

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
Software and Computational Systems Lab
Juristisches IT-Projektmanagement
Dr. Frank Sarre

Ausarbeitung

Analyse der Herausforderungen in der Gestaltung von IT-Verträgen
in der agilen Softwareentwicklung

Daniela Becker
daniela.becker@campus.lmu.de

Bearbeitungszeitraum: 12.12.2016 bis 22.01.2017 im Wintersemester 2016/2017
Betreuer: Dr. Frank Sarre

1. Einleitung	3
2. Eigenschaften der agilen Softwareentwicklung	3
3. Unterscheidung zu klassischen Projektmethoden	4
4. Herausforderungen bei der Auswahl des Vertragstyps	5
4.1 Relevante Vertragstypen für agile Softwareprojekte	6
4.2. Herausforderungen in der Beschreibung der einzelnen Vertragsbestandteile	7
4.2.1 Projektorganisation und Projektdurchführung	8
4.2.2 Mitwirkung des Auftraggebers	9
4.2.3 Dokumentation	9
4.2.4 Änderung	10
4.2.5 Abnahme	10
4.2.6 Vergütung	11
5. Fazit	12
Literaturverzeichnis	13

1. Einleitung

Durch den in den letzten Jahren vermehrten Einsatz¹ agiler Projektmethoden wie zum Beispiel Scrum kommt die Frage auf, wie diese Art von Softwareprojekten in IT-Verträgen festgehalten werden sollen. In dieser Ausarbeitung soll der Frage nachgegangen werden, welche Herausforderungen es in der Gestaltung von IT-Verträgen in der agilen Softwareentwicklung gibt. Zu Beginn wird in Abschnitt 2 auf die Eigenschaften der agilen Softwareentwicklung eingegangen. Es soll gezeigt werden, was diese Methoden ausmacht und was man sich von deren Einsatz verspricht. In Abschnitt 3 soll zudem gezeigt werden, was agile Projekte von klassischen Softwareprojekten unterscheidet. In Abschnitt 4 soll analysiert werden, welche Herausforderungen es in der Erstellung von Verträgen für agile Projekte gibt. Dabei wird zunächst auf die vertragstypologische Einordnung eingegangen um zu zeigen, welche Vertragstypen generell in Frage kommen.

Bei der Erstellung der einzelnen Vertragsbestandteile von IT-Verträgen sind aufgrund der Besonderheiten agiler Vorgehensweisen einige Dinge zu beachten. Es soll gezeigt werden, wie diese Besonderheiten in Verträgen umgesetzt bzw. beachtet werden können und welche Lösungsvorschläge es für häufig auftretende Fragestellungen gibt.

2. Eigenschaften der agilen Softwareentwicklung

In dem sogenannten „Agile Manifesto“ aus dem Jahr 2001 beschreiben Beck et al.² vier Werte³, die agile Softwareentwicklung ausmachen. Zudem wurden 12 Prinzipien⁴ aufgestellt, die diese Werte genauer beschreiben und Entwicklerteams helfen sollen, agil Software zu entwickeln. Im Gegensatz zu klassischer Softwareentwicklung sollen bei der agilen Softwareentwicklung Individuen und Interaktionen wichtiger sein als Prozesse und Werkzeuge. Um dies zu erreichen, sollen Projektmitglieder wie Projektmanager und Entwickler motiviert und selbst organisiert zusammenarbeiten. Die kontinuierliche Zusammenarbeit von Auftraggeber und Auftragnehmer ist dabei wichtiger als eine initiale

¹ siehe dazu beispielsweise eine Studie der Hochschule Koblenz zum Thema „Status Quo Agile“, Hochschule Koblenz, Status Quo Agile (Studie 2012), URL: <http://www.hs-koblenz.de/rmc/fachbereiche/wirtschaft/forschung-projekte-weiterbildung/forschungsprojekte/status-quo-agile/status-quo-agile-studie-2012/>

² vgl. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., et al. Manifesto for agile software development, 2001. URL: <http://agilemanifesto.org>

³ Die vier Werte: 1. Individuals and interactions over processes and tools, 2. Working software over comprehensive documentation, 3. Customer collaboration over contract negotiation, 4. Responding to change over following a plan

