

Juristisches IT-Projektmanagement

Wintersemester 2017/18

Dr. Frank Sarre

Dokumentationen in agilen IT-Projekten

Maximilian Frainzl

21. Januar 2018

Inhaltsverzeichnis

Einleitung	3
Grundsätzliches zur Dokumentation.....	4
<i>Definition im Kontext des juristischen IT-Projektmanagements</i>	<i>4</i>
<i>Dokumentation als wesentlicher Bestandteil von Software</i>	<i>4</i>
Arten von Dokumentationen in agilen IT-Projekten	5
<i>Anforderungsdokumentation.....</i>	<i>5</i>
Anforderungsdokumentation am Beispiel „Scrum“	5
<i>Quellcode-Dokumentation.....</i>	<i>6</i>
<i>Anwenderdokumentation („Benutzerhandbuch“)</i>	<i>7</i>
<i>Systemdokumentation und Installationsanweisung</i>	<i>7</i>
Notwendigkeit von Dokumentation	9
Beschreibung der Dokumentation im Vertrag	11
<i>Erstellung einer Leistungsbeschreibung.....</i>	<i>11</i>
Fälligkeit der Dokumentation in agilen IT-Projekten	13
Zusammenfassung	14
Literaturverzeichnis.....	15

Einleitung

Im Gegensatz zur klassischen Projektabwicklung, zum Beispiel nach dem Wasserfallmodell, liegen die Vorteile bei einer agilen Vorgehensweise, beispielsweise nach „Scrum“, unter anderem in kürzeren Auslieferungszeiten und weniger Overhead.

Häufig vernachlässigt wird bei der Wahl eines agilen Ansatzes jedoch das Thema „Dokumentation“. Begründet wird dies häufig durch folgenden Satz aus dem agilen Manifest: „Funktionierende Software ist höher zu bewerten als umfassende Dokumentation.“ (Beck et al.)

Dies bedeutet jedoch nicht, dass Dokumentation komplett vernachlässigt werden darf. So ist im agilen Manifest ebenfalls folgende Passage zu finden: „Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden, schätzen wir die Werte auf der linken Seite höher ein“. (Beck et al.) Eine umfassende Dokumentation wird also auch von den Verfassern des agilen Manifests als wichtig erachtet. (Hoppen)

Auch - aber nicht ausschließlich - die Fehlinterpretation des agilen Manifests dient in IT-Projekten häufig als Begründung dafür, „Dokumentationen, klare vertragliche Vereinbarungen und Planungen als unwichtig darzustellen“. (Liesegang)

Die vorliegende Arbeit gibt im folgenden Kapitel eine Definition zur Dokumentation im Kontext des juristischen IT-Projektmanagements. Außerdem wird aufgezeigt, in welchen Fällen Dokumentationen ein wesentlicher Bestandteil von Software sind.

Anschließend wird ein Überblick über Arten von Dokumentation in agilen IT-Projekten gegeben und im Kapitel „Notwendigkeit von Dokumentation“ aufgezeigt, wieso durch die Einführung eines agilen Vorgehens keinesfalls jegliche Dokumentation vernachlässigt werden sollte.

Im darauffolgenden Kapitel soll ein Einblick in die Vertragsgestaltung hinsichtlich Dokumentationen gegeben werden. Im vorletzten Kapitel geht es um den Zeitpunkt der Fälligkeit von Dokumentationen in agilen IT-Projekten.

Im letzten Abschnitt werden die Ergebnisse dieser Arbeit noch einmal zusammengefasst.

