

Software and Computational Systems Lab  
Institute of Informatics  
Faculty of Mathematics, Informatics, and Statistics  
Ludwig-Maximilians-Universität München, Germany



Juristisches IT-Projektmanagement  
Wintersemester 2017/18

Dr. Frank Sarre

Notwendiges Feintuning von SCRUM für die  
Vertragsgestaltung eines IT-Projekts

Jan Franz

28.01.2018

## Inhalt

|                        |    |
|------------------------|----|
| Einleitung.....        | 1  |
| SCRUM.....             | 1  |
| Grundidee .....        | 1  |
| Problematik .....      | 2  |
| Rollen .....           | 2  |
| Scrum-Master .....     | 2  |
| Product Owner .....    | 3  |
| Development-Team ..... | 4  |
| Sprints .....          | 5  |
| Sprint Review .....    | 5  |
| Dokumentation.....     | 6  |
| Vergütung .....        | 6  |
| Lösungsansätze .....   | 7  |
| Scrum Light .....      | 7  |
| Scrum Medium .....     | 8  |
| Echtes Scrum .....     | 9  |
| Fazit .....            | 10 |

## Einleitung

Durch ihren vermehrten Einsatz in den letzten Jahren wird häufig die Frage gestellt, wie mit agilen Entwicklungsmethoden in Hinblick auf die vertragliche Gestaltung umzugehen ist. In dieser Ausarbeitung wird dabei auf das Vorgehensmodell „SCRUM“ eingegangen. Insbesondere sollen alle Punkte aufgeführt und erläutert werden, die in der Grundidee von SCRUM offen erscheinen.

Zunächst wird hierbei der Grundgedanke von SCRUM erläutert sowie dessen Unterscheidung von klassischen Modellen hervorgehoben. Im Anschluss daran wird aufgeführt, inwiefern Punkte in SCRUM offen erscheinen und welche Lösungsvorschläge für diese existieren. Im Abschluss werden einige verbreitete Modelle dargestellt und beleuchtet.

## SCRUM

SCRUM ist, wie eingangs bereits erwähnt, ein Vorgehensmodell für die agile Projektgestaltung (Schwaber, 2013). Ursprünglich wurde es für die Softwaretechnik entwickelt und eingesetzt, jedoch findet SCRUM mittlerweile aufgrund seiner Unabhängigkeit von diesem Bereich auch Anwendung in vielen anderen Bereichen.

### Grundidee

Die Grundidee von SCRUM beruht darauf, ein Projekt empirisch, iterativ und inkrementell zu gestalten, da viele Entwicklungsprojekte zu umfangreich seien, als dass sie zu Beginn direkt in einen vollständigen Plan gefasst werden könnten. Dementsprechend ist ein wesentlicher Teil der endgültigen Anforderungen zu Beginn nicht zu klären, sondern nur durch Zwischenergebnisse zu definieren.

Die empirische Grundhaltung von SCRUM wird in die drei folgenden Bereiche aufgeteilt:

- **Transparenz:** Fortschritt und Hindernisse eines Projektes werden regelmäßig und für alle sichtbar festgehalten.
- **Überprüfung:** In regelmäßigen Abständen werden Produktfunktionalitäten geliefert und sowohl das Produkt als auch das Vorgehen beurteilt.
- **Anpassung:** Anforderungen an das Produkt, Pläne und Vorgehen werden nicht ein für alle Mal festgelegt, sondern kontinuierlich detailliert und angepasst. SCRUM reduziert die Komplexität der Aufgabe nicht, strukturiert sie aber in kleinere und weniger komplexe Bestandteile, die Inkremente.

## Problematik

### Rollen

Ein SCRUM-Team besteht aus drei festgelegten Instanzen: Dem Product Owner, SCRUM-Master und dem Development Team.

Während das Development-Team aus vertraglicher Sicht relativ leicht zu bestimmen ist, ergeben sich bei den beiden anderen Rollen Fragen nach der Zuständigkeit.

### Scrum-Master

Der SCRUM-Master ist die Person, die für das generelle Verständnis und die Optimierung des SCRUM-Prozesses verantwortlich ist. Ihre Position befindet sich per Definition zwischen dem Development -Team und dem Product Owner. Sie sind für die erfolgreiche Planung von Sprints und die Koordination zwischen verschiedenen Teams zuständig bzw. verantwortlich.

