

Bernburg  
Dessau  
Köthen



**Hochschule Anhalt**

Anhalt University of Applied Sciences

Hochschule Anhalt

Fachbereich 5: Informatik und Softwarelokalisierung

Studiengang: Master Informationsmanagement

# MASTERTHESIS

## *Projektcontrolling im Rahmen von agilen Softwareentwicklungsprojekten*

**Vorgelegt von:** Lisa Grotepaß

*Matrikelnummer:* 4063641

Gutachter: Prof. Dr. Martin Kütz

Zweitgutachter: Prof. Dr. Michael Worzyk

Abgabe: \_\_.\_\_.2018

# Inhaltsverzeichnis

---

|  |    |
|--|----|
| Inhaltsverzeichnis .....                         | I  |
| Abbildungsverzeichnis .....                      | IV |
| Tabellenverzeichnis .....                        | V  |
| Abkürzungsverzeichnis .....                      | VI |
| 1 Einleitung.....                                | 1  |
| 2 Grundlagen.....                                | 2  |
| 2.1 Projekt .....                                | 2  |
| 2.2 Projektmanagement.....                       | 4  |
| 2.3 Controlling .....                            | 8  |
| 2.4 Projektcontrolling .....                     | 11 |
| 2.5 Agilität.....                                | 11 |
| 2.5.1 Agile Softwareentwicklung .....            | 12 |
| 2.5.2 Agiles Management .....                    | 16 |
| 3 Controlling im Projektbereich.....             | 18 |
| 3.1 Balanced Scorecard & Projekt-Scorecard ..... | 18 |
| 3.1.1 Balanced Scorecard.....                    | 20 |
| 3.1.2 Projekt-Scorecard .....                    | 22 |
| 3.2 Einzelprojekt .....                          | 23 |
| 3.2.1 Beispielhafte Projekt-Scorecard.....       | 25 |
| 3.2.2 Vorteile der Projekt-Scorecard .....       | 25 |
| 3.3 Multiprojektmanagement .....                 | 26 |
| 3.3.1 Beispielhafte Projekt-Scorecard.....       | 29 |
| 3.3.2 Vorteile Scorecard.....                    | 30 |
| 3.4 Vorgehensmodelle in der IT.....              | 31 |
| 3.4.1 Klassische Vorgehensmodelle .....          | 33 |
| 3.4.2 Agile Ansätze .....                        | 40 |

|       |   |     |
|-------|---|-----|
| 4     | Controlling in den unterschiedlichen Projektwelten .....                  | 54  |
| 4.1   | Controlling in der spezifikationsorientierten Projektwelt.....            | 56  |
| 4.1.1 | Kommunikation & Dokumentation .....                                       | 56  |
| 4.1.2 | Steuerung .....   | 58  |
| 4.2   | Controlling in der agilen Projektwelt.....                                | 62  |
| 4.2.1 | Kommunikation .....   | 62  |
| 4.2.2 | Dokumentation .....   | 68  |
| 4.2.3 | Steuerung .....   | 70  |
| 4.3   | Vergleich der Anforderungen.....  | 76  |
| 4.3.1 | Analyse & Planung in der spezifikationsorientierten Projektwelt ....      | 76  |
| 4.3.2 | Analyse & Planung in der agilen Projektwelt .....                         | 77  |
| 4.3.3 | Umsetzung & Abschluss in der spezifikationsorientierten Projektwelt<br>77 |     |
| 4.3.4 | Umsetzung & Abschluss in der agilen Projektwelt .....                     | 78  |
| 4.3.5 | Zusammenfassung der Unterschiede .....                                    | 79  |
| 5     | Agile Anwendung in der Praxis.....  | 82  |
| 5.1   | Typische Project-Constraints in den beiden Projektwelten .....            | 82  |
| 5.1.1 | Projektspezifische Project-Constraints.....                               | 82  |
| 5.1.2 | Unternehmensbezogene Project-Constraints.....                             | 84  |
| 5.1.3 | Kosten als Project-Constraints.....                                       | 85  |
| 5.1.4 | Projekt-Individuen als Project-Constraints .....                          | 86  |
| 5.1.5 | Umgang mit Project-Constraints .....                                      | 86  |
| 5.2   | Agile Projektarbeit in einer klassischen Unternehmensstruktur .....       | 87  |
| 5.3   | Skalierung von agilen Modellen für Großprojekte.....                      | 88  |
| 5.3.2 | Skalierung – Vor- und Nachteile.....                                      | 95  |
| 5.4   | Entscheidungsfindung für Unternehmen.....                                 | 96  |
| 5.4.1 | Szenario 1: Neuentwicklung.....   | 99  |
| 5.4.2 | Szenario 2: Migration .....   | 101 |

|   |                                   |     |
|---|-----------------------------------|-----|
| 6 | Fazit und Ausblick .....          | 103 |
|   | Quellenverzeichnis .....          | 104 |
|   | Selbstständigkeitserklärung ..... | 112 |

# Abbildungsverzeichnis

---

|   |    |
|---|----|
| Abbildung 1: Magisches Dreieck.....   | 6  |
| Abbildung 2: Veränderung der Parameter im Magischen Dreieck. ....               | 7  |
| Abbildung 3: Controlling-Regelkreis.....  | 10 |
| Abbildung 4: Schnittstelle Projekt- & Unternehmenscontrolling .....             | 20 |
| Abbildung 5: Wasserfallmodell.....  | 34 |
| Abbildung 6: Spiralmodell .....   | 35 |
| Abbildung 7: V-Modell.....  | 36 |
| Abbildung 8: Beispiel Story Map .....   | 44 |
| Abbildung 9: Rollen in Scrum.....   | 45 |
| Abbildung 10: Timeboxing am Beispiel Sprint Planning Meeting.....               | 46 |
| Abbildung 11: Lasten- und Pflichtenheft als Grundlage für Zielvereinbarung .... | 57 |
| Abbildung 12: Beispiel Projektstrukturplan.....                                 | 59 |
| Abbildung 13: Beispiel Story Card .....   | 63 |
| Abbildung 14: Feedback am Beispiel Sprint Review Meeting .....                  | 65 |
| Abbildung 15: Visualisierung einer Epic-User-Story mittels Story Cards .....    | 69 |
| Abbildung 16: Beispiel eines Burndown-Charts für die Epic-Story „Checkout“ ..   | 71 |
| Abbildung 17: Verantwortung in klassischer Projektsituation.....                | 72 |
| Abbildung 18: Verantwortung in Scrum .....                                      | 73 |
| Abbildung 19: Agile Kapsel. ....  | 87 |
| Abbildung 20: Scrum of Scrums. ....   | 90 |
| Abbildung 21: LeSS Framework .....  | 91 |
| Abbildung 22: LeSS Hufeisen.....  | 92 |
| Abbildung 23: Requirement Areas .....   | 93 |
| Abbildung 24: Nexus Framework.....  | 94 |

