

Ludwig-Maximilians-Universität München

Ausarbeitung  
in Juristisches IT-Projektmanagement

# Agile Entwicklung und das V-Modell XT

Christian Hemauer

Verantwortlicher Hochschullehrer: Dr. Frank Sarre  
Betreuer: Dr. Frank Sarre  
Abgabedatum:

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>2</b>
<b>2</b>	<b>Grundlagen</b>	<b>3</b>
2.1	Agile Softwareentwicklung . . . . .	3
2.1.1	Agile Prinzipien . . . . .	3
2.1.2	Agile Praktiken . . . . .	3
2.1.3	Agile Prozesse . . . . .	4
2.2	V-Modell XT . . . . .	5
2.2.1	Aufbau des V-Modell XT . . . . .	5
<b>3</b>	<b>Vereinbarkeit agiler Methoden mit dem V-Modell XT</b>	<b>8</b>
3.1	Agile Ansätze im V-Modell XT . . . . .	8
3.2	V-Modell XT und Scrum . . . . .	9
3.3	V-Modell XT und eXtreme Programming . . . . .	11
<b>4</b>	<b>Vertragliche Auswirkungen</b>	<b>13</b>
4.1	Vertragsgestaltung im V-Modell XT . . . . .	13
4.2	Vertragsgestaltung bei agilen Vorgehensweisen . . . . .	13
4.3	Herausforderungen beim kombinierten Vorgehen . . . . .	14
4.3.1	Vertragsanpassungen . . . . .	14
4.3.2	Grenzen beim kombinierten Vorgehen . . . . .	15
<b>5</b>	<b>Fazit</b>	<b>16</b>
	<b>Abbildungsverzeichnis</b>	<b>17</b>
	<b>Literaturverzeichnis</b>	<b>18</b>

# Kapitel 1

## Einleitung

Spätestens seit dem 2001 veröffentlichten Agilen Manifest erlangen agile Vorgehensweisen zunehmendes Interesse bei Auftragnehmern wie -gebern. Aspekte wie Kundennähe und iterative Lieferungen von funktionierendem Produktivcode sind dabei essentielle Bestandteile und Vorteile der agilen Softwareentwicklung. Wie auch bei den klassischen, meist wasserfallartigen und schwergewichtigeren Vorgehensmodellen entscheiden jedoch Rahmenbedingungen wie Projektanforderungen, Teamgröße und -zusammensetzung nach welchem Verfahren sinnvollerweise vorgegangen wird. IT-Projekte der öffentlichen Hand in der Bundesrepublik Deutschland beispielsweise fordern im Rahmen der EVB-IT Verträge ein Vorgehen nach einem der klassischen Vorgehensmodellen - dem V-Modell XT.

Diese Ausarbeitung beschäftigt sich mit der Möglichkeit agile Ansätze im V-Modell XT zu integrieren. Dabei stehen Integrationsmöglichkeiten agiler Vorgehensmodelle im V-Modell XT, die damit zusammenhängende Projektdurchführung, sowie die dabei entstehenden, vertraglichen Herausforderungen im Vordergrund.

Dazu werden zunächst in Kapitel 2 wesentliche Grundlagen im Bereich agiler Softwareentwicklung und dem V-Modell XT vermittelt. In Kapitel 3 wird dann auf agile Ansätze im V-Modell selbst, sowie die Möglichkeit der Kombination mit bestehenden agilen Vorgehensmodellen und die Folgen für die Projektdurchführung eingegangen. In Kapitel 4 werden kurz vertragliche Voraussetzungen bei agilen Vorgehensmodellen und dem V-Modell XT erläutert, bevor auf vertragliche Herausforderungen und Grenzen bei der Kombination beider Vorgehensmodelle eingegangen wird.

# Kapitel 2

## Grundlagen

Dieses Kapitel deckt Grundlagen ab, die für das Verständnis der Arbeit wichtig sind. Dazu werden zunächst wesentliche Bestandteile der agilen Softwareentwicklung erläutert bevor auf Elemente und Aufbau des V-Modell XT eingegangen wird.

### 2.1 Agile Softwareentwicklung

Während klassische Vorgehensweisen in der Softwareentwicklung ihren Fokus auf Prozessen, Prozessorientierung und Dokumentation haben, verfolgt die agile Softwareentwicklung häufige Lieferungen wertschöpfender Produkte, Individuen und Kommunikation. Ziel ist es dabei den organisatorischen und prozessbedingten Overhead, der klassische Vorgehensweisen üblicherweise auszeichnet, zu minimieren und projektspezifischer sowie produktorientiert zu arbeiten. Wesentlicher Bestandteil sind ein iteratives oder inkrementelles Vorgehen, sowie ein hohes Maß an Verantwortlichkeit im Team.

In diesem Abschnitt werden grundlegende Elemente der agilen Softwareentwicklung vermittelt. Unterschieden werden dabei Agile Prinzipien, Agile Praktiken und Agile Prozesse.

#### 2.1.1 Agile Prinzipien

Die grundlegenden agilen Prinzipien wurden 2001 von Kent Beck et al. im Manifest für agile Softwareentwicklung [1] festgehalten. Sie lassen sich grob in zielorientierte, sozial- und technisch-fokussierte Prinzipien einteilen. Im Wesentlichen postulieren sie die direkte Kommunikation - ob nun mit dem Kunden oder dem Team -, den Willen zu "[...] *technischer Exzellenz* [...]" und den Fokus auf funktionstüchtige Produkte. Diese Prinzipien bedienen die Leitsätze des agilen Manifest. Wie auch in [1] vermerkt, fordern die Leitsätze inhaltlich keine grundsätzliche Substitution traditioneller Vorgehensweisen, sondern stellen die Präferenzen im Rahmen der agilen Softwareentwicklung klar.

#### 2.1.2 Agile Praktiken

Agile Praktiken beschreiben ein konkretes Vorgehen im Rahmen einer Problemstellung innerhalb eines agilen Projekts. Sie können als *Best Practices* zur Erfüllung der agilen Prinzipien und Leitfäden verstanden werden [2].