Während dieses noch extern zu regeln ist, sollte der Scrum-Master auch an den sogenannten Sprint-Kickoffs, daily Stand-Ups und Sprint-Reviews mitwirken und deren erfolgreiche Umsetzung erwirken, um eine nachhaltige Verbesserung des Teams zu erzielen. Insbesondere diese Aufgaben führen dazu, dass diese Rolle vom Auftragnehmer zu besetzen ist.

Ungeklärt dabei bleibt jedoch die Frage, ob der SCRUM-Master auf Teil des Development-Teams sein darf oder als gesonderte Person fungieren muss. Hierzu gibt es auch in dem offiziellen Scrum-Guide keine verbindliche Aussage, weshalb es verschiedene Interpretationen davon gibt.

Für den Punkt, dass der Scrum-Master ein Teil des Teams sein darf, spricht, dass Probleme frühzeitig entdeckt werden. Auch ist er dadurch eher in das Projekt integriert und kann Aufwände und kommende Sprints besser einschätzen und planen.

Dagegen spricht jedoch der potentielle Interessenkonflikt: Sollte der Arbeitsaufwand unvorhergesehen steigen, droht die Funktion des Scrum-Masters unterzugehen, um jene Arbeit zu erledigen. Anstehende Sprint-Planungen würden hierbei vernachlässigt werden.

### Product Owner

Auch in Bezug auf die Rolle des Product Owners fasst der Scrum-Guide keine direkte Aussage darüber, welche der beiden Vertragsseiten diese Rolle zu besetzen hat. Jedoch entspricht es am ehesten den Scrum-Werten, wenn der Auftraggeber diese Position übernimmt (Foegen). Insbesondere der Grundgedanke der „engen Zusammenarbeit“ kommt hier zum Tragen.

Durch die regelmäßige Bereitstellung der Inkremente und Sprint Reviews erhält der Product Owner eine immer besser werdende Vorstellung vom Endprodukt und kann seine Anforderungen an jenes Produkt klarer definieren und darlegen.

Dieses entspricht allerdings der Idealform der Scrum-Anwendung. In der Praxis ergeben sich zwei Schwierigkeiten mit diesem Vorgehen.

Zum einen ist durch die enge Zusammenarbeit der Auftraggeber auch zeitlich sehr an das Projekt gebunden. Es kann vorkommen, dass der Auftraggeber nicht über die erforderlichen Ressourcen – Zeit oder Mitarbeiter – verfügt, um dieser Rolle wirklich gerecht zu werden. Auch umgekehrt ergeben sich daraus für das Team Implikationen, wenn beispielsweise Rückfragen und Probleme nicht rechtzeitig beantwortet werden können.

Zum anderen sieht Scrum vor, dass der Product-Owner die Verantwortung für das Projekt und die Arbeit des Teams übernimmt. Dies ist mit den Vorstellungen der meisten Auftraggeber nicht zu vereinbaren und auch juristisch nicht direkt so zu vertreten (OLG Frankfurt, 2017).

Daher ergibt sich aus der Praxis häufig der Ansatz, die Rolle des Product-Owners zu teilen. So gibt es eine Person beim Auftragnehmer, die für das Team als Product-Owner agiert. Alle Fragen werden an ihn gerichtet, auch ist er für die Arbeit des Teams verantwortlich. Dieser „Product Owner“ agiert wiederum mit dem Auftraggeber.

Nachteilig ist an dieser Herangehensweise ist jedoch, dass der Auftraggeber nun nicht mehr direkt am Werk beteiligt ist, sondern um über einen Mittelsmann agiert. Dieser muss eine Vorstellung des Werkes haben, die mit der des Auftraggebers übereinstimmt. Eine regelmäßige Verständigung zwischen dem Product-Owner und dem Auftraggeber ist also zu gewährleisten.

### Development-Team

Meist ist das Team dem Auftragnehmer direkt unterstellt, sodass sich keine weiteren Implikationen für die Vertragsgestaltung und dieser Hinsicht ergeben.