## Tabellenverzeichnis

---

|   |    |
|---|----|
| Tabelle 1: Die 12 Agilen Prinzipien .....   | 15 |
| Tabelle 2: Projekt-Scorecard vom 01.03.2018.....  | 25 |
| Tabelle 3: Multiprojekt-Scorecard vom 01.03.2018.....                                   | 29 |
| Tabelle 4: Beispiel eines Kanban-Boards in Tabellenform .....                           | 49 |
| Tabelle 5: Typische Gründe für Projektscheitern .....                                   | 55 |
| Tabelle 6: Unterschiede in den Bereichen Kommunikation, Dokumentation & Steuerung. .... | 79 |
| Tabelle 7: Project-Constraint Komplexität.....  | 82 |
| Tabelle 8: Project-Constraint Projektgröße .....  | 83 |
| Tabelle 9: Project-Constraint Zeit .....  | 83 |
| Tabelle 10: Project-Constraint Vertragsart.....   | 83 |
| Tabelle 11: Project-Constraint Kundenerfahrung.....                                     | 83 |
| Tabelle 12: Project-Constraint Kommunikation.....                                       | 83 |
| Tabelle 13: Project-Constraint PM-Erfahrung .....                                       | 84 |
| Tabelle 14: Project-Constraint Erfahrung mit Vorgehensmodellen .....                    | 84 |
| Tabelle 15: Project-Constraint Projektarbeit .....                                      | 84 |
| Tabelle 16: Project-Constraint Prozesskonflikte .....                                   | 84 |
| Tabelle 17: Project-Constraint Mangel an Fähigkeiten .....                              | 85 |
| Tabelle 18: Project-Constraint Führungsetage .....                                      | 85 |
| Tabelle 19: Project-Constraint Kosten der Projektorganisation.....                      | 85 |
| Tabelle 20: Project-Constraint Schulungskosten .....                                    | 85 |
| Tabelle 21: Project-Constraint Rollenverständnis .....                                  | 86 |
| Tabelle 22: Project-Constraint Kundenmitwirkungspflicht.....                            | 86 |
| Tabelle 23: Projekteigenschaften zur Bewertung eines möglichen Vorgehens..              | 97 |

## Abkürzungsverzeichnis

---

AG *Auftraggeber*

AN *Auftragnehmer*

BIC *Bank Identifier Code*

BLZ *Bankleitzahl, Bankleitzahl*

BSC *Balanced Scorecard*

KVP *Kontinuierlicher*

*Verbesserungsprozess*

LDS *Lean Development System*

LeSS *Large Scale Scrum*

PM *Projektmanagement*

PSP *Projektstrukturplan*

SAFe *Scaled Agile Framework*

SEPA *Single Euro Payments Area*

SLA *Service Level Agreement*

WiP *Work in Progress*

# 1 Einleitung

---

Mit zunehmender Bedeutung von Projekten neben dem prozessorientierten Tagesgeschäft spielt das Thema Projektcontrolling und Projektmanagement für viele Unternehmen heute eine zentrale Rolle bei ihrer wirtschaftlichen Ressourcenbetrachtung. Mittlerweile bilden die sogenannten agilen Ansätze eine Alternative zum managementgetriebenen Denkansatz, auf dem die wasserfall- oder phasenmodellbasierten „klassischen“ Vorgehensmodelle basieren. Es kann sogar der Eindruck entstehen, dass „agil“ in aller Munde ist: Während die Befürworter unbedingt agil sein wollen, weil das häufige Scheitern von Projekten zu einem Umdenken im Projektmanagement führen müsse, halten die Gegner dagegen, dass agile Projekte automatisch in einen chaotischen Zustand münden, in dem Projektmanagement und Projektcontrolling keinen Platz hätten.

Auf Basis dieses zugegebenermaßen theoretisch überspitzten Verständnisses von jeweils einer agilen und einer klassischen, spezifikationsorientierten Projektwelt untersucht diese Arbeit, welche charakteristischen Unterschiede zu einer Abgrenzung der beiden Projektwelten voneinander führen. Diese finden sich in den Vorgehensmodellen selbst, aber auch im Controlling wieder, weshalb im Vorfeld in Kapitel 2 geklärt wird, wo Controlling im Projektbereich zu finden ist, um im Anschluss daran in Kapitel 3 die Unterschiede des Projektcontrolling in den beiden Projektwelten zu analysieren.

Die Ergebnisse dieser Gegenüberstellung führen schlussendlich zu den praktischen Überlegungen des vierten Kapitels mit dem Ziel, eine Methode zu finden, um zu entscheiden, ob ein Projektvorhaben sinnvollerweise agil oder spezifikationsorientiert umgesetzt werden sollte und warum.



## 2 Grundlagen

---

### 2.1 Projekt

Unternehmen, deren Kernkompetenzen in der Herstellung bestimmter Produkte liegen, stellen ihre Geschäftsprozesse<sup>1</sup>, in welchen diese Produktion stattfindet, in den Mittelpunkt des Tagesgeschäfts. Durch das Erzeugen der jeweiligen Outputs stehen den Unternehmen somit Endprodukte für den weiteren Handel zur Verfügung. Auch bei serviceorientierten Unternehmen, die beispielsweise Supportdienste an Geschäftskunden verkaufen, stehen im Zentrum des Geschehens jene Tätigkeiten, die unmittelbar dazu führen, dass entsprechende Dienstleistungen entstehen. Mit dem Unterschied, dass Dienstleistungen nicht im Voraus produziert und gelagert werden können, sondern eben nach Bedarf dem Kunden zur Verfügung gestellt werden müssen.

Neben der täglichen Abwicklung von Geschäftsprozessen führen Unternehmen zusätzlich Projekte unterschiedlicher Art durch, so zum Beispiel Forschungs- oder Entwicklungsprojekte. Mit dem Ziel, unternehmerische Nutzen daraus zu generieren.<sup>2</sup>

Im Unterschied zur Anfertigung von Verkaufsgütern in hoher Stückzahl oder zur Abwicklung von festgelegten Serviceprozessen dient die Projektarbeit dazu, ein individuell festgelegtes Projektziel zu erreichen. Das Endprodukt eines Projekts ist also immer einzigartig, wie auch der jeweilige Projektverlauf. Nach dem Project Management Institute wird der Projektbegriff folgendermaßen definiert:

*„It's a temporary endeavor undertaken to create a unique product, service, or result.“<sup>3</sup>*

Während das Prozessmanagement genau vorgibt, welche Tätigkeiten in welcher Reihenfolge abzuarbeiten sind und wie die jeweiligen Out- und Inputs an den Schnittstellen der einzelnen Prozessschritte gestaltet sein müssen, müssen die notwendigen Tätigkeiten zur Schaffung dieses einzigartigen Produkts, Services

---

<sup>1</sup> vgl. Gabler Wirtschaftslexikon:2018a.

