

Rechtliche und Haftungsfragen bei RE im agilen Umfeld

Ausgabe 2013 / 4

Erscheinungsart: ca. 4x jährlich in elektronischer Form

In dieser Ausgabe:

- Allgemeine rechtliche Aspekte aus Sicht des Requirements Engineering
- Vertragsbasis und Vertragserfüllungspflicht
- Agile Spezifikation und Festpreis
- Öffentliche Ausschreibungen
- Haftung des Teams
- Standards und Normen
- Wer hat rechtlich mehr von der Spezifikation – Auftraggeber oder Lieferant?

In agilen Vorgehensweisen wird meist sehr locker mit ungenauen Requirements umgegangen mit dem Wissen, dass dies der Grundgedanke der agilen Methoden ist.

Doch wie sieht es in der Praxis mit verschiedenen rechtlichen Aspekten und Haftungsfragen aus, an die die Entwickler oder auch Kunden oft nicht denken? Erst im Schadensfall kommen dann Probleme, die durch schlechtes Requirements-Engineering verursacht wurden, zu Tage. Doch dann ist es zu spät!

Im agilen Manifest steht „Funktionierende Software wird höher gewertet als umfassende Dokumentation“. In jedem agilen Methodenbuch steht, dass es besser ist, mehr zu kommunizieren, als detailliert zu dokumentieren. Im Idealfall eines Projekts kann dies voll unterstützt werden. Das Leben besteht jedoch nicht ausschließlich aus Idealfällen.

Requirements-Engineering kann man in verschiedenen Lebenssituationen wiederfinden, bei denen Risiken und Schadenssituationen auftreten. Die meisten Menschen werden bestrebt sein, Risiken und Schäden gering zu halten. Das ist im täglichen Leben beim Autofahren, Hausbauen oder auch in Beziehungen meist der Fall.

Solange alles gut und in den erwarteten Bahnen verläuft, wird kaum jemand darauf drängen, alles aufzuschreiben, um sich abzusichern. Wenn dann aber die Sache schiefläuft, kommt meist die Aussage: „Hätten wir das doch vorher schriftlich festgehalten oder vereinbart“.

Bei Juristen gibt es folgendes Sprichwort: „Wer schreibt der bleibt und wer red't der geht!“

Dies kommt daher, dass in einem Streitfall – im schlimmsten Fall vor Gericht – alles Niedergeschriebene eine um vieles höhere Beweiskraft hat, als eine mündliche Aussage. Dies ist natürlich ein konträrer Standpunkt zu den agilen Methoden und das ande-

re Ende des Spektrums. In der Praxis kommen solche Fälle auch immer wieder vor, wie auch ein Fallbeispiel eines gescheiterten agilen Projekts konkret zeigen wird.

Ein weiteres rechtliches Problem zeigt sich bei der Vereinbarung von **Festpreisen in Projekten**. Der Kunde möchte verständlicherweise möglichst bald wissen, was sein Vorhaben kosten wird. **Der viel zitierte Schätz-Unschärfetrichter ist dabei ein Faktum und kann zwar ignoriert aber nicht eliminiert werden.**

Damit tritt in praktisch jedem Projekt, für das vor der Fertigstellung ein Festpreis vereinbart wird, das Problem auf, dass der geschätzte Preis und die Leistung nicht zusammen passen und bei größeren Abweichungen beginnen dann die Diskussionen, wer nun (darauf-)zahlt.

Es gibt mittlerweile auch Literatur zum Thema „agiler Festpreis“¹. Darin wird jedoch nicht der Stein der Weisen beschrieben oder gar eine Lösung für den Unschärfetrichter gefunden, sondern meist alter Wein in neuen Schläuchen verkauft – also Methoden, die schon seit Jahrzehnten in Festpreisausschreibungsverfahren verwendet werden, werden in einem modernen, agilen Kontext dargestellt. Das eigentliche Problem wird dabei aber nicht gelöst.

Ein wichtiger Punkt bezüglich Haftung ist auch das Thema **Normen und Standards**. Praktisch alle gängigen Normen im Umfeld der Softwareentwicklung verlangen gewisse Mindestdokumentationen und Spezifikationen für eine ordnungsgemäße Projektabwicklung.