⁴ vgl. Agile Alliance. 12 principles behind the agile manifesto. URL: <https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/>

Aushandlung von Verträgen. Während des Projektes soll die Erstellung funktionierender Software immer über der Erstellung einer Dokumentation stehen. Die funktionierende Software soll zudem immer wieder an den Auftraggeber ausgeliefert werden um ihm die Möglichkeit zu geben, diese einzusehen oder gar fortlaufend zu veröffentlichen. Werden während des Projekts Änderungen nötig, soll darauf reagiert und diese umgesetzt werden, solange sie der Software und damit dem Kunden zugute kommen. Seit der Beschreibung dieses Manifests haben sich viele verschiedene agile Methoden bzw. Vorgehensmodelle entwickelt. Die Methoden basieren auf den Gedanken des agilen Manifests und sind konkrete Vorschläge, wie sich agile Entwicklung in der Praxis umsetzen lässt - Scrum, der Rational Unified Process (RUP), oder eXtreme Programming (XP) sind nur einige Beispiele.⁵

3. Unterscheidung zu klassischen Projektmethoden

Im Gegensatz zu klassischen, oftmals sequentiellen, phasenorientierten Methoden der Softwareentwicklung (z.B. Wasserfallmodell)⁶ wird in der agilen Softwareentwicklung meist iterativ vorgegangen. Das bedeutet unter anderem, dass zu Beginn der Entwicklung meist noch kein Bild des fertigen Produkts besteht, da sich dieses erst durch Erweiterung, Anpassung und Änderungen der Anforderungen im Laufe der Zeit vervollständigt. Das bedeutet zum Beispiel, dass zu Beginn des Projektes kein vollständiges Lasten- oder Pflichtenheft vorliegt, sondern vielmehr eine grobe, meist nicht-technische Beschreibung von Anforderungen. Änderungen an Anforderungen werden auch als „Change Requests“ bezeichnet und sind zu jeder Zeit willkommen und erwünscht, solange das Endprodukt davon profitiert.⁷

Die Einbringung dieser Change Requests in das Projekt erfordert verstärkte Kommunikation von Auftraggeber und Auftragnehmer. Daher zeichnen sich agile Projekte oft durch eine intensive Zusammenarbeit beider Parteien über die gesamte Länge des Projekts aus.⁸ Dies steht beispielsweise im Gegensatz zu einem Vorgehen wie dem Wasserfallmodell, in dem die meiste Zusammenarbeit in den ersten Phasen, der Spezifikation und dem Grob- und Feinentwurf, stattfindet.⁹

⁵ vgl. Hruschka, P., Rupp, C., Starke, G. Agility kompakt. Springer Science & Business Media, 2009, S. 52 ff.

⁶ vgl. Berg, B., Knott, P., Sandhaus, G. Hybride Softwareentwicklung: Das Beste aus klassischen und agilen Methoden in einem Modell vereint. Springer-Verlag, 2014, S. 28 ff.

⁷ vgl. Witte, A. Agiles Programmieren und § 651 BGB. ITRB 2010, 44, 47

⁸ vgl. Kremer, S. Gestaltung von Verträgen für die agile Softwareerstellung. ITRB 2010, 283, 284

⁹ vgl. Hruschka et al. (2009), S. 29 ff.

Wie im zweiten Wert des „Agile Manifesto“ beschrieben, soll die Erstellung funktionierender Software im Vordergrund stehen. Daher wird bei agilen Methoden meist unter anderem aus Gründen der Kosteneinsparung und Aufwandsminimierung auf die Erstellung einer klassischen Dokumentation verzichtet.

Bei der Anwendung iterativer Vorgehensmodelle in der agilen Entwicklung werden ähnliche Schritte wie in phasenorientierten Modellen durchlaufen. Allerdings werden diese immer nach kurzen Zeitabständen wiederholt. Bei dem Modell Scrum soll beispielsweise zum Ende eines solchen Zyklus („Sprint“ genannt) eine lauffähige Softwareversion entstehen und zum Auftraggeber ausgeliefert werden können. Dieser hat daraufhin die Möglichkeit, die Version anzusehen oder zu veröffentlichen.¹⁰