<sup>2</sup> vgl. Kerzner:2009, 2.

<sup>3</sup> siehe Project Management Institute, Inc:2018.

oder Ergebnisses erst noch erörtert und in sinnvolle Reihenfolge gebracht werden.

Demzufolge ist das Projekt zeitlich befristet: Es existiert ein Start- sowie ein Endzeitpunkt. Es umfasst die Gesamtheit aller Tätigkeiten, die zur Projektzielerreichung notwendig sind und setzt somit unterschiedliche Fähigkeiten voraus, die das Projektteam zum Teil mitbringen, erweitern und sogar neu erlernen muss.

Daher sind Projekte risikoreich und komplex, zumal es nicht selten vorkommt, dass Teammitglieder sich in großer räumlicher Distanz zueinander befinden.<sup>4</sup>

Je nach Art des Projektes und seiner speziellen Ziele existiert ein gewisser Innovationscharakter, was sich auch in der erläuterten Einmaligkeit widerspiegelt.

Um das Projektziel zu erreichen, wird, ähnlich wie im strategischen Management<sup>5</sup>, auf Basis einer Vision gearbeitet. Diese beschreibt den angestrebten Projekt-Output. Dabei ist es wichtig, dass alle Projektbeteiligten diese Vision kennen: So soll Einigkeit darüber geschaffen werden, wie das Endprodukt beschaffen sein soll. Damit dies der Fall ist, werden sämtliche Ziele im Voraus festgelegt.

Zwar sind Projekte demzufolge nicht mit Routinetätigkeiten gleichzusetzen, dennoch existieren Projektroutinen, die sich im jeweiligen Unternehmen im Laufe der Zeit etablieren. So können zum Beispiel in einem Entwicklungsprojekt die vom Projektteam entwickelten Software-Features erst getestet werden, nachdem diese tatsächlich in Programmcode geschrieben wurden. Auch, wenn die konkrete Vorgehensweise in den einzelnen Tätigkeitsbereichen *Analyse*, *Entwicklung* und *Test* je nach Projekt und je nach Feature unterschiedlich ist, gibt die Logik eine situationsbedingte Reihenfolge vor. So muss etwa im Falle, dass Fehler auftreten und diese anschließend erkannt werden, eine Fehlerbeseitigung stattfinden. Wird dieser Schritt einfach

---

<sup>4</sup> vgl. *Young:2007*, 11.

<sup>5</sup> Das strategische Management beschäftigt sich mit der Strategienbildung und -umsetzung eines Unternehmens. Im Zentrum dabei steht die Unternehmensvision, also die langfristige Entwicklung des Unternehmens (vgl. *Gabler Wirtschaftslexikon:2018b*).

übersprungen, kann es im weiteren Projektverlauf zu erheblichen Problemen kommen, weil bekannte Fehlerquellen ignoriert werden.

Aufgrund der zeitlichen und der finanziellen Begrenzung von Projekten müssen diese sorgfältig geplant, gesteuert und überwacht werden, um zu gewährleisten, dass die jeweiligen Projektziele erreicht werden können. Dies geschieht zum einen durch das Projektmanagement, zum anderen trägt aber auch das Projektteam zur Planung, Steuerung und Überwachung bei, je nach gewähltem Vorgehensmodell in spezifischer Ausprägung. In Kapitel 2 findet eine differenziertere Betrachtung gängiger Vorgehensmodelle statt.

Schließlich wird anhand der festgelegten Ziele und deren schlussendlicher Erreichungsgrade das Projektergebnis auf seinen Erfolg hin beurteilt.

## 2.2 Projektmanagement

Das Project Management Institute definiert Projektmanagement als

*„The application of knowledge, skills, tools, and techniques to project activities to meet the project requirements.“<sup>6</sup>*

Um diese Projektanforderungen zu erfüllen, müssen die für das Projektmanagement zentralen Größen *Leistung, Kosten* und *Zeit* sorgfältig geplant und kontinuierlich überwacht werden. Als Beispiel für eine Projektanforderung kann die Entwicklung der geplanten Features angeführt werden, mit denen das Endprodukt ausgestattet sein muss.

Zu Beginn festgelegte Projektpläne können im weiteren Projektverlauf bei Bedarf geändert werden oder an veränderte Rahmenbedingungen, wie etwa das Vorliegen von Informationen, die bei der Erstplanung noch nicht bekannt waren, angepasst werden.

Konkret umfasst das Projektmanagement die *„Gesamtheit von Führungsaufgaben, -organisation, -techniken und -mittel“*, um das Projekt von der Initiierung, Definition und Planung über die Steuerung bis hin zum Abschluss zu bringen.<sup>7</sup>

---

<sup>6</sup> siehe *Project Management Institute, Inc:2012*, Stichwort "Project Management".

<sup>7</sup> siehe *Gabler Wirtschaftslexikon:2018c*.

Demzufolge findet Projektmanagement über sämtliche Projektphasen begleitend statt. Je nach Phasen- oder Vorgehensmodell existieren unterschiedliche Phasenaufteilungen, grundsätzlich jedoch gibt es folgende Projektphasen<sup>8</sup>, die jedes Projekt in irgendeiner Form durchlaufen muss:

1. Projektauswahl
2. Projektinitiierung
3. Projektplanung
4. Projektumsetzung und -steuerung
5. Projektabschluss

Das Ziel bei der Projektauswahl ist es, aus der Menge an möglichen Projekten, die zum Wahlzeitpunkt zur Verfügung stehen, diejenigen zu identifizieren, die dem Unternehmen den größten Nutzen bringen, mit Ausnahme derjenigen Projekte, die, zum Beispiel aufgrund von rechtlichen Vorgaben, auf jeden Fall umgesetzt werden müssen. Bei der Wahl der Projekte spielen nicht nur finanzielle Überlegungen eine Rolle, denn auch nichtmonetäre Ziele können dazu führen, ein bestimmtes Projekt aufzusetzen oder auch Gründe für eine Ablehnung liefern.