Grundsätzliches zur Dokumentation

Definition im Kontext des juristischen IT-Projektmanagements

Im Internet lässt sich keine klare Definition für den Begriff „IT-Dokumentation“ finden. Weder im Duden, noch in Wikipedia oder anderen Lexika findet man eine eindeutige Beschreibung. (Reiss)

Im Allgemeinen verkörpert „eine Dokumentation [...] Wissen dauerhaft in verschiedenen Erscheinungsformen, beispielsweise textuell in natürlicher Sprache, in Diagrammen, in Bildern, in Audio- und Videoaufzeichnungen. Eine Dokumentation eröffnet weiterhin die Möglichkeit, „Wissen an eine Vielzahl von Personen zu kommunizieren.“ (Liesegang) Es stellt sich dabei natürlich die Frage, ob ein direkter Wissensaustausch nicht praktikabler ist. Anzumerken ist dabei jedoch, dass dieser nicht immer möglich ist, beispielsweise wenn ein IT-Projekt bereits beendet wurde. Am wichtigsten ist Liesegang zufolge, dass die Dokumentation nicht aus formalen Gründen erstellt wird, sondern den Zweck hat „das notwendige Wissen adressatengerecht [zu] präsentieren.“ (Liesegang)

Im „Praxisbuch IT-Dokumentation“ von Manuela und Georg Reiss wird beschrieben, dass es „keinen allgemeingültigen Standard oder gar ein Gesetz für IT-Dokumentation gibt“. Es gibt lediglich einige unternehmensbezogene, abgeleitete und fachspezifische Gesetze. (Reiss)

Dokumentation als wesentlicher Bestandteil von Software

Zunächst einmal ist zu erwähnen, dass Software nach BGH-Rechtssprechung „aus zwei wesentlichen Bestandteilen [...], die praktisch und rechtlich gleichrangig sind, nämlich dem Programm und den Handbüchern (der Dokumentation)“ besteht. (Schneider)

Diese Rechtssprechung gilt besonders dann, wenn die Vertragspartner nichts explizit vereinbart haben. (Schneider 178)

Jedoch werden hier lediglich Handbücher (im Folgenden auch als „Anwendungsdokumentation“ bezeichnet) genannt.

Es hängt nach der Fertigstellung des Softwareprodukts vom Auftraggeber ab, ob weitere und falls ja, welche Dokumentation benötigt wird. Er muss die „nötige Dokumentation zur Ausübung des bestimmungsgemäßen Gebrauchs erhalten“, wie Schreiber-Ehle es im Artikel „Dokumentation in Softwareerstellungsverträgen“ formuliert. (Schreiber-Ehle) Da die Dokumentation abhängig vom jeweiligen Gebrauch ist, kann auch deren Umfang unterschiedlich ausfallen.

In einem IT-Projekt sollten neben der Anwenderdokumentation in den meisten Fällen noch weitere Arten von Dokumentationen angefertigt werden. Um welche Dokumentationen es sich dabei handelt, wird im nachfolgenden Kapitel aufgezeigt. Warum dies notwendig ist, wird im Kapitel „Notwendigkeit von Dokumentation“ beschrieben.

Arten von Dokumentationen in agilen IT-Projekten

Im nachfolgenden Kapitel werden einige Arten von Dokumentationen beschrieben, wobei zu erwähnen ist, dass es sich nicht um eine vollständige Liste, sondern lediglich um eine Auswahl an für agile IT-Projekte relevanten Dokumentationen handelt.

Die wichtigsten Arten von Dokumentationen sind die Anforderungsdokumentation als Projektmanagementdokumentation, die Quellcode-Dokumentation, Systemdokumentation und die Anwenderdokumentation, welche häufig auch als „Benutzerhandbuch“ bezeichnet wird.

Anforderungsdokumentation

Zunächst ist zu erwähnen, dass beim Erstellen einer Dokumentation häufig festgestellt wird, dass Anforderungen „unklar, unvollständig oder inkonsistent sind“. Liesegang sieht eine Begründung dafür, dass Dokumentation bei den Projektbeteiligten des Öfteren als unangenehm wahrgenommen wird darin, dass „der Aufwand, der bei der gedanklichen Auseinandersetzung mit diesen Problemen und Lösungen entsteht, [fälschlicherweise] oft einfach der Dokumentation zugeschrieben“ wird. (Liesegang)