4. Herausforderungen bei der Auswahl des Vertragstyps

Laut Müller¹¹ ist die Zuordnung eines Softwareerstellungsvertrags zu einem bestimmten Vertragstyp (vertragstypologische Einordnung) aus mehreren Gründen von Wichtigkeit. Da Softwareverträge meist mit vorgefertigten Vertragsklauseln (als AGB zu werten) beschlossen werden, kann über „die Vorschriften des dispositiven Gesetzesrechts des [zuvor festgelegten, DB] jeweiligen Vertragstyps“¹² darüber entschieden werden, welche Rechtssprechung anstelle der unwirksamen AGB-Klausel treten soll. Kann kein Vertragstyp bestimmt werden, kann die Klausel durch eine anderweitige Vertragsauslegung ersetzt werden.¹³

Zudem bestimmt der gewählte Vertragstyp, welche Gewährleistungsregelungen im Falle eines Mangels angewendet werden können.¹⁴

In den folgenden Abschnitten soll untersucht werden, welche Vertragstypen für agile Softwareentwicklungsprojekte prinzipiell in Frage kommen ohne dabei die zuvor beschriebenen Eigenschaften agiler Projekte einzuschränken. Zudem soll die Frage beantwortet werden, welche Herausforderungen es in der Gestaltung der einzelnen Bestandteile eines solchen Vertrags gibt und welche Lösungsvorschläge in der Literatur dargelegt werden.

¹⁰ vgl. Kremer, S. ITRB 2010, 283, 284

¹¹ vgl. Müller, M. Vertragsgestaltung bei Agilen Softwareentwicklungsverträgen. HMD Praxis der Wirtschaftsinformatik 53, 2 (2016), S. 215

¹² Thewalt, S. Der Softwareerstellungsvertrag nach der Schuldrechtsreform: Rechtsnatur, Leistungsbestimmung und Mängelhaftung. Rhombos-Verlag, 2004, S. 13

¹³ vgl. Thewalt, S. (2004), S.13

¹⁴ vgl. Müller, M. (2016), S. 215

4.1 Relevante Vertragstypen für agile Softwareprojekte

Als adäquate Vertragstypen für agile Softwareentwicklung werden vor allem der Werkvertrag, Kaufvertrag und Dienstvertrag diskutiert.

Der Werkvertrag zeichnet sich dadurch aus, dass dem Auftraggeber ein mangelfreies Werk bzw. Erfolg geschuldet wird. Erst nach der erfolgreichen Abnahme des fertiggestellten Werks durch den Auftraggeber findet ein Gefahrübergang vom Auftragnehmer zum Auftraggeber statt. Die Vergütung wird erst nach erfolgreicher Abnahme fällig und ist meist ein Festpreis. Nach der Abnahme geht zudem die Beweislast auf den Auftraggeber über. Er muss dann beweisen, dass ein Mangel vorliegt, der Mängelansprüche nach sich ziehen kann.¹⁵

Liegt, wie in der Regel bei agilen Projekten der Fall, kein Pflichtenheft vor, wird beim Werkvertrag laut Koch¹⁶ ein mittlerer Ausführungsstandard geschuldet.

Ein Kaufvertrag könnte angewendet werden, wenn „die im konkreten Fall erforderliche Planungsleistung sich bloß als eine jedem Vertrag über die Lieferung herzustellender Teile immanente Tätigkeit darstellt“¹⁷, also nur einen geringen Teil des gesamten Projekts ausmacht.

Generell lässt sich sagen, dass sich die Planung bei agilen Projekten nicht nur auf eine vorgelagerte Planungsphase wie bei klassischen Vorgehensmodellen beschränkt.

Aufgrund von eingehenden Change Requests, wiederkehrenden Review Meetings oder Erweiterung der Anforderungen während des Projekts erstreckt sich die Planung über den gesamten Projektzeitraum. Müller ebenso wie Witte¹⁸ führen daher an, dass aufgrund der sich durch das ganze Projekt ziehenden Planungsleistung die Planung einen Hauptbestandteil bzw. Schwerpunkt des Vertrags ausmacht. Aus diesem Grund sei es nicht möglich, den Kaufvertrag als Vertragstyp anzuwenden- es sei ein Werkvertrag vorzuziehen.

