen Produktentwicklung, angewendet wird (vgl. *Dräther et al.*, Scrum, kurz & gut, Heidelberg, 2013, S. 13). Scrum ist ein Rahmenwerk (Framework), das Menschen bei der Herangehensweise an komplexe Aufgabenstellungen hilft (vgl. Schwaber/Sutherland, *Software in 30 days*, S. 136).

Die Namensgebung von Scrum lässt sich auf Ikujiro Nonaka und Hirotaka Takeuchi zurückführen (vgl. *Schwaber/Beedle*, *Agile Software Development*, S. 1). Sie haben für Produktentwicklungsteams, die aus mehreren Fachdisziplinen bestehen, eine Analogie zu dem Gedränge im Rugby (englisch: Scrum) gezogen. Die Erfinder von Scrum, so wie wir es heute kennen, sind Jeff Sutherland und Ken Schwaber (vgl. *Schwaber/Sutherland*, *Software in 30 days*, S. 136). Vor über 20 Jahren haben sie dieses Projektmanagement-Framework kreiert. Sie vertraten die Ansicht, dass die Softwareentwicklung mit dem bis heute noch existenten „Wasserfallmodell“ zu viele Fehler aufweist, ineffektiv und nicht flexibel genug ist, wie Sutherland in seinem Buch „SCRUM – The Art of Doing Twice the Work in Half the Time“ auch beschreibt (vgl. *Sutherland*, *SCRUM*, S. 32).

Scrum basiert auf Empirismus, was bedeutet, dass das umzusetzende Team aus seinen eigenen Erfahrungen im Projekt fortlaufend lernt (vgl. *Schwaber/Sutherland, The Scrum Guide, S. 3, 05.02.2017*). Anschließend können notwendige Anpassungen in der Implementierung oder der Art und Weise der Umsetzung vorgenommen werden und mögliche Risiken werden dadurch frühzeitig erkannt. In Scrum wird in diesem Zusammenhang von „Inspektion“ und „Adaption“ gesprochen (vgl. *Schwaber/Sutherland, Software in 30 days, S. 17*). Darüber hinaus bildet bei Scrum, wie in allen anderen agilen Vorgehensmodellen auch, ein Wertesystem das Fundament für das gesamte Framework (vgl. *Dräther et al., Scrum, kurz & gut, S. 31*). Scrum kennt von Grund auf fünf Werte (vgl. *Schwaber/Beedle, Agile Software Development, S. 147 – 154*), die im späteren Verlauf einige Anpassungen erfahren haben, worauf an dieser Stelle allerdings kein Bezug genommen wird, da dies für das Verständnis von Scrum nicht notwendig ist:

1. Selbstverpflichtung (Commitment) – *Das Team baut von selbst und freiwillig das Wollen zur ständigen Weiterentwicklung und Verbesserung auf* (vgl. *Schwaber/Beedle, Agile Software Development, S. 148*).
2. Fokus (Focus) – *Jeder konzentriert sich auf seine Aufgabe und muss sich um nichts Weiteres sorgen* (vgl. *Schwaber/Beedle, Agile Software Development, S. 149*).
3. Offenheit (Openness) – *Jeder ist durch die Transparenz täglich über alle Entwicklungen informiert und kann im Falle einer Fehlentwicklung sofort handeln* (vgl. *Schwaber/Beedle, Agile Software Development, S. 151*).
4. Respekt (Respect) – *Mit Einbezug verschiedener Persönlichkeiten sowie Stärken und Schwächen ist gegenseitiger Respekt sehr wichtig* (vgl. *Schwaber/Beedle, Agile Software Development, S. 152*).
5. Mut (Courage) – *Das Vertreten der eigenen Meinung und die Verpflichtung zu einem Ziel sind essentiell für Scrum* (vgl. *Schwaber/Beedle, Agile Software Development, S. 153*).

2. Das Scrum-Vorgehen

Das Vorgehensmodell Scrum besteht, laut Scrum Guide, aus den folgenden Bausteinen, die im Weiteren näher erläutert werden: drei Rollen, vier Meetings, drei Artefakten und der Definition of Done (vgl. *Dräther et al., Scrum, kurz & gut, S. 44*).