In der Phase der Projektinitiierung wird das Projekt vorbereitet. Die Projektziele werden definiert, ein vorläufiger Plan für benötigte Ressourcen wird aufgesetzt und Meilensteine<sup>9</sup> werden definiert. Aus diesen Grobplanungen kann dann abgeleitet werden, welche Fachkenntnisse benötigt werden und demzufolge wird das Projektteam zusammengestellt.

Die Planungsphase dient nachfolgend der detaillierten Planung des Projekts, woraufhin ein konkreter Projektplan entsteht.

In der Phase der Projektumsetzung und -steuerung wird dieser Plan dann realisiert. Nachdem alle Arbeiten erledigt wurden, kann das Projekt in der Abschlussphase geschlossen und das entstandene Produkt vom Auftraggeber abgenommen werden.

Zwar lassen sich die Aufgaben, die in einem Projekt anfallen, in verschiedene Phasen einteilen, jedoch gibt das Vorgehensmodell vor, wie die einzelnen

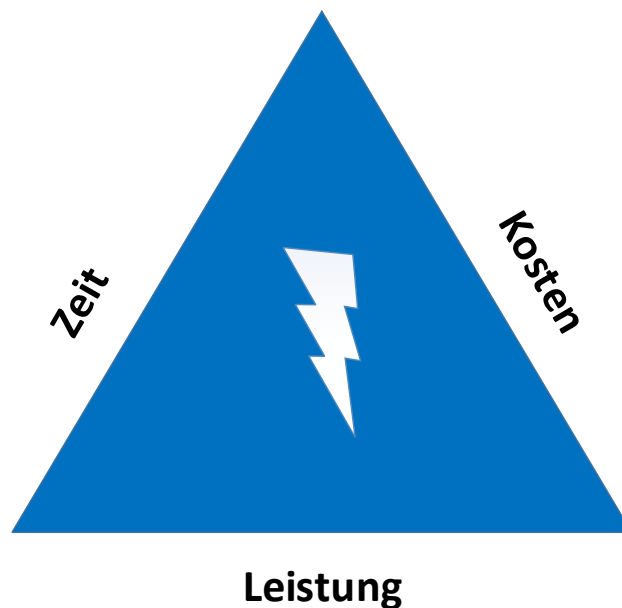
---

<sup>8</sup> vgl. *Young:2007*, 24.

<sup>9</sup> vgl. *InLoox:2018b*.

Phasen ablaufen, sodass diese sich überschneiden, durch bestimmte eintretende Ereignisse ab- oder unterbrochen oder auch wiederholt werden können. Im späteren Verlauf dieser Arbeit werden daher verschiedene Vorgehensmodelle vorgestellt und voneinander abgegrenzt.

Die zentralen Projektmanagement-Größen *Leistung*, *Kosten* und *Zeit* können im Magischen Dreieck des Projektmanagements visualisiert werden, wie *Abbildung 1* zeigt.<sup>10</sup>



*Abbildung 1: Magisches Dreieck (Quelle: Eigene Darstellung).*

Dabei umfasst der Parameter *Leistung* zum einen den Umfang der Funktionalitäten, die das im Projekt entwickelte Produkt besitzt, zum anderen aber auch deren Qualität. Sowohl die einzelnen Funktionalitäten, als auch die angestrebte Qualität des Endergebnisses werden in Absprache mit dem Kunden geplant und in Form eines Projektstrukturplanes festgehalten.

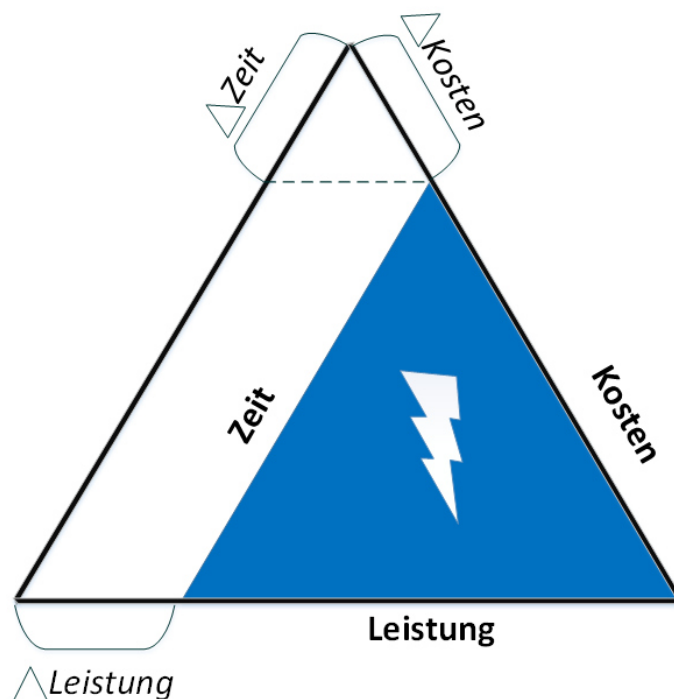
Die zeitliche Dimension beinhaltet die Projektlaufzeit sowie Termine bzw. Meilensteine. Die *Kosten* stellen das Budget dar, also die finanziellen Mittel, die dem Auftragnehmer für die Projektumsetzung zur Verfügung stehen. Anhand der jeweiligen (geplanten) Ausprägung von *Leistung*, *Zeit* und *Kosten* im Verhältnis wird das Magische Dreieck projektspezifisch definiert: Ist das Budget vorgegeben sowie ein fixer Abschlusstermin, dann muss die *Leistung* dementsprechend an den Parametern *Zeit* und *Kosten* orientiert werden.

---

<sup>10</sup> vgl. Kerzner:2009, 715f.

Das zeigt, dass *Zeit*, *Kosten* und *Leistung* in enger Beziehung zueinander stehen, sodass darauf einwirkende Maßnahmen nicht nur einzelne dieser Parameter betreffen: Soll etwa der geplante Funktionsumfang des Endprodukts im Projektverlauf erweitert werden, so kann ein erhöhter Personalbedarf die Folge sein, was sich wiederum auf die Kosten- und Zeit-Dimensionen auswirken kann.<sup>11</sup>

Dies wird in *Abbildung 2* schematisch dargestellt, wobei das ursprüngliche Magische Dreieck verzerrt wird, da ein Mehr an *Leistung* bedeutet, dass die *Kosten* steigen und gleichzeitig mehr *Zeit* benötigt wird.