In der klassischen Projektabwicklung, zum Beispiel nach dem Wasserfallmodell, wird zur Anforderungsdokumentation ein Lastenheft erstellt. Auch wenn dieser Begriff im agilen Kontext nicht auftaucht, so rät Hoppen vor Beginn der Entwicklungsphase dazu, diese Art der Dokumentation zu erstellen: Ein solches Lastenheft, oder vielmehr eine „Beschreibung dessen, **was** beabsichtigt ist“, ist zwingend notwendig, da die Anforderungsdefinition am Anfang des Projekts steht und damit unabhängig vom Projekt-Vorgehensmodell ist. Das Vorgehensmodell (z.B. ein agiles wie „Scrum“) bestimmt daraufhin lediglich, **wie** die Anforderungen schließlich in Lösungen umgesetzt werden. (Hoppen)

Weiterhin merkt Hoppen an, dass - im Gegensatz zu Softwareentwicklungsprojekten basierend auf klassischer Vorgehensweise - bei agilen Modellen „vordergründig auf Pflichtenhefte verzichtet [wird], tatsächlich entstehen diese [jedoch] implizit und parallel zur laufenden Softwareentwicklung.“ (Hoppen)

Wichtig ist, speziell bei der agilen Softwareentwicklung, den Fokus auf das zu setzen, was man bei der klassischen Anforderungsdokumentation als „Grobkonzept“ bezeichnen würde, denn „es ist zwar gewollt, dass die (Detail-)Spezifikation so spät wie möglich entsteht, aber auch in agilen Projekten ist nicht gewollt, dass keine oder nur eine unzulängliche Anforderungsdokumentation entsteht.“ (Hoppen)

Anforderungsdokumentation am Beispiel „Scrum“

Es gibt in den agilen Verfahren entsprechende Methoden zur Erstellung einer Anforderungsdokumentation. Am Beispiel „Scrum“ soll im Folgenden eine Empfehlung zur Dokumentation in IT-Projekten gegeben werden:

Generell macht „Scrum“ den Entwicklern keinerlei Einschränkungen oder genaue Vorgaben, welche Inhalte in welcher Form dokumentiert werden sollen. Eine elementare Bedingung an die Dokumentation ist jedoch die Anforderungen an das zu entwickelnde System zu definieren. (Pflüger)

„Scrum“ bietet dazu einige Methoden zur Anforderungsdokumentation: In sogenannten „Backlogs“ wird ein Themenspeicher für Anforderungen angelegt, es gibt „Visions“ und

„Goals“ als Anforderungen an das Gesamtsystem, „Epics“ als Beschreibung noch unklarer und sehr grober Anforderungen und „User Stories“ als Beschreibungsmittel für konkrete funktionale Anforderungen. In „Test Cases“ werden bei agiler Vorgehensweise Anforderungsdetails konkretisiert. (Hoppen)

Zwei der in „Scrum“ verwendeten Instrumente werden im Folgenden kurz genauer erläutert:

Backlogs

„Backlog“ wird ein „Themenspeicher für Anforderungen“ genannt. (Hoppen)

Ein „Backlog“ „ist das Steuerungsinstrument des Product-Owners in Scrum [und] gleichermaßen bei der Definition der Anforderungen als auch beim Tracking des Fortschritts von zentraler Bedeutung.“ (Wiegand)

Dieses meist tabellarische Dokument sollte mindestens die Spalten „Priorisierung“, „User-Story mit Akzeptanzkriterien“ und „Grobabschätzung“ enthalten. (Wiegand)

User Stories

Sogenannte „User Stories“ sind ein beliebtes Verfahren, um bei einer agilen Vorgehensweise „kurz und knapp formulierte funktionale Anforderungen, die später vor allem in der agilen Softwareentwicklung genutzt werden“, festzulegen. (Hoppen)

Ein großer Vorteil bei der Verwendung von „User Stories“ ist, dass sie sich sehr einfach von den Stakeholdern formulieren lassen. Meist werden dazu in den Projekten sehr einfache Satzmuster (Templates) vorgegeben, etwa: „Als *Mitarbeiter* möchte ich *neue Kontakte speichern können*, so dass *es mich möglichst wenig Zeit kostet*.“ (Hoppen)