In einem zeitlich und strukturell festgelegten Rahmen, einer Iteration, was bei Scrum „Sprint“ genannt wird, arbeiten die drei Rollen zusammen an der Entwicklung des Produktes (vgl. *Dräther et al., Scrum, kurz & gut, S. 44*). Ein Sprint kann, je nach Projekt, zwischen wenigen Tagen und einem Monat andauern. Länger als ein Monat sollte ein Sprint jedoch nicht sein und ebenfalls ist bei der Sprintlänge zu beachten, dass innerhalb eines Projektes jeder Sprint die gleiche Länge aufweisen sollte (vgl. *Schwaber/Sutherland, Software in 30 days, S. 73*). Des Weiteren sind eine einheitliche Struktur und ein konkretes Ergebnis am Ende eines jeden Sprints notwendig (vgl. *Dräther et al., Scrum, kurz & gut, S. 44*). In jeder Iteration wird

ein fertiges Produktinkrement bzw. ein fertiges und funktionsfähiges Teilergebnis des Projektes, hergestellt (vgl. *Preußig*, *Agiles Projektmanagement*, S. 137).

Aus dem „Product Backlog“ werden vor Beginn des Sprints bestimmte „Backlog Items“ vom „Product Owner“ ausgewählt, die im nächsten Sprint umgesetzt werden sollen. Anschließend schaut sich das „Entwicklungsteam“ die ausgewählten Items an und gibt eine Einschätzung darüber, welche davon im nächsten Sprint umgesetzt werden können. Das Entwicklungsteam entscheidet also selbst darüber, welche der vom Product Owner ausgewählten Items in den nächsten „Sprint Backlog“ aufgenommen werden (vgl. *Dräther et al.*, *Scrum, kurz & gut*, S. 44). Jedes Backlog Item beschreibt eine Anforderung. Im täglichen „Daily Scrum“ werden die Aktivitäten der Teammitglieder auf einen einheitlichen Stand gebracht. Am Sprint-Ende ist das „Produktinkrement“ um die Funktionen erweitert, die durch die umgesetzten Backlog Items gewonnen wurden (vgl. *Dräther et al.*, *Scrum, kurz & gut*, S. 44).

2.1 Rollen

Im Scrum Guide werden drei Rollen beschrieben, die das Scrum-Team bilden: Der „Scrum Master“, der „Product Owner“ und das „Entwicklungsteam“ (vgl. *Schwaber/Sutherland*, *The Scrum Guide*, S. 5, 05.02.2017).

Scrum Master

Der Scrum Master trägt bei Scrum für alles die Verantwortung und ist gleichzeitig der „Herr“ über das Scrum-Vorgehen (vgl. *Dräther et al.*, *Scrum, kurz & gut*, S. 66). Er trägt dafür Sorge, dass das Scrum-Framework stets eingehalten wird, dass alle notwendigen Meetings stattfinden, dass die Timings eingehalten werden und dass dem Scrum-Team alle notwendigen Ressourcen zur Verfügung stehen, die es für die tägliche Arbeit benötigt (Hardware, Lizenzen, Zugriffsrechte, Räumlichkeiten, Kontakt zu externen Experten, etc.) (vgl. *Dräther et al.*, *Scrum, kurz & gut*, S. 66, 67). Weiterhin sorgt der Scrum Master dafür, dass das Team unbehindert arbeiten kann und schützt es vor äußeren unberechtigten Eingriffen (vgl. *Schwaber/Sutherland*, *The Scrum Guide*, S. 7, 05.02.2017). Falls Probleme oder Hindernisse auftauchen, die das Ziel gefährden, beseitigt der Scrum Master diese (vgl. *Rubin*, *Essential Scrum*, S. 227). Der Scrum Master ist der Moderator und Organisator der Scrum Meetings und in diesen auch für die Effektivität und den adäquaten Ablauf der Meetings verantwortlich (vgl. *Dräther et al.*, *Scrum, kurz & gut*, S. 68).

Product Owner

Der Product Owner ist der „Besitzer des Produktes“ und damit auch für das Produkt und für die Wertmaximierung des Produktes verantwortlich (vgl. *Schwaber/Sutherland*, *The Scrum Guide*, S. 5, 05.02.2017). Er ist von dem Produkt überzeugt, hat eine genaue Vision davon und versucht seine Begeisterung auf die anderen Teammitglieder zu übertragen (vgl. *Gloger*, *Wie schätzt man in agilen Projekten*, S. 21). Der Pro-

duct Owner hat fachliche Entscheidungen fundiert und schnell zu treffen, daher muss er die uneingeschränkte Entscheidungsbefugnis haben (vgl. *Dräther et al.*, Scrum, kurz & gut, S. 63). Sofern das notwendige Know-how auf Kundenseite vorhanden ist, kann auch ein Mitarbeiter des Kunden die Rolle des Product Owners besetzen. Da der Product Owner die Kapitalrendite der Produktentwicklung verantwortet, ist sein wichtigstes Ziel, mit jedem Sprint die maximale Erhöhung des Produktgeschäftswertes zu bewirken (vgl. *Rubin*, Essential Scrum, S 206).