*Abbildung 2: Veränderung der Parameter im Magischen Dreieck (Quelle: Eigene Darstellung in Anlehnung an Kerzner, 717).*

Das Projektmanagement trägt also die Verantwortung dafür, dass die geplante Balance zwischen *Leistung*, *Kosten* und *Zeit* geschaffen wird, wie sie im spezifischen Magischen Dreieck visualisiert werden kann. Dies bedeutet, dass die drei Größen im Verlauf des Projekts überwacht und bei Bedarf wieder in Einklang zueinander gebracht werden müssen.

Welche Projektmanagement-Methoden im Einzelnen verwendet werden, wie die Verantwortung der entsprechenden Tätigkeiten bzw. Rollen personell

---

vgl. Kerzner:2009, 716.

verteilt wird oder wie die einzelnen Projektphasen tatsächlich ausgestaltet werden, ist vom jeweiligen Kontext abhängig. Dabei spielt etwa die Projektart eine Rolle: Handelt es sich um ein Projekt, das hausintern stattfindet oder wird für einen externen Auftraggeber gearbeitet? Wird im Projekt neue Software entwickelt, bereits bestehende Software weiterentwickelt oder Standardsoftware an die Unternehmensprozesse des Auftraggebers angepasst? Wie ist der Arbeitsumfang des Projektes einzuschätzen? Wie risikoreich ist das Projekt und was für Risikoarten sind damit verbunden? Fragen wie diese müssen von den Verantwortlichen beantwortet werden, um das situationsspezifische Projektmanagement sinnvoll zu gestalten. Die Anforderungen für ein Projekt finden sich mitunter im entsprechenden Business Case, auf dem dieses Projekt fußt, wieder.<sup>12</sup>

Der eigentliche Zweck des Projektmanagements ist es sicherzustellen, dass das geplante Projektergebnis erreicht wird. Dabei müssen die Projektziele sowohl dem Projektmanagement als auch allen anderen involvierten Stakeholdern bekannt sein. Außerdem muss ein einheitliches Verständnis darüber vorherrschen, damit am Ende ein Produkt entstehen kann, welches den gewünschten Zweck erfüllt. Dabei ist ein Projektstakeholder *„eine Person, eine Gruppe oder eine Organisation, welche ein Interesse an einem beliebigen Aspekt eines Projektes hat, vom Projekt betroffen ist, auf das Projekt Einfluss nehmen kann oder in ihrem Empfinden vom Projekt betroffen ist“*.<sup>13</sup>

### 2.3 Controlling

*„Controlling ist ein funktionsübergreifendes Steuerungsinstrument, das den unternehmerischen Entscheidungs- und Steuerungsprozess durch zielgerichtete Informationener- und -verarbeitung unterstützt. Der Controller sorgt dafür, dass ein wirtschaftliches Instrumentarium zur Verfügung steht, das vor allem durch systematische Planung und der damit notwendigen Kontrolle hilft, die aufgestellten Unternehmensziele zu erreichen.*

---

<sup>12</sup> vgl. Young:2007, 80f.

<sup>13</sup> siehe Lent:2013, 80.

*Inhalt der Zielvorhaben können alle quantifizierbaren Werte des Zielsystems sein.“<sup>14</sup>*

Dieser Definition folgend ist Controlling eng verzahnt mit den Aktivitäten des Managements: Während letzteres für die Entscheidungsfindung verantwortlich ist, liefert das Controlling im Rahmen des Berichtswesens dem Management die dafür nötigen Informationen in entsprechend aufbereiteter Form. Dabei kann es sich zum Beispiel um eine grafische Darstellung ausgewählter Kennzahlen handeln, die dem Management diejenigen zentralen Informationen zur Verfügung stellt, welche in der jeweiligen Situation gebraucht werden. Das Berichtswesen ist das klassische Werkzeug des Controllings, um effektiv und effizient die vom Empfänger geforderten und benötigten Informationen zu liefern und zwar in möglichst verständlicher Form, um damit Transparenz zu schaffen und den Entscheidungsträger bei seiner Entscheidungsarbeit zu unterstützen. Dabei spielt bei der Verdichtung von Informationen und der Gestaltung der Berichte der Informationsbedarf des Empfängers eine entscheidende Rolle.<sup>15</sup> Weiterhin sucht und erkennt das Controlling Risiken und grenzt Unsicherheiten soweit wie möglich ein, mit denen das Unternehmen im Rahmen eines Risikomanagements umgehen muss. Kernfunktion des Risikomanagements ist es, Abweichungen von Unternehmenszielen zu verhindern, indem Risiken, die zu solchen Abweichungen führen, verhindert werden und im Falle des Risikoeintritts Gegenmaßnahmen ergriffen werden, um die negativen Konsequenzen bestmöglich abzuwenden.<sup>16</sup>

Neben der Transparenzfunktion, die das Controlling durch das Beschaffen und Aufbereiten von Informationen erfüllt, hat es zudem eine Steuerungsfunktion inne.

Als Werkzeuge dienen mitunter Soll-Ist-Vergleiche, um die tatsächlichen Zustände denjenigen gegenüberzustellen, die erreicht werden sollen, wie beispielsweise die Soll- und Ist-Kosten eines durchgeführten Projekts. Auf den Soll-Ist-Vergleichen aufbauend können zukunftsorientierte Schätzungen und Prognosen durchgeführt werden sowie rückblickende Analysen stattfinden.

---

<sup>14</sup> siehe *Preißler:2014*, 2.

<sup>15</sup> vgl. *Gabler Wirtschaftslexikon:2018d*.

<sup>16</sup> vgl. *Gabler Wirtschaftslexikon:2018e*.



Als Basis dieser Vergleiche dienen die vom Controlling veranlassten Messungen, die mithilfe von Kennzahlen bzw. Kennzahlensystemen umgesetzt werden. So kann zum Beispiel die Wirtschaftlichkeit des durchgeführten Projektes berechnet werden, indem die ermittelten Kennzahlenwerte, etwa der Ertrag des Projekts sowie dessen Gesamtkosten, ins Verhältnis gesetzt werden.<sup>17</sup> Eine Möglichkeit, die Wirtschaftlichkeit dieses Projekts zu berechnen, wäre also folgendermaßen:

$$\text{Ertrag/Gesamtkosten} = \text{Wirtschaftlichkeit}$$

Während zukunftsorientierte Informationen zum Beispiel in die Planung einfließen können, können Abgleiche zwischen einstmals geplanten und tatsächlich erreichten Werten dabei helfen, Maßnahmen abzuleiten, um ein Steuerungsobjekt wieder „auf Kurs“ zu bringen.