Im Verlauf dieses Abschnitts werden dabei das *Test Driven Development*, das *Pair Programming* sowie die *Retrospective* näher erläutert.

**Test Driven Development** Das Test Driven Development beschreibt eine Praktik bei der Tests als Steuerungselement der Softwareentwicklung verwendet werden und zeitlich immer vor Produktivcode erstellt werden. Der Ablauf - häufig auch als *Red-Green-Refactoring* bezeichnet - lässt sich nach [3] wie folgt beschreiben. Unabhängig von der eingesetzten Programmiersprache wird zunächst immer ein Test geschrieben, der minimale Anforderungen an den Produktivcode stellt und aufgrund der fehlenden Implementierung fehlschlägt (Red). In der Folge wird der Produktivcode soweit geschrieben, dass er den Ansprüchen des Test gerecht wird und ihn erfolgreich durchläuft (Green). Der nächste Schritt sieht eine Refaktorisierung des Produktivcode vor, die über die Minimalanforderung des Tests hinausgeht und den Einsatz verbreiteter Best Practices sowie Design Patterns vorsieht (Refactoring). Das Verhalten des Codes wird dabei nicht verändert. Nach dem Refaktorisieren werden bestehende Test erweitert oder neue Tests erstellt, wobei wieder deren Fehlschlag den Übergang in die nächste Phase darstellt.

**Pair Programming** Nach [5] stellt das Pair Programming eine Praktik dar, bei der Softwareentwickler zu zweit vor einer Maschine Problemstellungen lösen und Produktivcode entwickeln. Je nach Interpretation können dabei die Rolle des *Driver* und des *Navigator* unterschieden werden. Ersterer macht die Eingaben am Computer und entwirft das Design, letzterer stellt die Einhaltung gängiger Industriestandards und die Vermeidung von Fehlern sicher. Ziel dieser Praktik ist neben der Förderung der Kommunikation, die Vermeidung von Fehlern und die Verbreitung von (projekt- und nicht projekt-bezogenem) technischem Know-How innerhalb des Teams.

### 2.1.3 Agile Prozesse

Agile Prozesse können weitestgehend als Vorgehensmodelle verstanden werden, deren grundlegende Philosophie und Werkzeuge den in 2.1 betrachteten Prinzipien und Leitfäden gerecht werden. Exemplarisch werden in diesem Abschnitt das sog. *eXtreme Programming* (XP) und *Scrum* kurz erläutert.

**eXtreme Programming** Das eXtreme Programming (XP) zeichnet sich durch ein zyklisches Vorgehen auf allen Ebenen aus [4]. Das *Pair-Programming* und *Unit-Test* entsprechen dabei den in den Abschnitten 2.1.2 angesprochenen Inhalten. eXtreme Programming zielt darauf ab neu entwickelte Komponenten schnellst möglich in ein lauffähiges System zu integrieren (siehe dazu *Continuous Integration*). Zur Synchronisation innerhalb des Entwicklerteams sind sog. *Standup Meetings* gedacht. Ein sehr hohes Maß an Agilität gewinnt eXtreme Programming durch die kurzen Iterationszyklen im Wochenrhythmus und den Verzicht auf ein Pflichtenheft. Folglich werden dem Kunden innerhalb von wenigen Wochen lauffähige Systemversionen erzeugt, bevor das weitere Vorgehen ausgehen von dessen Interessen und Präferenzen geplant wird. Ebenso wie die Iterationszyklen sind auch die Releasezyklen kurz. Anstelle eines finalen Releases, zielt eXtreme Programming auf monatliche Releases ab. Ein weiterer

Aspekt des eXtreme Programming ist die starke Teamorientierung. Dokumente und Programmcode werden als Kollektiveigentum angesehen und verantwortet.

**Scrum** Bei Scrum werden alle zu einem Projekt - dem sog. *Product* - gehörigen Aufgaben als *User Stories* im *Product Backlog* gepflegt und verwaltet. Zur Auswahl der Aufgaben für die nächste Iteration, dem sog. *Sprint*, ist das *Sprint Planning* gedacht. Bei diesem Meeting diskutiert das Team über den Schwierigkeitsgrad der einzelnen User Stories, verteilt sie auf die Teammitglieder und hinterlegt sie im sog. *Sprint Backlog*. Während des Sprints arbeitet das Team die Aufgaben ab, wobei wie auch beim XP tägliche Treffen zur Synchronisation innerhalb des Teams verwendet werden (sog. *Daily Scrum*). Wesentlicher Bestandteil des Sprints ist, dass das Resultat eine läuffähige (*potentially shippable*) Verbesserung (*product increment*) gegenüber dem vorangegangenen Sprint darstellt. Nach der Durchführung eines Sprints sieht Scrum eine *Sprint Retrospective* vor, bei der das Team über die Qualität des Sprints diskutiert und Verbesserungsvorschläge eingebracht werden können. Dieses Vorgehen wiederholt sich bis der Product Backlog abgearbeitet ist und die in der *Definition of Done* gelisteten Fertigstellungskriterien erfüllt sind.

## 2.2 V-Modell XT

Das V-Modell XT ist ein klassisches Vorgehensmodell, das als Weiterentwicklung des V-Modell '97 gilt aktuell in der Version 2.0 vorliegt. Ziel der Weiterentwicklung ist die “[...] erhöhte Anpassbarkeit für spezifische Projektumfelder [...]” [7] gegenüber dem klassischen Wasserfallmodell, die durch das *eXtreme Tailoring (XT)* erreicht wird. Im Rahmen der EVB-IT Verträge ist es das Standardvorgehensmodell für öffentliche IT-Projekte in Deutschland.

In diesem Abschnitt wird der Aufbau des V-Modell XT kurz erläutert.