Zu beachten ist jedoch, dass der Umfang und die Detailstufe für die Dokumentation in „Scrum“ durch die Entwickler selbst gewählt werden kann. Pflüger empfiehlt den Einsatz von Use Case Diagrammen als Hilfsmittel. Er weist außerdem darauf hin, dass es wichtig ist, alle Fachbegriffe zu definieren, um einen etwaigen Interpretationsspielraum bei den Projektbeteiligten einzuschränken. (Pflüger)

Die Vorteile einer ausführlichen Anforderungsdokumentation liegen sowohl in klassischen sowie insbesondere in agilen Projekten darin, dass sich Projektbeteiligte stärker gedanklich mit der „Klarheit und Vollständigkeit der Anforderungen“ befassen. Hoppen stellt sehr klar dar, dass diese Art der Dokumentation noch aus einem anderen Grund sehr nützlich ist:

„Außerdem eröffnet nur eine schriftliche Unterlage die Möglichkeit einer zeitlich entkoppelten Kommunikation mit den Stakeholdern, die auch bei agilem Projektvorgehen in größeren Projektumgebungen fast immer erforderlich ist.“ (Hoppen)

Zusammenfassend stellt die Anforderungsdokumentation eine wichtige Unterlage dar, die „als Grundlage für weitergehende Planungsprozesse auch in agilen Projekten erforderlich“ ist. (Hoppen)

Quellcode-Dokumentation

Quellcode wird für gewöhnlich innerhalb des Codes dokumentiert. Das Kommentieren des Codes wird auch als „Inline-Dokumentation“ bezeichnet. Durch den Einsatz von verschiedenen Tools und der Einhaltung von Kommentierungsstandards, ist es möglich, automatisch eine detaillierte und gute verständliche Quellcode-Dokumentation zu erzeugen. Voraussetzung für die automatische Generierung ist jedoch selbstverständlich, dass diese Inline-Dokumentation vorher angelegt und Kommentierungsstandards eingehalten wurden.

Hoppen merkt dazu an, dass Inline-Kommentierungen nicht selten „in den Quellcode eingesetzt [werden, diese] aber inhaltlich trivial sind und keinen Mehrwert bieten. Es ist daher sinnvoll, in der Leistungsbeschreibung qualifizierte Anforderungen an die Kommentierung des Quellcodes zu stellen.“ (Hoppen) Bei der Erzeugung der Dokumentation kann auf Analysetools zurückgegriffen werden, die den Kommentierungsgrad eines Quellcodes messen können. (Hoppen)

Dass eine agile Vorgehensweise kein Grund ist, auf Quellcode-Dokumentation zu verzichten, beschreibt auch Stiernerling: „Auch in einem agilen Projekt lässt sich ein gewisses Niveau von Quellcode-Dokumentation praktizieren, ohne die Vorteile der agilen Methodik zu verlieren.“ (Stiernerling)

Weiterhin führt er als Beispiel an, dass „das Vertrauensverhältnis zwischen den beiden Vertragspartnern im Verlauf des Projekts zerbrechen“ könnte. In diesem Fall steht der Auftraggeber, insbesondere in agilen Projekten, häufig vor dem Problem, dass er eine undokumentierte, noch nicht abgeschlossene Entwicklung an einen neuen Auftraggeber übergeben muss. Eine Möglichkeit wäre, dass für diesen Fall vorher vertraglich eine zeitlich begrenzte Unterstützung bei der Übergabe an einen neuen Entwickler vereinbart wird. Einfacher und für den Auftragnehmer günstiger ist es jedoch in den meisten Fällen, wenn vorher vereinbart wird, dass der Quellcode und die Schnittstellen zu Partnersystemen laufend dokumentiert werden. Dies erleichtert es einem anderen Auftraggeber die Entwicklung fortzusetzen. (Stiernerling)

Anwenderdokumentation („Benutzerhandbuch“)