¹ Es gibt in der Praxis nur genau einen wirklichen „agilen Festpreis“. Der ist dann möglich, wenn der Kunde einen festen Preis vereinbart und er das Projekt auf jeden Fall genau mit der Erreichung dieser Kosten beendet wird, egal, was herausgekommen ist. Dies ist jedoch in der Praxis ein sehr unwahrscheinlicher Fall.

Wir haben also die beiden Seiten „viel reden und weniger schreiben“ und „möglichst alles aufschreiben, um abgesichert zu sein“.

Die nachfolgenden Abschnitte beleuchten die genannten Themen noch detaillierter.

Allgemeine rechtliche Aspekte aus Sicht des Requirement-Engineerings

Warum ist das Thema Requirements-Engineering aus rechtlicher Sicht überhaupt relevant?

In der agilen Literatur (z.B. [Cohn, 2010] Seite 40 oder [Wirde-mann, 2011] Seite 65) wird explizit erwähnt, dass Requirements (z.B. User-Stories) kein Vertrag sind.

Dies ist nur bedingt richtig. Wenn nicht anders vereinbart, sind schriftlich formulierte Anforderungen – auch wenn es sich „nur“ um ungenau formulierte User-Stories handelt – ein Bestandteil der Vereinbarung mit dem Kunden, sobald diese in den Backlog eingefügt und als zu realisieren gekennzeichnet werden. Spätestens jedoch dann, wenn die Umsetzung dieser Requirements in einer beauftragten Iteration begonnen hat.

Es ist demzufolge in jedem Fall sinnvoll, schon vor Projektbeginn eine schriftliche Vereinbarung mit dem Kunden zu treffen, welche Arten von Requirements vertraglich bindend sind und welche Requirements nur als Diskussionsgrundlage und nicht bindend angesehen werden.

An sich soll es nicht das Ziel sein, eine Spezifikation ausschließlich unter dem Aspekt zu schreiben, damit dann irgendwann vor Gericht gehen zu müssen. Es sollte jedoch vor allem bei externen Kunden–Lieferanten-Verhältnissen und jedenfalls bei Festpreisprojekten aus Gründen der Risikoabsicherung zumindest eine angemessene Struktur und Detailliertheit vorhanden sein, die das finanzielle und rechtliche Risiko in überschaubaren Grenzen hält.

Der Grad der Spezifikation bzw. Dokumentation ist dabei sehr individuell für die jeweilige Projektsituation und das Projektumfeld zu sehen und kann nicht für alle gleichermaßen definiert werden.

Nun kann man natürlich auch den Standpunkt einnehmen: „Mir wird schon Nichts passieren!“ Die Praxis zeigt jedoch, dass mehr als die Hälfte aller Softwareprojekte (auch im agilen Umfeld) nicht so laufen wie geplant und ein Teil dieser Projekte auch komplett abgebrochen wird.

Die Situation, dass ein anfangs gutes Projekt auch im agilen Umfeld zu einem Problemprojekt und damit möglicherweise zu einem Rechtsstreit zwischen Kunde und Lieferant wird, darf daher nicht vernachlässigt werden.

Es gibt natürlich auch viele „interne“ Situationen, bei denen das Entwicklungsteam und der Kunde im selben Unternehmen sind. Hier ist die Lage rechtlich meist nicht so kritisch. Und die Beteiligten, die in diesem Fall ja Arbeitskollegen sind, werden sich typischerweise nicht gegenseitig verklagen, wenn Requirements

nicht korrekt umgesetzt wurden, sondern eine interne Lösung suchen.

Für gewöhnlich ist es bei erhöhten Aufwänden oder Schäden hier auch so, dass der Arbeitgeber den (finanziellen) Schaden zu tragen hat und die handelnden Personen als Arbeitnehmer rechtlich weitgehend frei von Haftung sind.

Es kann jedoch auch hier eine rechtliche Relevanz geben, wenn es durch die beteiligten Personen zu fahrlässigen Handlungsweisen kommt und damit ev. eine persönliche Haftung Einzelner im Schadensfall zum Tragen kommt. Fahrlässigkeit kann hier z.B. auch dadurch vorliegen, dass durch unterlassene Spezifikation oder Analysen wichtige und risikoreiche Punkte übersehen wurden, die zu einem Schaden führen.