### 2.2.1 Aufbau des V-Modell XT

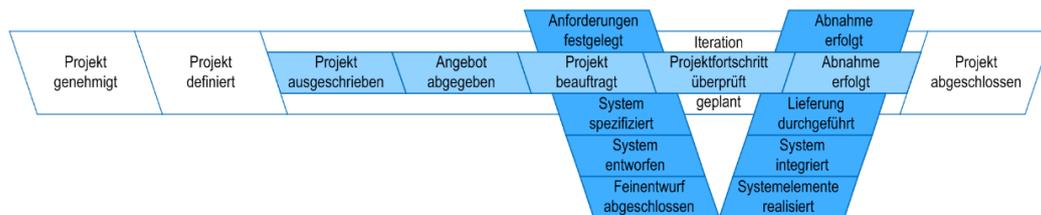


Abbildung 2.1: Entscheidungspunkte des V-Modell XT. Entscheidungspunkte, die Managementaspekte abdecken sind weiß hinterlegt, die der Auftraggeber-/Auftragnehmer-Schnittstelle hellblau und diejenige mit Bezug auf die Systemerstellung dunkelblau. (Quelle: nach [7], Seite 17)

Um das V-Modell XT für verschiedene Projektumgebungen zugänglich zu machen, werden während der Initialisierung des V-Modell XT verschiedene Projekttypen unterschieden. Diese sind basierend auf [7] (S.294 - 298) nachfolgend gelistet:

- **Systementwicklung (AG)** - befasst sich mit V-Modell Projekten auf Auftraggeberseite
- **Systementwicklung (AN)** - befasst sich mit V-Modell Projekten auf Auftragnehmerseite
- **Systementwicklung (AG/AN)** - befasst sich mit V-Modell Projekten ohne Trennung in zwei separate Projekte
- **Einführung und Pflege eines organisationsspezifischen Vorgehensmodells** - befasst sich mit V-Modell Projekten mit dem Ziel ein Vorgehensmodell in einer Organisation zu etablieren

Zusätzlich zu dieser Spezifizierung, wird das Projekt anhand sog. Projektmerkmale charakterisiert (siehe dazu [7] S. 298 - 302). Durch dieses sog. Tailoring auf Basis des Projekttyps, der Projektmerkmale und der Projektvariante wird ein Anwendungsprofil des Projekts erstellt. Ausgehend von diesem Anwendungsprofil ergibt sich die Projektdurchführungs- und Entwicklungsstrategie (siehe Abbildung 2.2 oben). Wie in Abbildung 2.2 zu sehen, gibt diese Strategie vor welche Entscheidungspunkte im Projekt auftauchen und in welcher Reihenfolge sie abgearbeitet werden. Entscheidungspunkte stellen dabei signifikante Meilensteine dar (vgl. Abbildung 2.1), wobei die dem Entscheidungspunkt zugeordneten Ergebnisse (sog. Produkte) evaluiert werden und die weitere Projektdurchführung diskutiert wird. Zu diesen Ergebnissen zählt auch eine umfangreiche Dokumentation. So fordert das V-Modell XT in jedem Fall neben einem Projekthandbuch und -plan ein Pflichten- und Lastenheft, sowie ein Handbuch zur Qualitätssicherung. Die im Rahmen des Projekts zu verwendenden Vorgehensbausteine gibt dabei die Projektdurchführungsstrategie bzw. die Entwicklungsstrategie vor.

Weitere Besonderheiten des V-Modell XT sind die Unterteilung eines Projekt in die Teilsysteme Gesamtsystem, System, Systemelemente (Hardware) und Systemelemente (Software), die Einbeziehung des Auftraggebers über die Auftraggeber-/Auftragnehmer-Schnittstelle (AG/AN), sowie das strenge Rollenmodell. Letzteres fordert unter anderem einen Projektleiter und einen separaten Qualitätsbeauftragten.

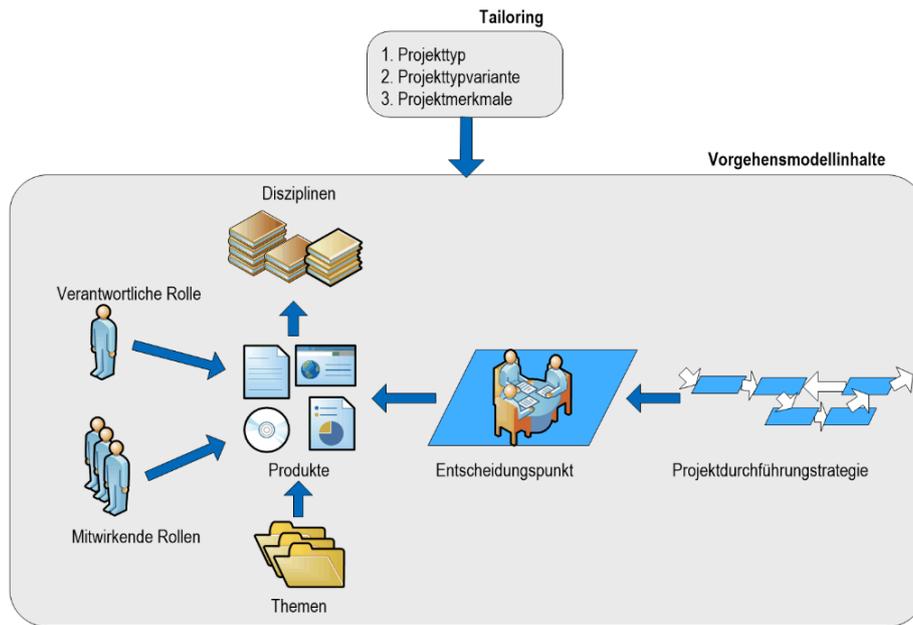


Abbildung 2.2: Grundkonzepte des V-Modell XT (Quelle: nach [7], Seite 12)

## Kapitel 3

# Vereinbarkeit agiler Methoden mit dem V-Modell XT

In den vorangegangenen Kapiteln 2.2 und 2.1 wurden Grundlagen zur agilen Softwareentwicklung und mit dem V-Modell XT zudem ein Vertreter klassischer, wasserfallartiger Vorgehensmodelle vorgestellt. In diesem Kapitel werden Möglichkeiten betrachtet, die die Vereinbarkeit von agilen Vorgehensweisen im V-Modell XT ermöglichen.

Dazu wird zunächst auf agile Ansätze im V-Modell XT selbst eingegangen, bevor die Kombination mit agilen Prozessen näher beleuchtet wird. Exemplarisch werden dazu das eXtreme Programming und Scrum verwendet.