Eine weitere, bereits erwähnte, Art der Dokumentation in IT-Projekten ist die Anwenderdokumentation. Witzel definiert diese als „eine gedruckte – oder zumindest ausdrückbare – Unterlage, [...] die den Einstieg in die (angepasste) Software ermöglicht und Konzepte und Verfahren beinhaltet.“ (Witzel 674)

Wie bereits erwähnt ist diese Anwenderdokumentation nach gefestigter Rechtsprechung auch ohne besondere Vereinbarung geschuldet. (Hoppen)

Hoppen empfiehlt, dass der Auftraggeber genau festlegt, welche Form der Anwenderdokumentation er vom Auftragnehmer erwartet.

Es gibt hier keine weiteren Besonderheiten bei einer agilen Projektdurchführung, weshalb an dieser Stelle nicht genauer auf die Anwenderdokumentation eingegangen wird.

Systemdokumentation und Installationsanweisung

Bei einer Systemdokumentation und Installationsanweisung handelt es sich um eine Dokumentation, in der die „bei der Installation vorgenommene Parametrisierung und alle für den Betrieb der Software erforderlichen Informationen dokumentiert sind.“ (Hoppen)

Bei größeren Anwendungen oder dem Betrieb in einem Rechenzentrum kann es auch Angaben für das Betriebskonzept sowie Anweisungen für ein Verhalten in besonderen Fällen (zum Beispiel zu treffende Vorsichtsmaßnahmen und ein Notfall-Konzept) enthalten. (Hoppen)

Laut Hoppen ist dieses „unstrittig und eigentlich in jedem Fall zu liefern“. (Hoppen) Eine gesetzliche Verpflichtung gibt es erst einmal nicht, wie im Kapitel „Beschreibung der

Dokumentation im Vertrag“ erläutert wird. Jedoch ist sie nach Schneider zum Beispiel dann unbedingt zu liefern, wenn vereinbart wurde, „dass der Auftraggeber die Software selbst pflegen will“. (Schneider 270)

Notwendigkeit von Dokumentation

Warum Dokumentation, bei der klassischen genauso wie bei einer agilen Vorgehensweise, wichtig und in den meisten Fällen notwendig ist, wird im Folgenden erläutert.

Eine Anforderungsdokumentation unterstützt insbesondere dabei, dass „[d]as frühzeitige Definieren, Dokumentieren und Praktizieren von Tests und Akzeptanztests [...] u.a. bei agilen Methoden regelmäßig betrieben [wird] und ebenfalls zu einer frühzeitigen intensiven geistigen Auseinandersetzung mit Anforderungen an die Software, Problemen und Lösungen [zwingt].“ (Liesegang)

Liesegang betont außerdem, dass der durch die Erstellung einer Anforderungsdokumentation entstehende Aufwand durch die Entwicklung der Tests selbst entsteht und nicht, wie fälschlicherweise häufig angenommen, durch deren Dokumentation. (Liesegang)

Hoppen stellt fest, dass keine ordnungsgemäße Projekt-Durchführung gesichert ist, wenn der Auftragnehmer sich einzig und allein auf „Scrum“ beziehungsweise die Verwendung agiler Entwicklungs-Tools alleine beruft. (Hoppen)

Wird beispielsweise auf eine Anforderungsdokumentation zu Beginn der Entwicklung verzichtet, kann dies zu weiteren Problemen führen. Insbesondere bei der Weiterentwicklung und Wartung des Produkts wird oft festgestellt, dass jegliche Basis dazu fehlt. (Bednarczyk)

Wie bereits erwähnt, werden Anforderungen bei der Anwendung von „Scrum“ häufig in Form von „User Stories“ dokumentiert. Bednarczyk weist darauf hin, dass diese funktionalen Anforderungen in der Theorie auf Kärtchen geschrieben werden und nach erfolgter Implementierung entsorgt werden. (Bednarczyk)

Auch Backlogs reichen in agilen Projekten nicht aus, da es sich dabei manchmal nur um „ein Sammelsurium inkonsistenter und unvollständiger Anforderungen“ handelt. (Hoppen)
Es sollte also noch eine zusätzliche Anforderungsdokumentation, ähnlich der bei einer klassischen Projektdurchführung angelegten, erzeugt werden.