Er ist die einzige Person, die Aufgaben an das Entwicklungsteam weitergeben darf und somit auch derjenige, der die Prioritäten festlegt und bestimmt, an welchen Anforderungen gearbeitet wird (vgl. *Schwaber/Sutherland*, The Scrum Guide, S. 5, 05.02.2017).

Weiterhin ist der Product Owner für das gesamte Product Backlog Management verantwortlich. Das bedeutet, dass er für das Entwicklungsteam die Anforderungen im Backlog verständlich formuliert, Akzeptanzkriterien für die Erfüllung aufstellt, das Backlog stets aktuell sowie transparent hält und sicherstellt, dass durch die Arbeit des Entwicklungsteams Geschäftswert geschaffen wird. Ebenfalls ist er derjenige, der nach einem Sprint die Ergebnisse anhand der vorher aufgestellten Akzeptanzkriterien prüft und annimmt (vgl. *Dräther et al.*, Scrum, kurz & gut, S. 64).

Entwicklungsteam

Das Entwicklungsteam, auch Development Team genannt, besteht meist aus drei bis neun Mitgliedern und ist für die Umsetzung der Anforderungen verantwortlich (vgl. *Schwaber/Sutherland*, The Scrum Guide, S. 6, 05.02.2017). Die Teammitglieder werden in Scrum Entwickler (Developer) genannt und sind Spezialisten unterschiedlicher Fachgebiete (vgl. *Dräther et al.*, Scrum, kurz & gut, S. 61, 62). Die Entwicklung erfolgt stets so, dass das Ergebnis nach jedem Sprint Produktinkremente beinhaltet, die ohne Nacharbeiten auslieferungsfähig sind. Weiterhin unterstützt das Entwicklungsteam den Product Owner in seiner Tätigkeit beispielsweise bei fachlichen Themen (vgl. *Dräther et al.*, Scrum, kurz & gut, S. 60). Ebenfalls informiert das Entwicklungsteam den Product Owner ehrlich und frühzeitig über Risiken und Schwierigkeiten während der Umsetzung (vgl. *Dräther et al.*, Scrum, kurz & gut, S. 61).

Das Entwicklungsteam organisiert sich selbst. Die Produktvision steht für das Team immer im Vordergrund und sie erarbeiten eigenständig, wie sie das Sprint-Ziel erreichen (vgl. *Dräther et al.*, Scrum, kurz & gut, S. 61).

2.2 Meetings

Der Scrum Guide benennt vier Meetings als festen Bestandteil im Scrum-Vorgehen. Wichtig dabei ist, dass diese Meetings regelmäßig und idealerweise am gleichen Ort, am gleichen Wochentag und zur gleichen Zeit stattfinden (vgl. *Dräther et al.*, Scrum, kurz & gut, S. 72). Ebenfalls soll die Länge der Meetings, die bei Scrum „Timebox“ genannt wird, innerhalb eines Projektes stets die gleiche sein (vgl. *Schwaber/Sutherland*, The Scrum Guide, S. 7, 05.02.2017). Laut Scrum Guide gibt es die folgenden vier Meetings: das „Sprint Planning“, das „Daily Scrum“, das „Sprint Review“ und die „Retrospektive“.

Sprint Planning

Das Sprint Planning findet stets zu Beginn eines Sprints statt. In diesem Meeting werden gemeinsam im Scrum-Team die Aufgaben für den Sprint besprochen und geplant (vgl. *Dräther et al.*, Scrum, kurz & gut, S. 75). Das Sprint Planning besteht aus zwei Teilen: dem Sprint Planning 1 (hier werden fachliche Anforderungen zur Entwicklung besprochen) (vgl. *Dräther et al.*, Scrum, kurz & gut, S. 76) und dem Sprint Planning 2 (hier wird das „Wie“ besprochen und damit die technischen Voraussetzungen und die Herangehensweise) (vgl. *Dräther et al.*, Scrum, kurz & gut, S. 78).

Daily Scrum

Beim Daily Scrum treffen sich das Entwicklungsteam und der Scrum Master täglich für 15 Minuten, um den aktuellen Stand aller Teammitglieder zu synchronisieren (vgl. *Schwaber/Sutherland*, The Scrum Guide, S. 11, 05.02.2017). Weiterhin findet ein Abgleich mit dem Sprint-Ziel statt und es wird überprüft, ob der aktuelle Stand der Teammitglieder mit der Erreichung des Sprint-Ziels einhergeht (vgl. *Dräther et al.*, Scrum, kurz & gut, S. 81).