Vertragsbasis und Vertragserfüllungspflicht²

In den meisten extern beauftragten Projekten³ wird zwischen Kunde und Lieferant (zumindest grob) vereinbart, was bis wann zu liefern ist. Damit entsteht für den Lieferanten eine rechtliche Pflicht, diese Vereinbarung auch zu erfüllen. Je unklarer nun ein Auftrag bzw. die zugrundeliegende Spezifikation ist, desto unklarer ist natürlich auch die rechtliche Vertragsbasis.

Beispiel einer unklaren Vertragsbasis:

Der Auftraggeber könnte z.B. als sehr unklare Ausgangsbasis im Extremfall formulieren: „Es ist ein neues Zeiterfassungssystem zu liefern.“

Hier wird jedem einleuchten, dass dies zu unklar formuliert ist. Aus Sicht der Vertragserfüllung ist dies natürlich sehr angenehm für den Lieferanten, weil dieser seine Verpflichtungen bereits damit erfüllt hat, wenn er *irgendein* neues Zeiterfassungssystem liefert, das grundsätzlich das macht, was ein Zeiterfassungssystem marktüblich kann. Der Kunde hat aufgrund der unklaren Anforderung keinen Anspruch darauf, ein bestimmtes System zu bekommen oder gar einzelne Funktionen, die ihm nicht gefallen, zu reklamieren.

Die Behauptung, dass durch die enge Kundenbindung der Vertrag bzw. das zugrundeliegende Projektziel trotz anfänglich unklarer Spezifikation effizienter und schneller erreicht werden kann, steht bei agilem Vorgehen im Raum.

Faktum ist, dass wir in einer Welt der begrenzten Ressourcen (Personen, Zeit, Geld) leben und dies auch für die agilen Vorgehensweisen gilt. Wenn also eine Spezifikation anfangs

² Schließen zwei Parteien einen Vertrag, dann verpflichten sie sich gegenseitig den Vertrag zu erfüllen. Durch die Vertragserfüllung erlischt das Schuldverhältnis gem. ABGB (Allgemeines bürgerliches Gesetzbuch). Dies bewirkt, dass beide Teile nicht mehr durch den Vertrag miteinander verbunden sind. Auf Vertragserfüllung kann auch geklagt werden. Auch wenn das Vertragsverhältnis erfüllt worden ist, sind die Partner nicht rechtlos. Es knüpfen sich dann sogenannte Sekundärrechte an, beispielsweise die Gewährleistung.

³ Ausnahme von dieser Regel sind z.B. beauftragte Forschungsprojekte.

Vollständiger Knowledge Letter Zugang

Wir freuen uns, dass Sie an diesem Thema Interesse haben und den Knowledge Letter von Software Quality Lab bis hierher gelesen haben.



Dieser Knowledge Letter ist eine Vorschau (gekürzte Version des gesamten Artikels).

Wenn Sie den ungekürzten Knowledge Letter lesen möchten, registrieren Sie sich bitte unter <http://www.software-quality-lab.com/download/knowledge-letter/anfrage-knowledge-letter/>

Sie erhalten nach der Registrierung vollen Zugang zu allen bisherigen Knowledge Letters von Software Quality Lab und erhalten automatisch künftige Knowledge Letter per E-Mail.

Software Quality Days — Die größte Konferenz zum Thema „Software Qualität“ in Europa!



Besuchen Sie die Top-Konferenz mit allen Infos rund um Software Qualität.

Beste Qualität der Vorträge und Tutorials sowie eine Mischung aus praktischen und wissenschaftlichen Beiträgen machen die Software Quality Days zum Top-Event.

In den 3 praktischen Tracks werden anwendungsorientierte Vorträge präsentiert. Der wissenschaftliche Track zeigt Beiträge mit hohem Innovationsgrad und praktischer Anwendbarkeit, basierend auf Forschungsergebnissen. Im Solution Provider Forum präsentieren Aussteller ihre neuesten Tools mit Praxis-Beispielen.

Nähere Infos unter

www.software-quality-days.com