Der Auftragnehmer behauptet insbesondere bei agiler Softwareentwicklung gerne, dass keine Dokumentation entstehen muss, weil der Quellcode selbst als Primärunterlage jederzeit zur Analyse und Weiterentwicklung herangezogen werden kann. (Hoppen)

Eine reine Quellcode-Dokumentation kann sich, wie bereits im Kapitel „Quellcode-Dokumentation“ beschrieben, jedoch häufig als Problem herausstellen, insbesondere dann, wenn eine Weiterentwicklung durch einen anderen Softwarelieferanten erfolgen soll.

Ein weiteres Problem kann das Fehlen einer Dokumentation über Funktionalitäten und Schnittstellen des Systems nach sich ziehen. Bei einer Weiterentwicklung des Systems muss im schlimmsten Fall ein Reverse-Engineering durchgeführt werden. Denn üblicherweise endet der Produktlebenszyklus nicht mit Abschluss des Entwicklungsprojekts. Aus diesem Grund ist es sehr wahrscheinlich, dass Anpassungen, Pflege und Wartung oder auch Bugfixing noch viel später durchgeführt werden müssen. (Bednarczyk)

Liesegang unterstreicht die Aussage von Bednarczyk mit dem folgenden Satz:

„Die Dokumentation der Softwareentwicklung ist, darunter beispielsweise auch die Dokumentation einer Softwarearchitektur, für den gesamten Lebenszyklus der Software relevant, um eine effektive Wartung und Weiterentwicklung zu ermöglichen.“ (Liesegang)

Als weiteren Grund warum (Anwender-)dokumentation unbedingt erforderlich ist, führt

Bednarczyk an, dass es ohne Dokumentation schwierig ist, „neue Mitarbeiter einzuarbeiten, da diese ohne Dokumentation keinen fachlich fundierten Überblick erhalten, was das System eigentlich tut“. (Bednarczyk)

Zusammenfassend lässt sich sagen, dass die Notwendigkeit der zu erstellenden Software-Herstellungsdokumentationen und Software-Projektdokumentationen unabhängig davon zu sehen ist, ob ein agiles oder beispielsweise phasenorientiertes top-down Vorgehensmodell bei der Softwareentwicklung gewählt wird. (Hoppen).

Beschreibung der Dokumentation im Vertrag

Da es, wie im vorhergehenden Kapitel beschrieben, wichtig ist, nicht auf bestimmte Arten von Dokumentationen zu verzichten, sollten diese unbedingt in den Vertrag oder die Leistungsbeschreibung des Vertrages aufgenommen werden.

Wenn vertraglich nichts anderes vereinbart ist, gehört zu Hardware und Software eine Dokumentation. Laut BGH macht das Handbuch einen wesentlichen Teil der geschuldeten Leistung aus:

„Das vollständige oder teilweise Fehlen der Dokumentation hat zur Folge, dass die Leistung des Softwareanbieters (teilweise) nicht erfüllt ist. Es können durch den Auftragnehmer dann Ansprüche aus (teilweiser) Nichterfüllung einer Hauptleistungspflicht des Vertrages geltend gemacht werden.“ (Witzel 671)

Von der Rechtsprechung aufgestellte Grundsätze sind:

- Zu Software gehört ein Handbuch oder eine Bedienungsanleitung mit einer Perpetuierungsfunktion
- Eine Programmbeschreibung ist grundsätzlich nicht geschuldet.