### 3.1 Agile Ansätze im V-Modell XT

Das V-Modell XT gehört zwar zu den klassischen, nicht agilen Vorgehensmodellen, dennoch bietet es durch das Tailoring Schnittstellen, die es erlauben agile Praktiken und Prozesse im V-Modell XT einzuführen. Dies wird im wesentlichen durch die verschiedenen Projektdurchführungsstrategien bzw. den darin spezifizierten Entwicklungsstrategien ermöglicht. Nachfolgend wird auf die inkrementelle und prototypische Projektdurchführungsstrategie auf Basis des V-Modell XT Version 2.0 eingegangen.

**Inkrementelle Entwicklungsstrategie** Bei dieser Entwicklungsstrategie *“[...] entwickelt, realisiert und liefert (der Auftraggeber) das System in einzelnen Stufen[...]”*

[7] (S.290-291). Dabei wird das in jeder Lieferung enthaltene Inkrement separat vom Auftragnehmer abgenommen, mit dem Ziel dem Auftraggeber möglichst frühzeitig einen mit den wesentlichen Grundfunktionalitäten ausgestatteten Systementwurf zu liefern. Diese Entwicklungsstrategie sieht dabei auch Änderungsvorschläge von Seiten des Auftraggebers vor, die vom Auftragnehmer im Rahmen eines Änderungsmanagements bei der Erstellung des nächsten Inkrements zu beachten sind. Auf diese Weise unterstützt diese Entwicklungsstrategie Iterationen unter Einflussnahme des Auftraggebers. Voraussetzung für diese Entwicklungsstrategie ist, dass die *“[...] Anforderungen als relativ stabil eingeschätzt werden [...]”* [7]. Grundlegende Änderungen an den in der ersten Iteration festgelegten Anforderungen sind bei dieser Entwicklungsstrategie nicht vorgesehen. Vertraglich besteht dabei die Möglichkeit die Inkremente separat zu handhaben.

**Prototypische Entwicklungsstrategie** Für den Fall, dass die Anforderungen vorab nur unzureichend definiert werden können und das Projekt zudem Realisierungsrisiken offenbart, bietet das V-Modell XT die Prototypische Entwicklungsstrategie [7] (S.291). Ähnlich wie bei Inkrementellen Entwicklungsstrategien werden unter Mitwirkung des Auftraggebers iterativ Inkremente geliefert und abgenommen. Gleiches gilt für die vertragliche Handhabung der Inkremente.

Im Gegensatz zur Inkrementellen Entwicklungsstrategie sind auch grundlegende Änderungen hinsichtlich Architektur im Zuge einer Iteration nicht vom Änderungsmanagement ausgeschlossen. Zudem ermöglicht diese Entwicklungsstrategie eine bis zur Fertigstellung des Prototypen verzögerte Lieferung der erforderlichen Dokumentation. Dieser Umstand macht diese Entwicklungsstrategie zur agilsten des V-Modell XT.

### 3.2 V-Modell XT und Scrum

Auch wenn das schwergewichtige und dokumentenzentrierte V-Modell XT zunächst unvereinbar mit dem leichtgewichtigen und Verwaltungs-Overhead vermeidenden Scrum scheint, besteht die Möglichkeit einer Integration von Scrum im V-Modell XT. Die Integrationsanleitung basiert dabei auf [8] und [9].

**Integration** Grundsätzlich lassen sich dabei nur Prozesse kombinieren, die ein iterativ-inkrementelles vorgehen erlauben. Für die Scrum-Prozesse ist dies in den meisten Fällen gegeben, das V-Modell XT erlaubt dies jedoch nur unter gewissen Umständen - bspw. bei der Wahl entsprechender Projektdurchführungsstrategien. Insgesamt sind dabei drei wesentliche Integrationen notwendig. Zum einen die Aufnahme der Produkte der Entscheidungspunkte des V-Modell XT in die Definition of Done von Scrum, zum anderen die Durchführung von Scrum auf den verschiedenen Ebenen bzw. Teilsystemen der V-Modell XT Systementwicklung (siehe Abbildung 3.1) und zuletzt die Anpassung der Rollen [9]. Dies ist vor allem deshalb notwendig, da sich die Scrum Releasezyklen und die damit einhergehende Lieferung eines *potentially shippable product increment* nach jedem Sprint nicht mit dem Releasezyklus des V-Modell XT vereinbaren lassen. Zwar sieht auch das V-Modell XT unter Verwendung der entsprechenden Projektdurchführungsstrategie iterative Releases vor, diese umfassen jedoch deutlich mehr Ergebnisse als ein *product increment*. Das Modell des Sprints lässt sich folglich im Teilsystem Softwareentwicklung realisieren, indem Entscheidungspunkte (und die damit verbundenen Produkte) durch Sprints erreicht werden.

Wie in Abbildung 3.1 zu sehen liefern die unteren Ebenen Ergebnisse oder Produkte, die in den oberen Ebenen integriert werden. An dieser Stelle führt die Kombination aus dem V-Modell XT und Scrum zu einem Mehraufwand, da zusätzliche Definitions of Done eingeführt werden müssen, die die Übernahme entsprechender Produkte von höheren Ebenen regeln.