Sprint Review

Das Sprint Review kennzeichnet das Ende eines Sprints. Teilnehmer des Meetings können, neben dem Scrum-Team, zukünftige Produktnutzer, Managementvertreter und alle anderen Stakeholder sein, die einen aktuellen Stand über die Produktentwicklung erhalten möchten (vgl. *Dräther et al.*, Scrum, kurz & gut, S. 86). Bei dem Review zeigt das Entwicklungsteam in einer Live-Präsentation, welche neuen Funktionen durch die Umsetzung im Sprint gewonnen wurden und führt diese, anhand der Abnahmekriterien, direkt am Produkt vor (vgl. *Dräther et al.*, Scrum, kurz & gut, S. 86). Der Product Owner prüft anschließend, mit Hinzunahme dieser Kriterien, ob die vorgeführten Funktionen wirklich fertig sind und als „Done“ deklariert werden können. Sollte dies nicht der Fall sein, wäre eine Möglichkeit, dass das Team die Items in das nächste Sprint Backlog mit aufnimmt, um sie dann fertigzustellen (vgl. *Dräther et al.*, Scrum, kurz & gut, S. 87). Die Alternative ist, dass diese Items zurückgestellt werden, um sie zu einem späteren Zeitpunkt

umzusetzen. Dies ist allerdings laut Scrum die schlechtere Wahl, da die nicht fertiggestellte Arbeit keinerlei Geschäftswert erbringt und damit, laut Scrum, eine Verschwendung darstellt (vgl. *Dräther et al.*, Scrum, kurz & gut, S. 87, 88).

Retrospektive

Die Retrospektive ist ein Meeting, das nach dem Sprint Review, aber vor dem nächsten Sprint Planning stattfindet. In diesem Meeting, das grundsätzlich für das Entwicklungsteam und den Scrum Master gedacht ist, geht es darum, die Arbeit und die Zusammenarbeit gemeinsam zu besprechen und Verbesserungspotenziale herauszustellen (vgl. *Dräther et al.*, Scrum, kurz & gut, S. 89). Das Team darf, soweit es dies als notwendig ansieht, jede weitere Person zu der Retrospektive einladen. Besonders bei dem Product Owner, der zu den wichtigsten Kommunikationspartnern des Entwicklungsteams zählt und Teil des Scrum-Teams ist, macht eine Anwesenheit bei der Retrospektive oft Sinn (vgl. *Dräther et al.*, Scrum, kurz & gut, S. 89).

2.3 Artefakte

Bei Scrum gibt es drei Artefakte, die im Scrum Guide als sog. „Scrum-Artefakte“ genannt werden: Product Backlog, Sprint Backlog und (Produkt-) Inkrement (vgl. *Preußig*, Agiles Projektmanagement, S. 137).

Die Scrum-Artefakte dienen dazu, mit wenig Aufwand volle Transparenz über den Projektstand zu geben (vgl. *Munz/Soergel*, Agile Produktentwicklung, S. 86; *Dräther et al.*, Scrum, kurz & gut, S. 96).

Product Backlog

Das Product Backlog beinhaltet alle Produkthanforderungen und enthält somit auch beispielsweise die Backlog Items (vgl. *Dräther et al.*, Scrum, kurz & gut, S. 97). Es ist nicht statisch, sondern wird vom Product Owner fortlaufend angepasst und weiterentwickelt. Er allein ist für das Product Backlog verantwortlich (vgl. *Schwaber/Beedle*, Agile Software Development, S. 34). Im ersten Schritt sind die Backlog Items, solange sie sich noch im Product Backlog befinden, veränderbar und meist auch noch zu verfeinern (vgl. *Dräther et al.*, Scrum, kurz & gut, S. 98). Weiterhin wird der Aufwand für die Umsetzung der Backlog Items vom Entwicklungsteam geschätzt (vgl. *Dräther et al.*, Scrum, kurz & gut, S. 101). Sobald sie hinreichend spezifiziert sind und vom Entwicklungsteam für den nächsten Sprint ausgewählt wurden, sind sie ab Sprintbeginn im Sprint Backlog, welcher, wie auch der gesamte Sprint, vor Änderungen geschützt ist (vgl. *Dräther et al.*, Scrum, kurz & gut, S. 99).