Der Controlling-Regelkreis stellt vereinfacht die vielfältigen Aufgaben des Controllings in einen Zusammenhang:

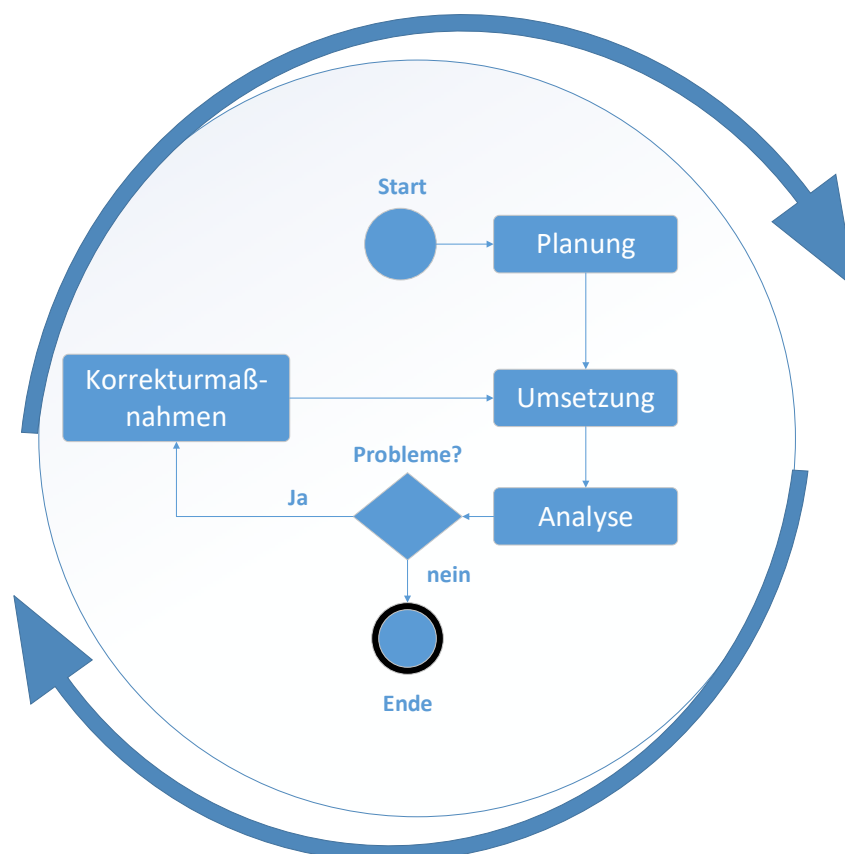


Abbildung 3: Controlling-Regelkreis (Quelle: Eigene Darstellung in Anlehnung an Kargl; Kütz:2007, 124).

<sup>17</sup> vgl. Gabler Wirtschaftslexikon:2018f.

## 2.4 Projektcontrolling

Laut der DIN 69901 handelt es sich beim Projektcontrolling um die

*„Sicherung des Erreichens der Projektziele durch: Soll-Ist-Vergleich, Feststellung der Abweichungen, Bewerten der Konsequenzen und Vorschlägen von Korrekturmaßnahmen, Mitwirkung bei der Maßnahmenplanung und Kontrolle der Durchführung.“<sup>18</sup>*

Dieselben Funktionen, die das allgemeine Controlling innehält, müssen demzufolge, angepasst auf den Rahmen eines Projekts, durch das Projektcontrolling abgedeckt werden, um das Projektmanagement bei seinen Tätigkeiten erfolgreich zu unterstützen.

Diese Unterstützung beginnt bereits mit der Initiierung des Projektes bei der Ermittlung der Wirtschaftlichkeit desselben durch eine Wirtschaftlichkeitsanalyse bzw. der Ermittlung der unterschiedlichen Nutzen des Projektergebnisses, welche das Unternehmen bei einem erfolgreichen Abschluss des Projekts erwartet. Dabei können nichtmonetäre Nutzen etwa vom Controlling durch eine Nutzwertanalyse<sup>19</sup> bestimmt werden.

Sowohl die Wirtschaftlichkeitsanalyse als auch die Nutzwertanalyse dienen dem Controlling als Werkzeug, welches bei der Unterstützung von Entscheidungen, die das Management treffen muss, zum Einsatz kommt.

Die tatsächliche Umsetzung der Controllingaktivitäten, welche im Abschnitt 1.3 beschrieben sind, hängt mitunter vom gewählten Vorgehensmodell ab, auf die im späteren Verlauf dieser Arbeit noch eingegangen wird. Projektcontrolling kann daher sehr individuell und situationsbedingt etabliert sein.

## 2.5 Agilität

Beschreibt man einen Menschen in seinem körperlichen und geistigen Tun als „agil“, so möchte man damit zum Ausdruck bringen, dass die beschriebene Person mit wachem Verstand handelt und weitestgehend die Kontrolle

---

<sup>18</sup> siehe Fiedler:2018, 1.

<sup>19</sup> vgl. Gabler Wirtschaftslexikon:2018g.

über ihre körperlichen Kräfte besitzt. So beschreibt das Adjektiv laut Duden etwas „von großer Beweglichkeit zeugend“ bzw. „regsam und wendig“.<sup>20</sup> Doch wie lässt sich der Begriff der Agilität auf die Softwareentwicklung und das Management übertragen?

### 2.5.1 Agile Softwareentwicklung

Während spezifikationsorientierte Softwareentwicklung, wie sie zum Beispiel beim Wasserfallmodell<sup>21</sup> zum Einsatz kommt, einen dokumentenlastigen Entwicklungsprozess aufweist, versuchen agile Modelle einen Gegenpol zu „planungsorientierten Methoden mit ausufernden Konzeptionsphasen“<sup>22</sup> zu bilden und zeigen in vielen Fällen Parallelen zum Lean Product Development. Im Zentrum des Lean Product Development steht das Bestreben, Aktivitäten mit keinem oder geringem Nutzen aus dem Produktionsprozess zu entfernen, um diesen zu verschlanken.

Das sogenannte Lean Development System bildet hierfür das Werkzeug:

*„LDS sind Systeme, die ebenfalls auf Prinzipien, Methoden und Tools basieren (Morgan und Liker 2006). Diese zielen auf eine schnelle und effiziente Entwicklung von Produkten und Dienstleistungen, die den Anforderungen des Kunden gerecht werden. Es sollen möglichst wenig Ressourcen verbraucht, eine hohe Qualität erreicht und wenig Kapital eingesetzt werden (Morgan und Liker 2006).“<sup>23</sup>*

Nach Dombrowski lassen sich sieben Gestaltungsprinzipien ableiten, die im Lean Product Development befolgt werden sollten:<sup>24</sup>

1. Kontinuierliche Verbesserung
2. Standardisierung
3. Fließ- und Pull-Prinzip
4. Mitarbeiterorientierung und zielorientierte Führung
5. Null-Fehler-Prinzip
6. Visuelles Management

---

<sup>20</sup> siehe *Bibliographisches Institut GmbH:2018*.