Undurchsichtiger wird es allerdings, wenn der Auftraggeber dem Auftragnehmer Arbeitskräfte mit zur Verfügung stellt. In diesem Fall hängt es dann deutlich von den vertraglichen Gegebenheiten ab, inwiefern die Hierarchie dieser Mitarbeiter geregelt ist.

In einem Optimalfall sind diese externen Mitarbeiter fest in die Team-Dynamik der einzelnen Development-Teams eingebunden und dem Product-Owner unterstellt.

An dieser Stelle ergibt sich allerdings nun die erste Problematik: Wenn der Product-Owner der Auftraggeber ist, wie wird dann der Fall betrachtet, dass diese Ressourcen an anderer Stelle dringend gebraucht werden? Der Zwischenschritt, den Product-Owner vom Auftragnehmer zu trennen, macht diese Angelegenheit noch schwieriger.

Laut Definition hat der Product-Owner die alleinige Entscheidungsgewalt über das Projekt. Die – ohnehin schon gegen den Scrum-Gedanken verstoßende – Trennung würde hierdurch noch weiter ausgehebelt werden.

Jede mögliche Änderung muss folglich im Vorfeld vertraglich festgehalten werden, um spätere Probleme und Unstimmigkeiten mit dem Vertrag zu vermeiden.

## Sprints

Sprints bezeichnen zeitlich begrenzte Intervalle, in denen eine vorgegebene Menge an Aufgaben erledigt werden soll. Im Bestfall bleibt der sogenannte „Workload“ dabei über die Sprints hinweg konstant. In Scrum bestehen Sprints aus zwei bis maximal vier Wochen.

## Sprint Review

Am Ende eines jeden Sprints werden die Ergebnisse des Sprints zusammengetragen und dem gesamten Team sowie dem Product-Owner vorgestellt und bewertet.

Schwierig wird es hier wieder, wenn eine Trennung des Product-Owners vom Auftraggeber vollzogen wurde. Für diesen Fall ist es unbestimmt, ob der Auftraggeber bei der Sprint-Review anwesend sein muss, obwohl es der direkten und engen Zusammenarbeit natürlich zuträglich ist.

Es ist also durchaus vorstellbar, dass Sprints komplett intern bewertet werden und erst zu einem späteren Zeitpunkt dem Auftraggeber vorgestellt werden. Auch ist dann nicht auszuschließen, dass der Auftraggeber nicht über jeden einzelnen Sprint informiert wird, sondern Zusammenfassungen erhält.

Dies hat zunächst keine offenen Auswirkungen auf das Development-Team, bringt allerdings den Umstand mit, dass der Auftraggeber weiter vom Scrum Prozess ausgeschlossen wird. Nach Scrum-Definition würde ihm in diesem Fall keine Rolle im Scrum-Prozess mehr zufallen.

## Teilabnahmen

Über die eigentliche Funktion der Sprint-Reviews hinaus stellt sich die Frage, ob diese auch als Teilabnahmen gewertet werden bzw. gewertet werden dürfen.

Dies unterliegt der Vertragsgestaltung. Prinzipiell erlaubt und gutheißt es Scrum, dass Teilabnahmen in diesem Rahmen gemacht werden. Jedoch gibt es keine gesetzliche Regelung zu diesem Thema, der Auftraggeber ist auch nicht verpflichtet eine Teilabnahme durchzuführen.

## Dokumentation

Insbesondere bei der Verwendung von agilen Vorgehensmodellen ergibt sich häufig die Frage nach der Dokumentation. Auch Scrum macht keine Anforderungen an die Dokumentation an sich.

Zunächst muss – offensichtlich – eine Grundidee des Projektes existieren und festgehalten werden. Dies geschieht in der Regel sowohl auf Seiten des Auftraggebers als auch auf der Seite des Auftragnehmers, um spätere Missverständnisse erfolgreich aufklären zu können. Diese Dokumentation kann jedoch per Definition nur unvollständig sein, sodass sie im Laufe des Projekts stetig erweitert, korrigiert und verfeinert werden muss (Bathe, 2017).