Sprint Backlog

Das Sprint Backlog ist das Artefakt, das alle Backlog Items, die für den aktuellen Sprint ausgewählt wurden, beinhaltet. Ebenfalls ist ein Plan zur Erstellung des Produktinkrements und zur Erreichung des Sprint-Ziels enthalten (vgl. *Dräther et al.*, Scrum, kurz & gut, S. 102). Während das Product Backlog ein Artefakt des

Product Owners ist, gehört das Sprint Backlog allein dem Entwicklungsteam (vgl. *Schwaber/Sutherland, The Scrum Guide*, S. 15, 05.02.2017). Einzig das Entwicklungsteam darf den Sprint Backlog auch während eines Sprints verändern, wenn es sich während der Umsetzung als notwendig herausstellen sollte (vgl. *Dräther et al., Scrum, kurz & gut*, S. 103).

(Produkt-) Inkrement

Das (Produkt-) Inkrement beinhaltet alle bisher fertiggestellten Inkremente des Produktes (vgl. *Dräther et al., Scrum, kurz & gut*, S. 103). Dass das Inkrement am Ende eines jeden Sprints funktionsfähig und einsetzbar an den Product Owner übergeben wird, liegt in der Verantwortung des Entwicklungsteams (vgl. *Dräther et al., Scrum, kurz & gut*, S. 104).

2.4 Definition of Done

Die „Definition of Done“ beschreibt, wann (anhand welcher Kriterien) ein Backlog Item oder ein Inkrement fertig umgesetzt ist. Das Wichtigste dabei ist, dass jedes Teammitglied das gleiche Verständnis davon hat, wann eine Anforderung fertig ist (vgl. *Schwaber/Sutherland, The Scrum Guide*, S. 16, 05.02.2017).

3. Wichtige Stakeholder bei Scrum

Zu den wichtigen Stakeholdern zählen in den meisten Fällen „Manager“, „Kunden“ und „Anwender“.

Manager

Die Manager des umsetzenden Unternehmens sind wichtige Kontaktpersonen für den Scrum Master, wenn es darum geht, dass Hindernisse aus dem Weg geräumt werden müssen. Sie werden für alle Investitionsentscheidungen benötigt (vgl. *Dräther et al., Scrum, kurz & gut*, S. 71).

Kunde

Der Kunde wird bei Scrum durch den Product Owner repräsentiert. Der Product Owner genießt das volle Vertrauen des Kunden und ist im Scrum-Team dessen Sprachrohr. Trotzdem ist der Kunde einer der wichtigsten Stakeholder im Scrum-Projekt, da er der Auftraggeber und damit auch der Geldgeber des Projektes ist (vgl. *Dräther et al., Scrum, kurz & gut*, S. 71).

Anwender

Die Anwender sind die Personen, die konkretes Wissen darüber haben, was durch das neue Produkt erreicht werden soll. Sie sind diejenigen, die anschließend mit dem Produkt arbeiten werden, weshalb sie auch genaue Vorstellungen von dem Produkt haben und konkrete Erwartungen daran setzen (vgl. *Dräther et al., Scrum, kurz & gut*, S. 71).

4. Zusammenfassung

Bei Scrum handelt es sich nicht lediglich um ein Rahmenwerk, welches gewisse Ablaufregeln vorgibt. Die weichen Faktoren (sog. Soft Skills) der beteiligten Personen spielen eine entscheidende Rolle für den erfolgreichen Einsatz dieses Vorgehensmodells. Themen wie Vertrauen, Verbindlichkeit und Transparenz sind unabdingbare Eigenschaften für ein Scrum-Projekt.

Für ein agiles Vorgehensmodell wie das Scrum-Vorgehen gibt es nicht immer nur einen passenden Vertragstyp. Jedes agile Projekt ist unterschiedlich und bereits aus dem Wort Agilität ergibt sich, dass flexibel, anpassungsfähig und beweglich gehandelt wird.

Es liegt nahe, von einem Vertragskonstrukt aus Dienst- und Werkvertrag auszugehen, um damit den Gegebenheiten von Scrum gerecht zu werden.

III. Fazit und Ausblick

Die Rechtsprechung zeigt, dass juristische Fragestellungen, die aus neuen Techniken und Verfahren resultieren, mit den bestehenden Werkzeugen unseres Zivilrechts gelöst werden können. Damit bestätigt sich die Überlegenheit abstrakter Rechtssysteme wie des deutschen Bürgerlichen Gesetzbuchs gegenüber einzelfallorientierten Legislativen.

Gleichwohl ist es unabdingbar, dass die Rechtsanwender sich mit den technischen Grundlagen auseinandersetzen und die Wechselwirkungen zwischen technischen und rechtlichen Entwicklungen nachvollziehen. Der schnelle technische Wandel wird auch in Zukunft für spannende Rechtsfragen in Theorie und Praxis sorgen.