Der Kaufvertrag kommt zudem aufgrund einer möglichen Sacheigenschaft von Software ins Gespräch.¹⁹ Witte²⁰ führt hierzu allerdings eine Entscheidung des OLG Münchens an, die Software eine Sacheigenschaft abspricht: „Selbst wenn man davon ausgeht, dass es sich bei dem Softwareprogramm um eine bewegliche Sache (Datenträger) handelt,

¹⁵ vgl. Müller, M. (2016), S. 215

¹⁶ Koch, F. Agile Softwareentwicklung - Dokumentation, Qualitätssicherung und Kundenmitwirkung. ITRB 2010, 114, 118

¹⁷ Witte, A. ITRB 2010, 44, 45

¹⁸ vgl. Fuchs, A., Meierhöfer, C., Morsbach, J., Pahlow, L. Agile Programmierung–Neue Herausforderungen für das Softwarevertragsrecht? Multimedia und Recht, 7 (2012), S. 432 und Witte, A. ITRB 2010, 44, 45

¹⁹ vgl. Fuchs, A. et. al. (2012), S. 432

²⁰ Witte, A. ITRB 2010, 44, 45

besteht die eigentliche Leistung in der geistigen Schöpfung des Programms, und nicht in der Lieferung der herzustellenden beweglichen Sache. Darüber hinaus wurde die Individualsoftware per Datenfernübertragung übertragen, so dass auch von einer beweglichen Sache nicht ausgegangen werden kann.“²¹

Im Zuge agiler Softwareprojekte wird ebenfalls der Dienstvertrag diskutiert. Hier liegt allerdings das Risiko für den Erfolg des Projekts bzw. die Projektverantwortung bei dem Auftraggeber. Da der Auftraggeber aber meist die Entwicklung der Software auslagert, weil ihm selbst die Kompetenzen dazu fehlen, ist es fraglich ob er die Projektverantwortung übernehmen möchte. Näher liegt, dass die Verantwortung vom Auftragnehmer übernommen wird, da dieser die nötigen Kompetenzen zur Führung des Projekts haben sollte.²² Lapp²³ schließt in der vertragstypologischen Einordnung den Dienstvertrag direkt aus und führt an, dass der Auftraggeber den Auftragnehmer in agilen Projekten zwar vermehrt unterstützt, die Projektverantwortung aber dennoch beim Auftragnehmer verbleibt. Koch²⁴ bringt dagegen vor, dass ein Dienstvertrag nach dem Erstellen einer ersten Version der Software eingesetzt werden könnte, da die nachfolgende Arbeit vor allem „auf Zuruf“ erfolgt.

Lapp²⁵ und Kremer²⁶ argumentieren, dass bei agilen Softwareprojekte unter bestimmten Voraussetzungen auch ein Gesellschaftsvertrag in Frage kommen könne. Dies wäre dann der Fall, wenn es sich um ein Gemeinschaftsprojekt handelt, in dem beide Parteien die gleiche Verantwortung tragen und zum Beispiel gleiche Rechte am Programmcode haben wollen. Meist sei dies aber nicht gewollt und damit der Gesellschaftsvertrag für das Projekt nicht anwendbar.

4.2. Herausforderungen in der Beschreibung der einzelnen Vertragsbestandteile

In den nachfolgenden Abschnitten soll für verschiedene grundlegende Fragestellungen in einem Softwareprojekt untersucht werden, inwieweit diese für agile Projekte von besonderer Bedeutung sind. Zu den Fragestellungen zählen beispielsweise, wie ein Projekt durchgeführt bzw. organisiert werden soll oder welche Mitwirkungspflichten

²¹ vgl. OLG München, Urteil vom 23. Dezember 2009, Az. 20 U 3515/09

²² vgl. Fuchs, A. et. al. (2012), S. 429

²³ vgl. Lapp, T. Interaktion und Kooperation bei IT-Projekten. ITRB 2010, 69, 70

²⁴ Koch, F. ITRB 2010, 114, 119

²⁵ vgl. Lapp, T. ITRB 2010, 69, 70

²⁶ vgl. Kremer, S. ITRB 2010, 283, 284

Auftraggeber und Auftragnehmer haben. Zudem soll geprüft werden, ob und wie diese Punkte als Bestandteile eines IT-Vertrags für agile Projekte aufgenommen werden sollen.