Scrum definiert für diesen Zweck das sogenannte Backlog. Dieses dient als das wichtigste Dokumentationswerkzeug für den Product-Owner um Anforderungen und Testfälle zu hinterlegen.

Die Projektdokumentation obliegt jedoch einzig und allein dem Development-Team. Dies hat zur Folge, dass entweder Mitglieder des Teams häufig sowohl programmieren als Funktionen dokumentieren sollen, oder dass für die Dokumentation eingestellte Mitarbeiter ebenso an den täglichen Scrum-Treffen teilnehmen müssen.

Eine erfolgreiche Dokumentation sorgt allerdings auch dafür, dass spätere Iterationen eventuell besser behandelt werden können und künftige Projekte davon profitieren können.

## Vergütung

Abhängig davon, welche Vertragsform gewählt wurde, ist auch die Bezahlung verschiedener Natur. Die beiden wesentlichen Unterscheidungen liegen hierbei bei der Wahl zwischen dem Festpreis und einer Bezahlung nach Aufwand.

In Bezug auf dieses Thema definiert der offizielle Scrum Guide keine Richtlinie. Vielmehr ist es so, dass das Vorgehensmodell von Scrum auf beide Arten auslegbar ist.

Ein Sprint besteht immer aus einer festen Zeit und einer festen Anzahl von Mitarbeitern. Somit ist ein Festpreis pro Sprint realisierbar und widerspricht dem Scrum-Gedanken nicht. Einzig die Anzahl der Sprints kann, je nach gewählter Scrum-Form, variieren. Des Weiteren soll an dieser Stelle auf Literatur zum „Agilen Festpreis“ verwiesen werden.



## Lösungsansätze

Trotz der umfangreichen Möglichkeiten, einen Vertrag mit Scrum verschieden zu gestalten, haben sich drei wesentliche Ansätze herauskristallisiert (Pfungsten, 2015). Dies ist zum einen auch bedingt durch direkte und indirekte Abhängigkeiten dieser Abänderungen untereinander.

### Scrum Light

Die schlichteste Methode Scrum einzusetzen, ohne den Vertrag oder Scrum großartig darauf anzupassen, nennt sich „Scrum Light“.

Hierbei wird von dem Auftraggeber ein klassisches Lastenheft erwartet. Der Vertrag wird als Werkvertrag ausgelegt. Dieses Vorgehen bietet sich also nur dann an, wenn wenige bis keine Änderungen an dem Endprodukt zu erwarten sind.

Unabhängig davon benutzt der Auftragnehmer zur Umsetzung des Vertrages Scrum, um die Software zu entwickeln. Dabei übernimmt der Auftragnehmer alle in Scrum vorgesehenen Rollen und stellt selbst Product-Owner, Scrum-Master und Development Team.

Weiterhin wird im Voraus das gesamte Projekt mit sämtlichen enthaltenen Sprints geplant. Es empfiehlt sich an dieser Stelle, mit dem Auftraggeber Teilabnahmen nach jedem Sprint zu vereinbaren, obgleich dies nicht zwingend erforderlich ist.

Änderungswünsche durch den Auftraggeber sind bei diesem Vorgehen als „Change Request“ zu betrachten und somit kommerziell zu vergüten.

Insgesamt profitiert der Auftragnehmer am ehesten von diesem Vorgehen. Dieser reduziert sein Risiko durch die regelmäßige Lieferung an den Kunden und durch dessen Feedback, außerdem ist die Vergütung gesichert. Von außen betrachtet ist dieses Vorgehen kein (erkennbares) Scrum.

### Scrum Medium

Eine weitere Mischform ist das sogenannte „Scrum Medium“. Diese setzt mehr von den Scrum-Prinzipien um, bleibt der Original-Form des Scrums nicht komplett treu.

Ähnlich wie bei Scrum Light werden auch hier die Sprints bis zu einem gewissen Maße durchgeplant und Werkvertrags-gemäß verrechnet.

Allerdings kommt hierbei hinzu, dass ein fester Prozess für Change-Requests definiert wird. Zum einen gibt es die Möglichkeit, ein Kontingent vertraglich zu vereinbaren, dass im Zweifelsfalle dazu gezogen wird. Zum anderen kann der Auftragnehmer gestatten, dass neue Features kostenfrei hinzugezogen werden, sofern ähnlich aufwendige – und noch nicht entwickelte – Funktionen aus dem Projekt gestrichen werden.