<sup>21</sup> vgl. *Hanser:2010*, 4.

<sup>22</sup> siehe *Rau:2016*, 11.

<sup>23</sup> siehe *Dombrowski:2015*, 9.

<sup>24</sup> siehe *Dombrowski:2015*, 21.

## 7. Frontloading

Auch in der Agilen Softwareentwicklung wird eine kontinuierliche Verbesserung, sowohl des Produkts, als auch der Abläufe im Projekt, angestrebt. Dies spiegelt sich mitunter in der zyklischen Vorgehensweise wider, wobei das Produkt von Arbeitszyklus zu Arbeitszyklus um Funktionen erweitert wird und diese Funktionen in Rücksprache mit dem Kunden verbessert werden sollen. Die Abläufe innerhalb des Teams werden ebenfalls mittels Retrospektiven auf den Prüfstand gestellt und gegebenenfalls verbessert. Auch die verkürzten Planungshorizonte, die in der Agilen Softwareentwicklung zum Einsatz kommen, sollen dafür sorgen, unnötige Arbeiten zu vermindern, indem immer nur für den aktuellen Arbeitszyklus detailliert vorausgeplant wird. Neben diesem Prinzip finden sich noch einige andere in agilen Ansätzen wieder, so etwa das Pull-Prinzip oder die Bedeutung der Individuen (in dem Fall Mitarbeiter) in einem Projektteam.

Geht der eigentlichen Softwareentwicklung in älteren Entwicklungsprozessen eine sehr detaillierte Planung voraus, welche die konkreten Bestandteile des zu entwickelnden Produkts bereits im Vorfeld möglichst genau zu beschreiben sucht, grenzt sich die Agile Softwareentwicklung von diesem Vorgehen durch die sogenannten *Agilen Werte* und die *Agilen Prinzipien* ab, welche im *Agile Manifesto* aus dem Jahr 2001 festgehalten sind.

Im Zentrum stehen dabei die folgenden vier Aussagen, wobei jede einzelne unterschiedliche Werte zueinander in Relation setzt:<sup>25</sup>

- „**Individuen und Interaktionen** *mehr als* Prozesse und Werkzeuge“
- „**Funktionierende Software** *mehr als* umfassende Dokumentation“
- „**Zusammenarbeit mit dem Kunden** *mehr als* Vertragsverhandlung“
- „**Reagieren auf Veränderung** *mehr als* das Befolgen eines Plans“

Diese Werte-Beziehungen sind relativ und können je nach Projekt unterschiedlich ausgelegt werden.

Sind Individuen und Interaktionen wichtiger als Prozesse und Werkzeuge, so deutet dies darauf hin, dass die Teammitglieder und ihre individuellen

---

<sup>25</sup> siehe *the Agile Manifesto authors:2018a*.

Kenntnisse und Fähigkeiten, sowie die Zusammenarbeit untereinander eine große Bedeutung erhalten. So kann bei der Umsetzung des Projekts zum Beispiel auf das Fachwissen eines Entwicklers zurückgegriffen und Technologien verwendet werden, die dieser beherrscht anstatt vielleicht auf in die Jahre gekommene Werkzeuge zu setzen.

Dass funktionierende Software einen höheren Stellenwert erhält als eine umfassende Dokumentation findet sich in vielen Methoden und Tools wieder, die bei der Agilen Softwareentwicklung zum Einsatz kommen: Statt mittels umfangreicher Lasten- und Pflichtenhefte die Funktionen einer geplanten Software zu beschreiben, werden zum Beispiel User Stories<sup>26</sup> verwendet – eine Möglichkeit, Funktionalitäten in einfachen Beschreibungen aus Anwendersicht zu dokumentieren. Die verkürzten Planungshorizonte der agilen Denkweise haben zur Folge, dass am Ende jedes Planungsabschnitts, auch Sprint genannt, funktionierende Software entstanden ist, die dem Kunden zur Verfügung gestellt werden kann. Das heißt, dass bereits kurz nach Projektstart lauffähige Software entsteht, die im fortschreitenden Verlauf erweitert und verbessert wird.

Die Zusammenarbeit mit dem Kunden wird als wichtiger eingestuft als Vertragsverhandlungen, was nahelegt, dass das Vertrauen zwischen Auftraggeber und Auftragnehmer Priorität hat gegenüber dem strengen Befolgen geplanter Vorgehensweisen. Wenn dieses Vertrauen gefährdet ist, sollte eine Möglichkeit gefunden werden, die Vorgehensweise im gegenseitigen Einverständnis anzupassen.

Das Reagieren auf Veränderungen soll wichtiger sein, als das genaue Befolgen von Plänen, was bedeutet, dass eine grundsätzliche Bereitschaft für Veränderungen vorausgesetzt wird. Diese sollen nicht mit einer ablehnenden Haltung behandelt, sondern offen begrüßt werden.

---

<sup>26</sup> vgl. Goll; Hommel:2015, 37f.

Auf diesen vier Kernaussagen aufbauend existieren zusätzlich die 12 *Agilen Prinzipien*:

Tabelle 1: Die 12 Agilen Prinzipien<sup>27</sup>

| PRINZIPIEN | BESCHREIBUNG  |
|------------|---|
| 1          | „Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen.“                                  |
| 2          | „Heisse Anforderungsänderungen selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.“                     |
| 3          | „Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.“  |
| 4          | „Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.“  |
| 5          | „Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.“ |
| 6          | „Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.“       |
| 7          | „Funktionierende Software ist das wichtigste Fortschrittsmaß.“  |
| 8          | „Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten ein gleichmäßiges Tempo auf unbegrenzte Zeit halten können.“       |
| 9          | „Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.“   |
| 10         | „Einfachheit -- die Kunst, die Menge nicht getaner Arbeit zu maximieren -- ist essenziell.“   |
| 11         | „Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams.“  |
| 12         | „In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an.“   |

Diese Werte legen nahe, dass das Grundverständnis darüber, wie ein Softwareentwicklungsprojekt durchgeführt werden sollte, um zum Erfolg zu führen, im klaren Kontrast steht zur Vorgehensweise bei der klassischen Softwareentwicklung. Nicht Pflichten- und Lastenhefte stehen im Mittelpunkt und sollen nach Plan abgearbeitet werden, sondern das gemeinsame

<sup>27</sup> siehe *the Agile Manifesto authors:2018b*.