(Witzel 672)

Bei näherer Betrachtung dieser Grundsätze fällt schnell auf, dass hier nur von einem „Handbuch“ die Rede ist. Wie bereits erwähnt, ist es also zwingend erforderlich in der Leistungsbeschreibung festzuhalten, dass beispielsweise eine Anforderungsdokumentation erstellt werden soll. Diesen Rat gibt auch Müller mit dem Hinweis auf ein wirksames IT Qualitätsmanagement:

„Im Sinne eines wirksamen IT Qualitätsmanagements sollte der Auftraggeber vorab unbedingt evaluieren, ob und in welchem Umfang eine Dokumentation erforderlich ist und gegebenenfalls deren Erstellung ausdrücklich zum Gegenstand des Vertrags machen. Dies gilt insbesondere auch für die Entwicklungsdokumentation, die Voraussetzung für eine Wartbarkeit, Änderbarkeit und Integrierbarkeit der Software ist.“ (Müller)

Üblicherweise wird die Vertragsgemäßheit („Abnahmereife“) dadurch geprüft, dass die Software gegen die Leistungsbeschreibung und im Anschluss die Dokumentation gegen die Software und Leistungsbeschreibung geprüft wird. „Die Erstellung der Dokumentation als expliziter Vertragsgegenstand verstärkt, wenn gewünscht, das werkvertragliche Moment“. (Schneider 204)

Es kann daher äußerst wichtig sein, dass eine Anforderungsdokumentation erzeugt wurde, gegen die geprüft werden kann. Im nachfolgenden Kapitel wird darauf eingegangen, wie man diese als „Leistungsbeschreibung“ im Vertrag festlegen kann.

Erstellung einer Leistungsbeschreibung

Anforderungen an die Dokumentation variieren in der Regel von Projekt zu Projekt und können auch in agilen Projekten sehr vielfältig sein. Die Vertragspartner müssen aus diesem Grund im Vertrag selbst oder in der Leistungsbeschreibung festlegen, welche Dokumentation, in welcher Form und in welchem Umfang erstellt werden soll. Dabei sollte außerdem beachtet werden, dass auch etwaige Anpassungen von Standardsoftware für die Dokumentation berücksichtigt wird. (Witzel 674) Neben Witzel rät auch Hoppen dazu, dass „die

Vertragspartner [...] in einer Leistungsbeschreibung eindeutig definieren, „welche Dokumente als Teil der geschuldeten Leistung gefordert werden“. (Hoppen)

Bei der Erstellung dieser Leistungsbeschreibung ist jedoch zu beachten, dass „Begrifflichkeiten nicht eindeutig besetzt sein müssen und die Vertragspartner sich in Folge dessen besser Gedanken darüber machen [sollten], was an Leistung des Softwareanbieters erwartet wird.“ (Witzel 674)

Witzel gibt hier einen Formulierungsvorschlag:

Der Auftragnehmer erstellt die für die Durchführung des Projekts „...“ (einschließlich Parametereinstellung, Anpassung und Erweiterung sowie Zusätze) geeigneten Dokumentationen in deutscher und englischer Sprache:

- *Projekt- und Ergebnisdokumentation („Projekthandbuch“) einschließlich Dokumentation der Befundsicherung, Dokumentation des Zustandes eines Gegenstandes der Leistung zum Zeitpunkt der Übergabe oder Produktivsetzung („Beweissicherung mit Datumsangaben und Versionsnummern“);*
- *Anwendungsdokumentation (inklusive Online-Hilfe) und Systemadministrationsdokumentation;*
- *Programmbeschreibungen, einschließlich Datenmodell;*
- *Aufstellung der Entwicklungswerkzeuge.*

Die Ausgestaltung der Dokumentation im Detail wird zwischen den Vertragspartnern abgestimmt und schriftlich festgehalten.

(Witzel 675)

Fälligkeit der Dokumentation in agilen IT-Projekten

Witzel gibt zum Zeitpunkt der Fälligkeit von Dokumentation folgenden Hinweis:

„Bei komplexeren Projekten und insbesondere bei solchen, in denen der Anwender häufig Änderungswünsche geäußert hatte, die dann auch realisiert wurden, wird man vom Softwareanbieter nicht zeitgleich mit der Fertigstellung auch die Fertigstellung der Dokumentation verlangen können.“ (Witzel 673).