Diese beiden Ergänzungen sind nicht konträr und können daher gleichermaßen in das Projekt mit eingebunden werden. Auch sind weitere Sonderformen nicht ausgeschlossen.

Auch hierbei werden alle Scrum-Rollen, auch der Product-Owner, von dem Auftragnehmer gestellt. Jedoch wird hierbei der Scrum-Aspekt nicht vollständig verborgen, vielmehr wird die Abarbeitung der gestellten Anforderungen in enger Zusammenarbeit mit dem Auftraggeber erarbeitet.

Bei dieser Methodik profitieren beide Seiten von der von Scrum anvisierten Agilität. Eine Schwierigkeit liegt jedoch in der Bewertung von Change-Requests. Hier müssen sich beide Seiten über den Aufwand der gewünschten Funktionalität und eventuell über den Aufwand der zu streichenden Funktionalität einig werden. Dies kann insbesondere dann zu Streitigkeiten führen, wenn das vertraglich vereinbarte Kontingent für Change Requests bereits aufgebraucht ist oder nicht ausreichen sollte.

Erwähnenswert ist an dieser Stelle noch eine Entscheidung des OLG Frankfurt. Dieses entschied am 17.08.2017, dass Scrum-Verträge sowohl Werk- als auch Dienstverträge sein können. Das entscheidende sei hierbei, „wie der Vertrag gelebt wird“. Auch sei es möglich, die Arbeit in verschiedene Vertragstypen einzuordnen. Die Planung kann dann als Dienstvertrag, die Realisierung – die eigentliche Programmierung – als eigenständiger Werkvertrag betrachtet werden.

Eine klare und generell gültige vertragliche Einordnung dieser Scrum-Form ist dadurch nicht möglich.

## Echtes Scrum

Bei „echtem Scrum“ akzeptiert der Auftraggeber, dass er die beste Lösung zu Beginn noch nicht kennt oder kennen kann. Er bestellt also komplette Sprints beim Lieferanten auf Basis einer groben Kosteneinschätzung beider Seiten. Die Form der Zusammenarbeit kann in diesem Falle natürlich nur über einen Dienstvertrag zustande kommen, wie auch durch das OLG Frankfurt bestätigt wurde (OLG Frankfurt, 2017).

Da der Auftraggeber damit das volle Risiko trägt, sollte die Rolle des Product Owners hierbei auch durch diesen besetzt werden, im Gegensatz zu den beiden bereits erwähnten Methoden. Die Projektleitung des Auftragnehmers würde dann lediglich die Rolle des Scrum-Masters einnehmen.

Der Auftraggeber steuert die Anforderungen und damit die Entwicklung über den Product Owner. Reichen die bestellten Sprints nicht aus, um das Projektziel (die Vision) zu erreichen, muss der Auftraggeber weitere Sprints bestellen – er trägt das alleinige Risiko. Wird das Ziel in weniger Sprints erreicht, sollte dem Kunden vertraglich die Möglichkeit gegeben werden, das Projekt vorzeitig zu beenden. Dies ist bereits bei der Vertragsgestaltung zu berücksichtigen. Natürlich sollte der Ausstieg mit einer gewissen „Kündigungsfrist“ ausgestattet werden, um sogenannte Stehzeiten und Mehrkosten beim Auftragnehmer zu vermeiden.

Da das komplette Risiko beim Auftraggeber liegt, eignet sich diese Art der Zusammenarbeit ebenfalls nur, wenn bereits ein enges Vertrauensverhältnis zwischen Auftraggeber und Auftragnehmer besteht. Außerdem muss der Auftraggeber in der Lage sein, die Agilität wirklich zu leben: Die beteiligten Fachabteilungen benötigen ausreichend Ressourcen und müssen fest in den Prozess mit eingebunden werden.

## Fazit

Das Feintuning von Scrum gestaltet sich in der Praxis oft unterschiedlich, je nach Vorgehensweise, die gewählt wurde.