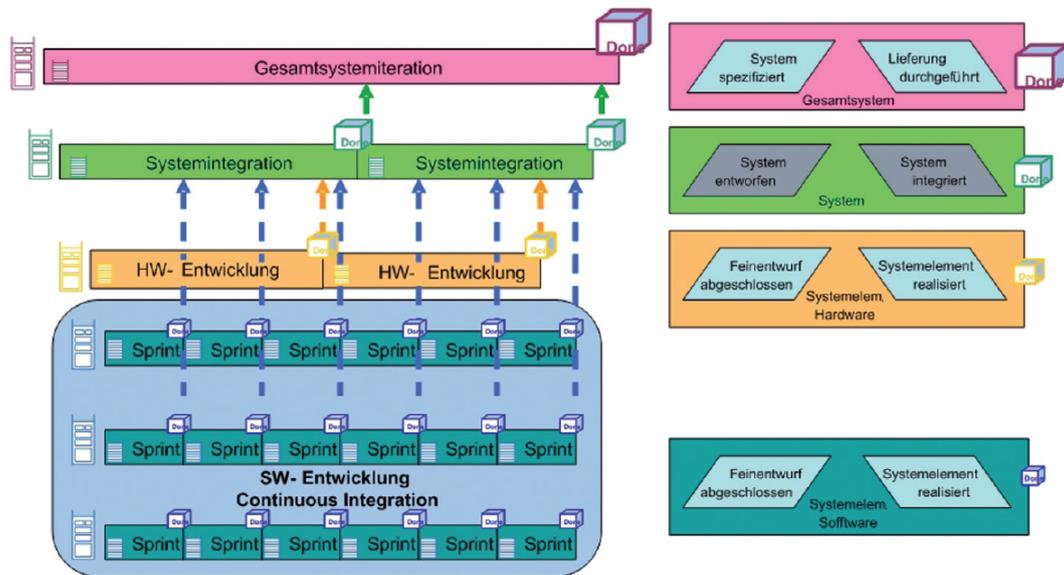


Abbildung 3.1: Scrum auf den verschiedenen Ebenen des V-Modell XT (links). Aufnahme der V-Modell XT Artefakte in die Scrum Definition of Done (links) (Quelle: nach [8], Seite 61)

**Projektdurchführung** Um eine Reibungslose Durchführung eines Projekts unter Verwendung der Kombination aus V-Modell XT und Scrum zu gewährleisten, ist als Projektdurchführungsstrategie die Prototypische Systementwicklung notwendig. Diese erlaubt Iterationen auf den verschiedenen Teilsystemen des V-Modell XT und sieht Unklarheiten bei der Spezifikation vor. Um die Iterationen in den Teilsystemen zu ermöglichen, sollte ein Product und ein Sprint Backlog eingerichtet werden. Der Product Backlog enthält die Produkte der Entscheidungspunkte und spiegelt damit die Definition of Done wider. Die Produkte, die innerhalb eines Sprints zu realisieren sind, werden entsprechend im Sprint Backlog hinterlegt.

Ein wichtiger Punkt dabei ist, dass generell alle einem Sprint zugeordneten Produkte vorliegen müssen und diese zudem Konsistenz zu den Produkte vorangegangener Iterationen erfordern. Diese Konsistenzforderung zieht auch in Kombination mit Scrum ein hohes Maß an Dokumentation nach sich. Der Dokumentationsaufwand kann zwar - wie es auch der Rahmen der Prototypischen Systementwicklung vorsieht - bis zur Fertigstellung der Systemelemente verzögert werden, eine Verringerung des Aufwands ist jedoch nur unter Absprache mit dem Auftraggeber inklusive entsprechender Dokumentation im Projekthandbuch möglich.

Weitere projektrelevante Dokumentation, wie etwa das Pflichten- und Lastheft werden durch den Product Backlog auf der Ebene des Gesamtsystems realisiert. Nach [8] stellt dieser zu Beginn das Lastenheft dar und wird im Laufe weiterer Iterationen um die Inhalte des Pflichtenhefts erweitert.

Hinsichtlich der Rollenbesetzung ist bei der Projektdurchführung zu beachten, dass die Anforderungen des V-Modell XT erfüllt werden. Dazu kann das in Scrum definierte Rollenmodell verwendet und erweitert werden. So übernimmt auf jeder Ebene der Scrum Master zusätzlich die Rolle des Qualitätsbeauftragten, während der Pro-

duct Owner die Rolle des Projektleiters inne hat und zudem die AG/AN-Schnittstelle bedient. Eine zusätzliche Rolle, die nicht vorgegeben ist, jedoch aufgrund der verschiedenen Ebenen, auf denen Scrum angewendet wird, von Vorteil ist, ist die Rolle der sog. Product Owner Vertreter. Diese sollten ebenfalls auf allen Ebenen vorhanden sein, synchronisieren den Fortschritt der verschiedenen Teams untereinander und vertreten Kundenanforderungen gegenüber dem Team der darunterliegenden Ebene. Das Rollenmodell und die Interaktion auf den unterschiedlichen Ebenen wird in Abbildung 3.2 verdeutlicht.

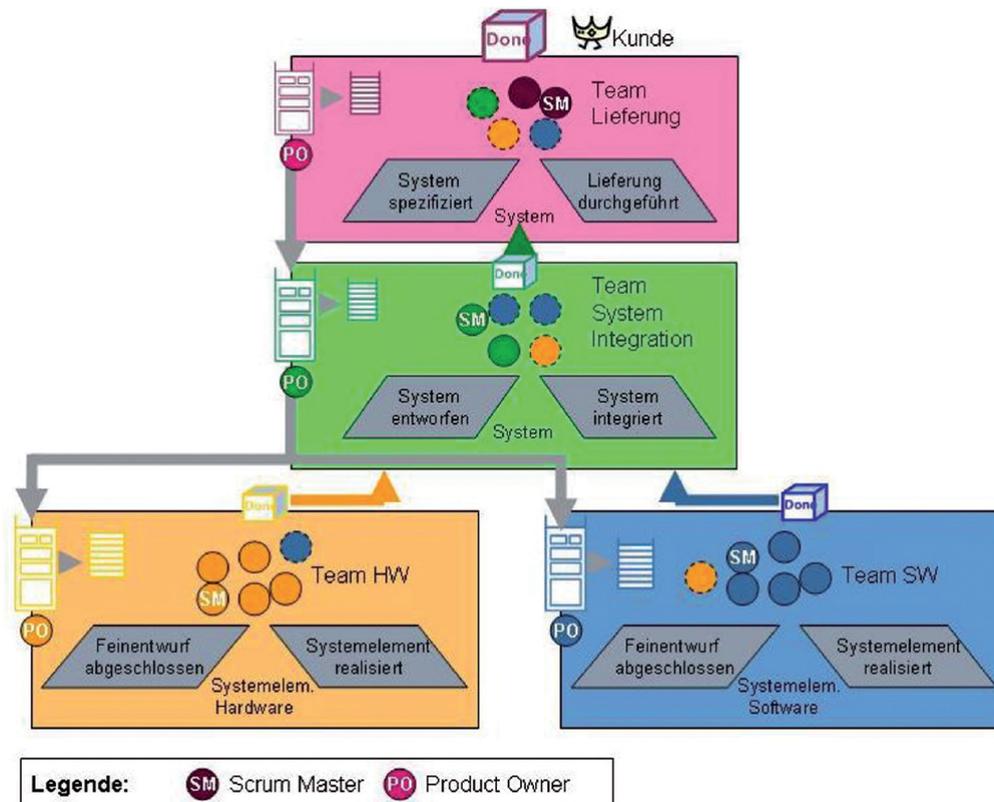


Abbildung 3.2: Das Rollenmodell bei Kombination von V-Modell XT und Scrum, sowie die Interaktion auf den verschiedenen Ebenen der Systementwicklung (Quelle: nach [8], Seite 62)