Dies ist insbesondere bei agilen Projekten häufig der Fall. So werden bei der Anwendung von „Scrum“ in einem iterativen Prozess Verbesserungen schrittweise durchgeführt.

Wichtig ist hier jedoch wieder, dass Dokumentation auch während eines „Sprints“ nicht komplett vernachlässigt wird.

Es gibt hinsichtlich des Zeitpunkts drei verschiedene Möglichkeiten, um bei einem agilen Projekt ein Dokument mit den Anforderungsspezifikationen zu erstellen: Vor, nach oder während der Realisierung. Im zuletzt genannten Fall kann entschieden werden, ob die Anforderungen vor, während oder nach Abschluss eines Sprints dokumentiert werden sollen. (Bednarczyk)

Zusammenfassung

BGH-Entscheidungen gibt es hauptsächlich zur Anwenderdokumentation, welche nach BGH-Urteilen ein wesentlicher Bestandteil von Software ist. Dabei handelt es sich, wie in dieser Arbeit dargestellt wurde, jedoch nicht um die einzige Art von Dokumentation.

In der vorliegenden Arbeit wurden einige Arten von Dokumentationen beschrieben und im agilen Kontext durchleuchtet. Bei der Frage, welche Art von Dokumentation notwendig ist, kommt es immer auf die Art des Projekts an. Um einen Rechtsstreit oder einen wirtschaftlichen Schaden von vorneherein zu vermeiden, empfiehlt es sich Anforderungen an die Dokumentation eines Projektes im Vertrag oder in der Leistungsbeschreibung des Vertrages klar zu definieren.

Agiles Vorgehen darf keine Ausrede für das Fehlen jeglicher Dokumentation sein. Die Tatsache, dass sich in agilen IT-Projekten viele Entwickler auf Scrum oder die Verwendung agiler Entwicklungs-Methoden wie z.B. Backlogs berufen, reicht nicht aus, um eine ordnungsgemäße Projektdurchführung zu begründen.

Literaturverzeichnis

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J. & Thomas, D. (2010): Manifesto for Agile Software Development

Bednarczyk, Marta; Queins, Stefan: Dokumentation in agilen Projekten – so geht's. In: ProjektMagazin (2013), Sonderdruck - Nr. 10/2013

Hoppen, Peter: Software-Anforderungsdokumentation. In: Computer und Recht (2015), Nr. 11, S. 747 – 760

Liesegang, Wiegand: Projektmanagement und die dazugehörige Dokumentation. In: Computer und Recht (2015), Nr. 8, S. 541 – 556

Müller, Markus: Vertragsgestaltung bei Agilen Softwareentwicklungsverträgen. HMD Praxis der Wirtschaftsinformatik 53.2 (2016): S. 213 - 223.

Pflüger, André; Rauh, Alexander: Dokumentieren in agilen Methoden (Teil 1) – Scrum. <http://blog.sophist.de/2013/10/23/dokumentieren-in-agilen-methoden-teil-1-scrum/> – Aufgerufen am 20.01.2018

Reiss, Manuela; Reiss, Georg. IT-Dokumentation – Was ist das? <http://update.hanser-fachbuch.de/2013/11/it-dokumentation-was-ist-das/> – Aufgerufen am 20.01.2018

Reiss, Manuela; Reiss, Georg: Praxisbuch IT-Dokumentation. Pearson Deutschland (2009).

Schneider, Jochen: Software-Erstellungsverträge. Verlag Dr. Otto Schmidt (2006): S. 177 – 274

Schreiber-Ehle, Sabine: Dokumentation in Softwareerstellungsverträgen. In: Computer und Recht (2015), Nr. 7, S. 469 – 481

Stiemerling, Otto: ITRB 2011- Verlag Dr. Otto Schmidt (2011): S. 286 – 290

Wiegand, Sven: Backlog-Management. <http://agilmanagen.de/backlog-management/> – Aufgerufen am: 20.01.2018

Witzel, Michaela: Software-Erstellungsverträge. Verlag Dr. Otto Schmidt (2006): S. 655-675