4.2.1 Projektorganisation und Projektdurchführung

Für eine erfolgreiche Projektorganisation empfiehlt Müller²⁷ das Vorgehensmodell, das im Projekt zum Einsatz kommen soll, im Vertrag zu beschreiben. Alle Projektbeteiligten sollen sich klar darüber sein, welche Aufgaben und Rollen sie im Projekt einnehmen um ein Scheitern des Projekts aufgrund einer fehlenden Organisationsstruktur zu vermeiden. Auch Söbbing²⁸ rät zum Festhalten der Lenkungsform des Projekts im Vertrag. Je nachdem welches Vorgehensmodell gewählt wurde, sollen Rollenbeschreibungen, Meetingbeschreibungen oder Artefakte (für das Beispiel Scrum könnte ein Artefakt z.B. das „Product backlog“ sein) genau erläutert werden.

Wie in Punkt 3 zuvor beschrieben, wird bei agilen Softwareprojekten meist auf die Erstellung eines detaillierten Lasten- und Pflichtenheft verzichtet. Laut Koch²⁹ bedeutet das aber nicht, dass es gar keine Beschreibung der Anforderungen zu Beginn des Projekts geben soll. Vielmehr könnte z.B. beim Vorgehensmodell Scrum zu Projektbeginn ein „Product backlog“ mit einzelnen Anforderungsbeschreibungen („user stories“³⁰) zusammengestellt werden, welches dann als Leistungsbeschreibung angesehen werden kann.³¹

Im weiteren Projektverlauf kann sich dann stückweise an die Zielvorstellungen des Auftraggebers angenähert werden. Lapp³² sieht als Basis eines Erfolgs dieser Vorgehensweise „Gegenseitiges Vertrauen, Respekt und gute Kommunikation“³³ und schlägt vor, dies auch als einen Grundsatz des Vertrags z.B. in der Präambel aufzunehmen.

²⁷ vgl. Müller, M. (2016), S. 218 f.

²⁸ vgl. Söbbing, T. Agile Projekte in der IT-rechtlichen Praxis. ITRB 2014, 214, 215 f.

²⁹ vgl. Koch, F. *ITRB 2010*, 114, 116 f.

³⁰ vgl. z.B. Hruschka et al. (2009), S. 65 ff. für detaillierte Beschreibung von „user stories“ und „Product backlog“

³¹ vgl. Müller, M. (2016), S. 218

³² vgl. Lapp, T. *ITRB 2010*, 69, 69

³³ Lapp, T. *ITRB 2010*, 69, 69

4.2.2 Mitwirkung des Auftraggebers

Im Gegensatz zu klassischen Softwareprojekten kommt dem Auftraggeber bei agilen Projekten eine bedeutendere Rolle zu. Da wie zuvor beschrieben meist kein Lasten- bzw. Pflichtenheft vorliegt, werden die Anforderungen über den Projektverlauf konkretisiert, erweitert oder geändert. Die Planung findet also nicht nur zu Beginn des Projekts statt, sondern erstreckt sich über dessen gesamten Verlauf. Der Auftraggeber ist daher immer wieder dazu aufgefordert, an Anpassungen von Anforderungen mitzuwirken und beispielsweise notwendige Informationen bereitzustellen.³⁴ Im Vorgehensmodell Scrum sollte der Auftraggeber zudem an sogenannten „Review“-Meetings teilnehmen, in denen der vergangene „Sprint“ reflektiert und überprüft wird. Sobald ein neues Inkrement der Software fertig gestellt wurde ist der Auftraggeber außerdem aufgefordert, dieses zu prüfen und zu bestätigen.³⁵

Am Beispiel eines Scrum-Projektes kommt dem Auftraggeber die bedeutendste Rolle zu, wenn er den sogenannten „Product Owner“ stellt. Dieser ist unter anderem für die Definition der Anforderungen und damit für die Ausgestaltung und den Einsatz des Softwareprodukts verantwortlich.³⁶ Laut Müller sollte darauf geachtet werden einen Product Owner „mit den zugehörigen Kompetenzen und Weisungsbefugnissen“ zu wählen, sowie diesen zu bevollmächtigen „seinen Arbeitgeber im Rahmen des Vertrages rechtsgeschäftlich zu vertreten und verbindliche Entscheidungen zu treffen“³⁷.