### 3.3 V-Modell XT und eXtreme Programming

Im Gegensatz zur Kombination des V-Modell XT mit Scrum, fällt die Kombination mit dem eXtreme Programming deutlich schmaler aus. Grund dafür ist, dass sich einige Praktiken nicht mit dem V-Modell XT kombinieren lassen, da sie sich gänzlich widersprechen.

**Integration** Die Integration des eXtreme Programming im V-Modell XT sieht im Wesentlichen die Integration aller mit dem V-Modell XT vereinbaren Praktiken auf der Ebene der Softwareentwicklung vor. Eine Integration auf anderen Ebenen gestalten sich aufgrund sehr unterschiedlicher Auffassungen schwierig. So ermöglicht

das eXtreme Programming die Zuweisung von Verantwortlichkeiten an Teams - ein Umstand den das feingranulare Rollenmodell des V-Modell XT nicht vorsieht. Des weiteren ermöglicht das extreme Programming einen vollkommenen Verzicht auf ein Pflichtenheft. Ein integrierbares Pendant zum vom V-Modell XT geforderten Pflichtenheft ist somit unter Umständen nicht vorhanden.

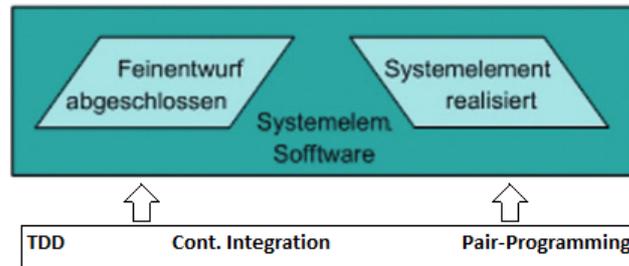


Abbildung 3.3: eXtreme Programming auf der Softwareebene des V-Modell XT. (Quelle: nach [8], Seite 61)

**Projektdurchführung** Wie auch bei Scrum ist für das eXtreme Programming die prototypische Entwicklungsstrategie der geeignete Entwicklungsrahmen des V-Modell XT. Für die Projektdurchführung bedeutet die Integration des eXtreme Programming vor allem Änderungen im Bereich der praktischen Softwareentwicklung. User Stories werden im Rahmen des Pair Programming (siehe Kapitel 2.1.2) abgearbeitet. Um dem Qualitätsanspruch des eXtreme Programming gerecht zu werden wird dabei nach dem Test-First-Ansatz (siehe Test-Driven Development in Kapitel 2.1.2) vorgegangen und Produktivcode häufig in lauffähige Systeme integriert. Konkrete Ansätze zur Ermöglichung

# Kapitel 4

## Vertragliche Auswirkungen

Dieses Kapitel behandelt die vertraglichen Aspekte, die bei der Verwendung des V-Modell XT, einer agilen Vorgehensweise oder dem kombinierten Vorgehen zu beachten sind. Der Schwerpunkt liegt dabei auf den Herausforderungen bei der Kombination von agilen Methoden und dem V-Modell XT.

Zunächst wird grundlegend auf die Vertragsgestaltung beim V-Modell XT und bei agilen Vorgehensweisen eingegangen, bevor die Herausforderungen beim kombinierten Vorgehen auf Basis des EVB-IT Erstellungsvertrags näher beleuchtet werden.

### 4.1 Vertragsgestaltung im V-Modell XT

Auch wenn das V-Modell XT durch das Tailoring eine Vielfalt von Herangehensweisen anbietet, so sieht das V-Modell XT unabhängig vom Resultat des Tailorings *“[...] einen Werkvertrag als Basis für die Zusammenarbeit zwischen Auftraggeber und Auftragnehmer [...]“* [10] vor, fordert (verpflichtend) die Existenz gewisser Produkte und macht strikte Vorgaben zur Meilensteinplanung. Weiter werden Dokumente wie das Projekthandbuch und das Handbuch zur Qualitätssicherung (QS-Handbuch) gefordert.

Im Rahmen des Werkvertrag fordert das V-Modell XT - auch unter Verwendung spezifischer Durchführungsstrategien - eine gewisse Grundmenge konkreter vorabdefinierte Anforderungen an das zu erstellende Werk, die bei der späteren Abnahme als Referenzcharakter haben. Dabei sieht der Vertrag keine aktive Mitarbeit des Auftraggebers am Werk selbst vor, wohl aber eine Mitgestaltung. Dieser Aspekt wird im V-Modell XT durch die separate Auftraggeber-/Auftragnehmerschnittstelle und die Unterteilung in Auftraggeber- und Auftragnehmerprojekte abgedeckt (siehe Kapitel 2.2).

### 4.2 Vertragsgestaltung bei agilen Vorgehensweisen

Im Gegensatz zu Projekten, die nach klassischer Vorgehensweise (wie bspw. dem V-Modell XT) bearbeitet werden, sieht das agile Projektmanagement nur grob vorab definierte Anforderungen ohne ausführlichem Lastenheft vor. Vielmehr läuft der Planungsprozess parallel zur Entwicklung und ermöglicht auf diese Weise schnelles Reagieren auf Anforderungsänderungen.