Zur richtigen Entscheidung gehören dabei mehrere, teilweise verknüpfte Anforderungen, die je nach Anwendungsbereich auch unterschiedlich zu gewichten sind. Dazu gehören sowohl Entscheidungen des Auftraggebers, als auch des Auftragnehmers.

Die ausschlaggebende Grundlage für die Entscheidung liegt jedoch an dem Willen und der Fähigkeit des Auftraggebers, Scrum zu benutzen und einzubinden. Ist eines von beiden nicht im ausreichenden Maße vorhanden, muss Scrum für das Projekt abgeändert oder eingeschränkt werden.

Auch ist das Vertrauen der beiden Vertragspartner ein wichtiger Punkt, den es zu beachten gilt. Fehlt dieses, so sind die Methodiken von „Scrum Medium“ und „Echtes Scrum“ wohl kaum umzusetzen, da die Gefahr der Eskalation bei Unstimmigkeiten und Streitigkeiten recht hoch ist.

Wichtig ist stets für den Auftragnehmer, dass sich die Zusammensetzung der Scrum-Rollen, insbesondere die des Product-Owners, ändert. Dies erfordert je nach gewählter Methodik unterschiedliche Kompetenzen innerhalb der Unternehmensstruktur des Auftragnehmers.

Auch die Zuständigkeit der Dokumentation ändert sich nach Vorgehensweise: So ist bei Scrum prinzipiell der Product-Owner für das Backlog verantwortlich. Wie allerdings mehrfach erwähnt, ist diese Person nicht fest dem Auftraggeber oder Auftragnehmer zuzuordnen, sodass diese Verantwortung je nach Vorgehen zwischen dem Auftraggeber und dem Auftragnehmer wechselt.

## Quellen

Ken Schwaber & Jeff Sutherland, „The Scrum Guide™“,

<https://www.scrum.org/resources/scrum-guide>, 28.01.2018.

TechDivision GmbH, „Agiles Projektmanagement – Projektentwicklung mit Scrum, Kanban & Co.“, [https://www.techdivision.com/Resources/Persistent/a90c984a454ba0b8478694b83f7a8822514b8fc8/Agiles%20PM\\_Whitepaper0502.pdf](https://www.techdivision.com/Resources/Persistent/a90c984a454ba0b8478694b83f7a8822514b8fc8/Agiles%20PM_Whitepaper0502.pdf), 28.01.2018.

Josef Willkommer, „Agile Projektentwicklung mit Scrum“,

<https://www.techdivision.com/blog/agile-projektentwicklung-mit-scrum.html>, 28.01.2018.

Malte Foegen, „Wie gehen Verträge mit Scrum?“,

<https://www.wibas.com/de/unternehmen/publikationen/artikel/scrum/wie-gehen-vertraege-mit-scrum>, 28.01.2018.

Dr. Jens Bücking, „Agile Softwareentwicklung und ‚Scrum‘: Vertragsrechtliche

Besonderheiten“, <https://www.kanzlei.de/rechtsgebiete/it-recht/agile-softwareentwicklung-und-scrum-vertragsrechtliche-besonderheiten>, 28.01.2018.

Mike Kaufmann, „Der Scrum-Reifegrad und der agile Festpreis“,

<https://writeabout.net/2014/02/18/der-scrum-reifegrad-und-der-agile-festpreis>, 28.01.2018.

Heike Bathe, „Technische Dokumentation im Scrum-Team“, <https://www.dev-insider.de/technische-dokumentation-im-scrum-team-a-582982>, 28.01.2018.

Ann-Cathrin Klose, „Der Product-Owner: Backlog-Boss und Chaos-Bezwinger“,

<https://entwickler.de/online/agile/product-owner-backlog-agile-scrum-579793245.html>, 28.01.2018.

OLG Frankfurt a.M., „Zur Abnahme von agil nach dem SCRUM-Verfahren entwickelter

Software“, <http://www.damm-it-recht.de/olg-frankfurt-a-m-zur-abnahme-von-agil-nach-dem-scrum-verfahren-entwickelter-software>, 28.01.2018.

Malte Foegen, „Der Ultimate Scrum Guide 2.0“, 2. Auflage, wibas, 2014.