4.2.3 Dokumentation

In Punkt 3 wurde erläutert, dass bei agilen Softwareprojekten die Erstellung einer Dokumentation zu Gunsten einer funktionierenden Software in den Hintergrund rücken soll. In der Literatur ist dennoch die Meinung verbreitet, dass eine Dokumentation nicht einfach ausgelassen werden sollte. Müller³⁸ argumentiert, dass bei einem Werkvertrag eine Dokumentation notwendig sein dürfte. Er rät, genau wie Schneider³⁹, zu Beginn zu prüfen ob eine Dokumentation nötig ist und die Regelung zur Erstellung oder Auslassung dann auch in den Vertrag aufzunehmen.

³⁴ vgl. Kremer, S. ITRB 2010, 283, 286

³⁵ vgl. Lapp, T. ITRB 2010, 69, 70

³⁶ vgl. Hengstler, A. Gestaltung der Leistungs- und Vertragsbeziehungen bei Scrum Projekten. ITRB 2012, 113, 115

³⁷ Müller, M. (2016), S. 220

³⁸ vgl. Müller, M. (2016), S. 221 f.

³⁹ vgl. Schneider, J. „Neue“ IT-Projektmethoden und „altes“ Vertragsrecht. ITRB 2010, 18, 21

Im Gegensatz zu einer Dokumentation eines klassischen Softwareprojekts geht der Trend in agilen Projekten laut Kremer⁴⁰ zu einer Inline- bzw. Online Dokumentation. Bei der Inline-Dokumentation kann eine Entwicklerdokumentation des Quellcodes über Kommentare im Code automatisch generiert werden. Eine Online-Dokumentation kann vom Auftraggeber zum Beispiel direkt in das Programm eingebaut werden und auf eine Hilfe-Website führen.

Auer⁴¹ rät zudem, Änderungen in den Anforderungen, Projektmeetings und andere Abstimmungsprozesse genau zu dokumentieren, da anhand dieser bei einer Abnahme ein klassischer Soll-Ist-Vergleich durchgeführt werden kann.

4.2.4 Änderung

Im Gegensatz zu klassischen IT-Projekten sind in agilen Projekten Änderungen der Anforderungen erlaubt bzw. gewünscht. Die Projektteilnehmer haben die Möglichkeit, sinnvoll erscheinende Änderungen („Change Requests“) basierend auf den Erkenntnissen der vorangegangenen Entwicklungsschritte in das Projekt einzubringen.⁴² Wie in Punkt 4.2.2. erläutert, ist der Auftraggeber vor allem wegen dieser Anpassungen des Anforderungskatalogs dazu angehalten, fortwährend am Projekt mitzuwirken.

Zur vertraglichen Umsetzungen rät Müller⁴³, dass festzulegen ist, wie Änderungen kommuniziert werden, z.B. per E-Mail. Zudem wäre es sinnvoll eine Frist festzusetzen, innerhalb derer der Auftraggeber der Änderung widersprechen kann. Auch Schneider empfiehlt die Festlegung von „Fristen mit „Redaktionsschluss“ und „Freigaben““⁴⁴.

Generell sollte der Auftragnehmer bei der Erstellung des Vertrags darauf achten, Änderungen mit einzukalkulieren um diese nicht komplett selbst tragen zu müssen.⁴⁵

4.2.5 Abnahme

Die Abnahme eines Softwareprodukts durch den Auftraggeber ist vor allem dann interessant, wenn der Softwarevertrag als Werkvertrag abgeschlossen wurde. Wie schon

⁴⁰ vgl. Kremer, S. ITRB 2010, 283, 287

⁴¹ vgl. Auer-Reinsdorff, A. Feststellung der versprochenen Leistung beim Einsatz agiler Projektmethoden. ITRB 2010, 93, 95

⁴² vgl. Fuchs, A. et. al. (2012), S. 432

⁴³ vgl. Müller, M. (2016), S. 218 f.

⁴⁴ Schneider, J. ITRB 2010, 18, 21

⁴⁵ vgl. Kremer, S. ITRB 2010, 283, 287

in Punkt 4.1. beschrieben, liegt die Beweislast genauso wie die Gefahr des zufälligen Untergang des Werkes bis zur Abnahme beim Auftragnehmer.