Eben dieser Umstand macht die Vertragsgestaltung bei agilen Projekten so kompli-

ziert. So könnte sowohl der Dienstvertrag als auch der Werkvertrag verwendet werden [11]. Der Dienstvertrag ist dabei insofern unzureichend, als das vom Auftraggeber am Ende des Projekts ein Ergebnis erwartet wird, für das der Auftragnehmer haftet. Dem Werkvertrag hingegen widerspricht die Tatsache, dass die Anforderungen vom Auftragnehmer zu erstellenden Werk bei Projektbeginn noch nicht hinreichend konkret feststehen. Beschaffungs- und Vergütungsvereinbarungen werden durch diese Umstände erschwert [11].

Ein weiterer wichtiger Punkt ist die Mitwirkung des Auftraggebers. Diese ist bei agilen Projekten vorgesehen und sollte im Hinblick auf einen effektiven Change-Management-Prozesses bei den Vertragsvereinbarungen berücksichtigt werden.

### 4.3 Herausforderungen beim kombinierten Vorgehen

Wie bereits in Kapitel 2.2 erwähnt, ist das V-Modell XT das Standardvorgehen für IT-Projekte der öffentlichen Hand. Aus diesem Grund wird in diesem Abschnitt auf die vertraglichen Herausforderungen im Rahmen des EVB-IT Erstellungsvertrag [12] eingegangen, da dieser sich mit einigen Anpassungen für das V-Modell XT in Kombination mit einem agilen Verfahren eignet. Im Folgenden wird dabei auf die notwendigen Anpassungen, aber auch auf die Grenzen des Erstellungsvertrags eingegangen

#### 4.3.1 Vertragsanpassungen

Grundsätzlich ist es notwendig, die agilen Verfahren oder Prinzipien, die in Kombination mit dem V-Modell XT eingesetzt werden, in den Vertrag mitaufzunehmen. Der EVB-IT Erstellungsvertrag ermöglicht diese Aufnahme über den Vertragspunkt 17.5 SONSTIGE VEREINBARUNGEN. Zu beachten ist dabei, dass der der EVB-IT Erstellungsvertrag trotz seiner “[...] *Öffnung für agile Vorgehensmodelle* [...]“ [14] auf einem Werkvertrag basiert und dafür auch keine Änderung vorsieht. Um dieser Tatsache unter dem Umstand, dass zu Beginn des Projekts die Anforderungen an die Software noch nicht hinreichend konkret feststehen, vertraglich gerecht zu werden ist es notwendig eine schriftliche Beschreibung der Software hinsichtlich Kernfunktionalitäten, Designanforderungen und Einsatzzweck in den Vertragsgegenstand (Kapitel 1.1 im EVB-IT Erstellung) mit aufzunehmen und damit das zu erstellende Werk zumindest grundlegend zu spezifizieren [14].

Ein weiterer wichtiger Punkt der vertraglich verankert werden muss, ist die Regelung der Verantwortlichkeiten. Zum einen muss gewährleistet werden, dass die vom V-Modell XT geforderten Rollen belegt sind und zum anderen weitere, im Rahmen des agilen Vorgehens eingeführte Rollen (vgl. Stellvertreter Product Owner in Kapitel 3.2) genehmigt sind und somit bei der späteren Aufgabenverteilung im Projektplan berücksichtigt werden.

Durch den Einsatz agiler Methoden wird auch die aktive Mitwirkung des Auftraggebers gefordert. Der EVB-IT Erstellungsvertrag sieht diese zwar grundsätzlich vor, macht dazu bis auf die grundlegenden Leistungen der Erstellungs-AGB [13] aber keine genaueren Angaben. Um einen reibungslosen Ablauf des agilen V-Modell XT zu ermöglichen, sollten Details zur Mitwirkung wie Umfang, Häufigkeit und Dauer dem Erstellungsvertrag als Anlagen zum Vertragspunkt 12 MITWIRKUNG DES AUFTRAGGEBERS beigefügt werden [14]. Auf diese Weise können entsprechend vom

Auftraggeber geforderte Produkte in den Projektplan aufgenommen werden. Gleiches gilt für Themen wie Kündigung, Sach-/Rechtsmangelhaftung und insbesondere Schadensersatz. Diese Aspekte werden im Vertragspunkt 14 MÄNGELHAFTUNG (GEWÄHRLEISTUNG) und 17.4 KÜNDIGUNGSRECHT DES AUFTRAGGEBERS grundlegend behandelt, lassen sich jedoch durch weitere Anlagen (im Rahmen der Erstellungs-AGB) erweitern. Dies ist notwendig zum den Vertrag dem iterativen Charakter der Projektdurchführung anzupassen.

### 4.3.2 Grenzen beim kombinierten Vorgehen

Wie im Vertragspunkt 1.1 VERTRAGSGEGENSTAND festgelegt, sieht der Erstellungsvertrag eine “[...] Erstellung bzw. Anpassung von Software auf der Grundlage eines Werkvertrages [...]“ vor [12]. Abweichungen davon sind somit nicht vertragskonform. Dieser Umstand ist somit eine grundlegende Forderungen an alle im Rahmen einer agilen Vorgehensweise eingeführten Anpassungen.

Weitere Einschränkungen beim Erstellungsvertrag gelten bei der Dokumentation. Ein wesentliches Ziel der Nutzung agiler Verfahren ist die Reduzierung des Verwaltungsaufwands [8] und damit auch der Dokumentation. Bei der Verwendung eines kombinierten Vorgehens aus V-Modell XT und agilen Vorgehensweisen auf Basis des EVB-IT Erstellungsvertrags bleibt den Vertragsparteien in dieser Hinsicht jedoch wenig Gestaltungsfreiraum. So fordert die Erstellungs-AGB im Punkt 5 DOKUMENTATION die Dokumentation der Werkleistungen durch den Auftragnehmer. Ein vollkommener Verzicht auf Dokumentation ist folglich rechtlich bedenklich [14]. Weitere, grundsätzlich nicht der Erstellungs-AGB widersprechende Regelungen zu Art und Umfang der Dokumentation, lassen sich jedoch im Vertragspunkt 4.6 DOKUMENTATION des Erstellungsvertrags unterbringen.