Vor allem in agilen Projekten ist es laut Müller⁴⁶ zu empfehlen, die genaue Art und Durchführung der Abnahme vertraglich festzuhalten. Es sei zu prüfen, ob zum Beispiel die bei Scrum stattfindenden „Review“-Meetings bzw. Einsichten in die Softwareinkremente als Teilabnahmen behandelt werden. Als Mittelweg schlägt er vor, Reviews als Teilabnahmen anzusehen, während das letztliche Zusammenspiel der Softwareteile, die Integration und die schlussendliche Performance der Software in einer finalen Gesamtabnahme geprüft werden.

Fuchs et al.⁴⁷ hingegen sehen Reviews nicht als Teilabnahmen, da dies nicht im Interesse der Vertragsparteien läge. Vielmehr würden Auftragnehmer und Auftraggeber Reviews als Sichtprüfungen sehen, in denen sie keine verbindlichen bzw. gewährleistungsrelevanten Aussagen treffen wollen.

Um für die Abnahme einen geeigneten „Prüfungsmaßstab“ zu haben, rät Kremer⁴⁸, wenn vorhanden, eine Dokumentation (z.B. eine Inlinedokumentation) oder einen Projektplan, der im Laufe des Projekts erstellt wurde, heranzuziehen.

4.2.6 Vergütung

Die in Punkt 4.2.4. beschriebenen, erlaubten Änderungen hängen auch damit zusammen, welches Preismodell im Vertrag eingesetzt wurde. Auer⁴⁹ rät dazu, dass egal welches Preismodell letztlich verwendet wird, dessen Kalkulation vertraglich akkurat festgehalten werden soll.

Söbbing schlägt einige für agile Softwareentwicklung geeignete Vergütungsmodelle vor. Zum einen ist dies „die Abrechnung der Dienstleistung nach Aufwand“⁵⁰ (Time & Material), was vor allem bei einem Dienstvertrag angemessen erscheint. Ein weiterer Vorschlag ist die Vergütung nach „Richtwert & Effizienbonus“. Dies bedeutet, dass ein maximaler Preis für das gesamte Projekt als Richtwert genannt wird- wird dieser unterschritten, kann der Auftragnehmer einen Effizienzbonus erhalten. Ein weiterer Vorschlag ist der „agile Festpreis“⁵¹, bei dem ein „indikativer Festpreisrahmen“ bestimmt wird. Innerhalb des

⁴⁶ vgl. Müller, M. (2016), S. 221

⁴⁷ vgl. Fuchs, A. et. al. (2012), S. 430

⁴⁸ vgl. Kremer, S. ITRB 2010, 283, 287

⁴⁹ vgl. Auer-Reinsdorff, A. ITRB 2010, 93, 94

⁵⁰ Söbbing, T. ITRB 2014, 214, 218

⁵¹ zum „agilen Festpreis“ vgl. Opelt, A., Gloger, B., Pfarl, W., und Mittermayr, R. Der agile Festpreis: Leitfaden für wirklich erfolgreiche IT-Projekt-Verträge, 2014

Preisrahmens können dann zum Beispiel Change Requests nach dem „Change for free“ - Prinzip eingefügt werden.⁵² Ein weiteres Vergütungsmodell, das „Time & Material“ sehr ähnlich erscheint, ist die Vergütung nach Story Points. Dies lässt sich beispielsweise am Vorgehensmodell Scrum erklären, bei dem einzelne „User stories“⁵³ erstellt werden. Um einzuschätzen, welchen zeitlichen Aufwand eine solche „User story“ erfordert, werden „Story points“ definiert. Die Story points werden dann mit einem festen Preis definiert. Je nach Menge umgesetzter „User stories“ muss der Auftraggeber dann einen gewissen Preis zahlen.⁵⁴

5. Fazit

Agile Methoden in der Softwareentwicklung stellen Auftraggeber und Auftragnehmer vor neue Herausforderungen in der IT-Vertragsgestaltung, die so in klassischen Softwareverträgen nicht berücksichtigt werden mussten.