# Kapitel 5

## Fazit

Auch wenn die Herangehensweisen agiler Vorgehensmodelle in gewisser Weise dem umfangreichen und in vielerlei Hinsicht restriktiven V-Modell XT widersprechen, schließen sich beide Ansätze generell nicht aus und lassen sogar kombinieren.

In den Kapiteln 3.2 und 3.3 wurden mit dem eXtreme Programming und Scrum Kombinationsmöglichkeiten mit zwei bekannten Vertretern der agilen Softwareentwicklung gezeigt. Ein wesentliches Erfolgskriterium bei der Projektdurchführung dabei ist die Konformität des V-Modell XT nicht durch die agilen Einflüsse zu gefährden. Dies ist ein entscheidender Faktor, wenn der Auftraggeber im Falle von Ungereimtheiten ein entsprechendes Audit zur Prüfung der Konformität zum V-Modell XT anfordert.

Für die Vertragsgestaltung existiert mit dem EVB-IT Erstellungsvertrag ein solides Grundgerüst, das sich mit den im Abschnitt 4.3.1 angesprochenen verträglichen Anpassungen für ein agiles V-Modell XT eignet. Um einen reibungsfreien Ablauf des Projekts zu gewährleisten, sollten diese schriftlich dokumentiert und in den entsprechenden Anhängen im Vertrag verankert werden. Dabei sollten die im Abschnitt 4.3.2 erwähnten Grenzen beachtet werden. Diese beziehen sich in erster Linie auf den Werkvertrag als Vertragstyp und die grundlegende Pflicht des Auftragnehmers zur Dokumentation.

# Abbildungsverzeichnis

2.1	Entscheidungspunkte des V-Modell XT. Entscheidungspunkte, die Managementaspekte abdecken sind weiß hinterlegt, die der Auftraggeber-/Auftragnehmer-Schnittstelle hellblau und diejenige mit Bezug auf die Systemerstellung dunkelblau. (Quelle: nach [7], Seite 17) . . . . .	5
2.2	Grundkonzepte des V-Modell XT (Quelle: nach [7], Seite 12) . . . . .	7
3.1	Scrum auf den verschiedenen Ebenen des V-Modell XT (links). Aufnahme der V-Modell XT Artefakte in die Scrum Definition of Done (links) (Quelle: nach [8], Seite 61) . . . . .	10
3.2	Das Rollenmodell bei Kombination von V-Modell XT und Scrum, sowie die Interaktion auf den verschiedenen Ebenen der Systementwicklung (Quelle: nach [8], Seite 62) . . . . .	11
3.3	eXtreme Programming auf der Softwareebene des V-Modell XT. (Quelle: nach [8], Seite 61) . . . . .	12

# Literaturverzeichnis

- [1] Beck, Kent et al. (2001): „Manifesto for Agile Software Development“. Abgerufen am 03. 12. 2015 von <http://www.agilemanifesto.org/>.
- [2] IT-Agile GmbH (o. J.): „Agile Praktiken“. Abgerufen am 03. 12. 2015 von <http://www.it-agile.de/wissen/praktiken/>.
- [3] IT-Agile GmbH (o. J.): „Was ist Testgetriebene Entwicklung“. Abgerufen am 04. 12. 2015 von <http://www.it-agile.de/wissen/praktiken/agilestesten/testgetriebene-entwicklung-tdd/>.
- [4] IT-Agile GmbH (o. J.): „eXtreme Programming (XP)“. Abgerufen am 05. 12. 2015 von <http://www.it-agile.de/wissen/methoden/extreme-programming/>.
- [5] Salinger, Stephan; Prechelt, Lutz (2013): Understanding Pair Programming: The Base Layer. BoD - Books on Demand. Seiten 18, 19.
- [6] ScrumAlliance (o. J.): Learn about Scrum“. Abgerufen am 05.12.2015 von <https://www.scrumalliance.org/why-scrum>.
- [7] (2015): V-Modell XT. Version 2.0. München: Verein zur Weiterentwicklung des V-Modell XT e.V. (Weit e.V.).
- [8] Canditt, Sabine; Rauh, Doris; Wittmann, Marion (2011): „Brückenschlag: Das V-Modell XT mit Scrum inside“. In: Objektspektrum., S. 54-61.
- [9] Schmidt, Holger; Kalkofen, Dennis; Böni, Mathias (2014): „Scrum, Kanban, V-Modell XT - was bringt Erfolg?“. CIO. Abgerufen am 10. 12. 2015 von <http://www.cio.de/a/scrum-kanban-v-modell-xt-was-bringt-erfolg,2944862,3>.
- [10] Wagner, Reinhard; Grau, Nino (2014): Basiswissen Projektmanagement - Prozesse und Vorgehensmodelle. Düsseldorf: Reinhard Wagner, Nino Grau.
- [11] Decker, Florian (2014): „Resmedia - Kanzlei für IT-Recht“. Vertragserstellung bei agiler Softwareerstellung - Teil I. Abgerufen am 14. 12. 2015 von <http://blog-it-recht.de/2014/02/21/vertragsgestaltung-bei-agiler-softwareerstellung-teil/>.
- [12] Bundesministerium des Inneren (BMI), (2013): EVB-IT Erstellungsvertrag. Beauftragter der Bundesregierung für Informationstechnik.
- [13] Bundesministerium des Inneren (BMI), (2013): EVB-IT Erstellungs-AGB. Beauftragter der Bundesregierung für Informationstechnik.

- [14] Bischof,; Schneider,; Schiffer, u. a. (2014): „Anwendung der neuen EVB-IT Verträge in der Praxis: Einige ausgewählte Aspekte“. 2014.
- [15] Cio.bund.de, (2015): „IT-Beauftragter der Bundesregierung | Häufig gestellte Fragen“. Abgerufen am 20. 12. 2015 von [http://www.cio.bund.de/Web/DE/Architekturen-und-Standards/V-Modell-XT/Haeufig-gestellte-Fragen/haeufig\\_gestellte\\_fragen\\_node.html](http://www.cio.bund.de/Web/DE/Architekturen-und-Standards/V-Modell-XT/Haeufig-gestellte-Fragen/haeufig_gestellte_fragen_node.html).