Schon die vertragstypologische Einordnung von agilen Softwareprojekten wird in der Literatur stark diskutiert. Generell gibt es hier keine Lösung, die für alle Projekte gewählt werden kann- vielmehr muss je nach Projektziel, nach Projektstruktur und Voraussetzungen ein geeigneter Vertragstyp gewählt werden.

Eine praktisch durchweg in der Literatur vertretene Meinung ist, dass stets auf die jeweiligen Besonderheiten des für das Projekt gewählten Vorgehensmodells im Vertrag eingegangen werden soll. Spezifische Eigenschaften sollen möglichst detailliert beschrieben werden, um diese später bei einem etwaigen irregulären Projektverlauf nachvollziehen zu können.

Um das oftmals fehlende Lasten- und Pflichtenheft und eine meist weniger ausführliche Dokumentation „auszugleichen“, wird dazu geraten, alle Rollen, Artefakte und Meetings festzuhalten. Dieses Vorgehen soll sicherstellen, dass dennoch ein Vergleichsmaßstab für die Gesamtabnahme vorhanden ist und diese dadurch erleichtert wird.

⁵² zu „Change for free“ vgl. Sutherland, J. Agile Contracts: Money for Nothing and Your Change for Free. URL: <https://www.scruminc.com/agile-contracts-money-for-nothing-and/>

⁵³ siehe dazu auch Punkt 4.2.1

⁵⁴ vgl. Söbbing, T. ITRB 2014, 214, 218

Literaturverzeichnis

1. Agile Alliance. 12 principles behind the agile manifesto. URL: <https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/> .
2. Auer-Reinsdorff, A. Feststellung der versprochenen Leistung beim Einsatz agiler Projektmethoden. ITRB 2010, 93.
3. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., et al. Manifesto for agile software development, 2001. URL: <http://agilemanifesto.org> (2001).
4. Berg, B., Knott, P., und Sandhaus, G. Hybride Softwareentwicklung: Das Beste aus klassischen und agilen Methoden in einem Modell vereint. Springer-Verlag, 2014.
5. Fuchs, A., Meierhöfer, C., Morsbach, J., und Pahlow, L. Agile Programmierung–Neue Herausforderungen für das Softwarevertragsrecht? Multimedia und Recht, 7 (2012), 427–433.
6. Hengstler, A. Gestaltung der Leistungs- und Vertragsbeziehungen bei Scrum Projekten. ITRB 2012, 113.
7. Hochschule Koblenz. Status quo agile (studie 2012).
URL: <http://www.hs-koblenz.de/rmc/fachbereiche/wirtschaft/forschung-projekte-weiterbildung/forschungsprojekte/status-quo-agile/status-quo-agile-studie-2012/> .
8. Hruschka, P., Rupp, C., und Starke, G. Agility kompakt. Springer Science & Business Media, 2009.
9. Koch, F. Agile Softwareentwicklung - Dokumentation, Qualitätssicherung und Kundenmitwirkung. ITRB 2010, 114.
10. Kremer, S. Gestaltung von Verträgen für die agile Softwareerstellung. ITRB 2010, 283.
11. Lapp, T. Interaktion und Kooperation bei IT-Projekten. ITRB 2010, 69.
12. Müller, M. Vertragsgestaltung bei Agilen Softwareentwicklungsverträgen. HMD Praxis der Wirtschaftsinformatik 53, 2 (2016), 213–223.
13. Opelt, A., Gloger, B., Pfarl, W., und Mittermayr, R. Der agile Festpreis: Leitfaden für wirklich erfolgreiche IT-Projekt-Verträge. Carl Hanser Verlag GmbH Co KG, 2014.
14. Schneider, J. "Neue" IT-Projektmethoden und "altes" Vertragsrecht. ITRB 2010, 18.
15. Söbbing, T. Agile Projekte in der IT-rechtlichen Praxis. ITRB 2014, 214.
16. Sutherland, J. Agile Contracts: Money for Nothing and Your Change for Free. URL: <https://www.scruminc.com/agile-contracts-money-for-nothing-and/> (2008).

17. Thewalt, S. Der Softwareerstellungsvertrag nach der Schuldrechtsreform: Rechtsnatur, Leistungsbestimmung und Mängelhaftung. Rhombos-Verlag, 2004.
18. Witte, A. Agiles Programmieren und § 651 BGB. ITRB 2010, 44.