Verständnis soll dazu führen, dass am Ende Software entsteht, die die Wünsche und Bedürfnisse des Kunden erfüllen kann, auch, wenn diese zu Projektbeginn gar nicht oder nur teilweise sichtbar sind oder sich im Lauf des Projekts aus unterschiedlichsten Gründen verändern. Allen Beteiligten eines agilen Softwareprojekts sollte von Beginn an klar sein, dass Änderungen, z. B. den Funktionsumfang des angestrebten Produkts betreffend, nicht notwendiges Übel, sondern essentiell sind, um erfolgreich mit dem agilen Ansatz fahren zu können. Die Zusammenarbeit mit allen Stakeholdern ist dabei von besonderer Bedeutung, damit die anvisierten Projektnutzen realisiert werden können.

Agile Softwareentwicklung beansprucht eine gewisse Selbstorganisation des Entwicklungsteams, die ihren Ausdruck in agilen Methoden und Vorgehensmodellen findet. Da Kreativität und Flexibilität im Zentrum des Entwicklungsprozesses stehen, muss das Team mit genügend organisatorischer Kompetenz, also der Fähigkeit, die eigene Arbeit im Hinblick auf die Projektziele zu organisieren, ausgestattet sein, um auf entscheidende Veränderungen, wie zum Beispiel technologische Neuerungen, aus denen der anvisierte Nutzen für das Projekt generiert werden kann, schnell genug reagieren zu können. Daher werden in agilen Projekten Iterationen<sup>28</sup> durchlaufen, wobei das Ziel-Produkt Stück für Stück Gestalt annehmen soll. Änderungen sind dabei stets fester Bestandteil der Vorgehensweise: So werden etwa Entscheidungen über Produktfunktionalitäten, die für die Anfangsphase des Projekts keine Rolle spielen, bewusst verschoben.

### 2.5.2 Agiles Management

Auch abseits des Projektkontexts gibt es Gründe, die Unternehmen dazu bewegen, mehr Agilität in das eigene Management zu integrieren. Dabei soll u.a. Flexibilität geschaffen werden, um das nötige Reaktionsvermögen zu besitzen, das gebraucht wird, sich ändernden Bedingungen bei Bedarf zu stellen. So soll die Agilität das Unternehmen in seiner Wettbewerbssituation stets dazu befähigen, wirtschaftlich zu handeln und sich verändernden Marktbedingungen nicht verschließen zu müssen. Kundenorientierung muss

---

<sup>28</sup> In Kapitel 3 wird das Konzept der iterativen Vorgehensweise näher erläutert.

demzufolge ebenso, wie in der agilen Softwareentwicklung, in den Entscheidungen und Handlungen des Managements entsprechend gewürdigt werden. Sie bildet eine von mehreren Dimensionen, wie sie zum Beispiel von André Häusling im nachfolgend verwiesenen Artikel beschrieben werden.<sup>29</sup>

Auch die Werte-Relationen aus dem Agilen Manifest können von Unternehmen herangezogen werden, um sie auf ein agiles Management zu übertragen.<sup>30</sup> Zum Beispiel dadurch, dass den einzelnen Individuen mehr Handlungsfreiheiten eingeräumt werden, also eine gewisse Offenheit geschaffen wird, sodass Mitarbeiter im Management Kreativität etwa bei Entscheidungsprozessen und Maßnahmenumsetzungen zeigen können. Auch das Zusammenarbeiten mit den vom Management betroffenen Abteilungen kann einen höheren Stellenwert erhalten, indem das strenge Befolgen von Plänen einer flexibleren Planungsweise weicht, in welcher z. B. auch die Stimmen und damit das Wissen der Betroffenen Eingang in die Planung finden.

---

<sup>29</sup> vgl. *Häusling (2016)*.

<sup>30</sup> siehe *Muth; Valentini:2017*, 1ff.



### 3 Controlling im Projektbereich

---

Projektmanagement und Projektcontrolling kann in der Praxis in unterschiedlichen Unternehmenskontexten stattfinden, wobei diese Arbeit zwei Ebenen differenziert betrachtet:

1. Projektmanagement und Controlling im Rahmen eines Einzelprojekts
2. Projektmanagement und Controlling im Rahmen einer Mehrprojektsituation (Multiprojektmanagement)

Im nachfolgenden Teil dieses Kapitels wird zunächst das Konzept der BSC erläutert, um im Anschluss daran die hiervon abgeleitete Projekt-Scorecard zu betrachten. Diese soll in den weiterführenden Abschnitten zum Einzelprojekt und der Mehrprojektsituation zum Einsatz kommen, um die unterschiedlichen Anforderungen des Projektmanagements und insbesondere des Controllings dieser Betrachtungsebenen darzustellen und gleichzeitig die Vereinbarkeit beider Sichtweisen mittels der Projekt-Scorecard herauszustellen.

Eine Unterteilung in Einzelprojekt- und Multiprojektsichtweise erscheint an dieser Stelle sinnvoll, denn während innerhalb eines laufenden Projekts der Fokus von Projektmanagement- und Controllingseite auf der erfolgreichen Abwicklung desselben gerichtet ist, wobei der Projekterfolg sich an den im Vorfeld definierten Projektzielen messen lassen muss, sollte es gleichzeitig eine Instanz geben, die eine übergeordnete Steuerungsfunktion innehält. Das Multiprojektmanagement sorgt dafür, dass die parallel laufenden Projekte auf einen möglichst optimalen Gesamtnutzen für das Unternehmen ausgerichtet werden.<sup>31</sup>

#### 3.1 Balanced Scorecard & Projekt-Scorecard

Allgemein gilt bezüglich des Controllings im Projektbereich, dass dieses mit dem übergeordneten Unternehmenscontrolling und dessen (strategischen) Zielen harmonisieren muss. So sollten beispielsweise Projekte, die zu wenig Bedeutung für das Unternehmen haben, erst gar nicht in den Projektauswahlprozess gelangen. Stattdessen sollten Projektvorschläge nach Möglichkeit

---

<sup>31</sup> vgl. *Seidl:2011*, 9.

mit der strategischen Ausrichtung des Unternehmens in Einklang stehen, sodass aus den Projektvorschlägen „ein Projektportfolio von hoher Effektivität für die Unternehmensziele entsteht“.<sup>32</sup> Davon ausgenommen sind jedoch Projekte, die durchgeführt werden müssen, zum Beispiel aufgrund von gesetzlichen Vorgaben. Muss-Projekte können zwar von geringer Bedeutung sein im Hinblick auf die Unternehmensziele, müssen aber dennoch durchgeführt werden. Ein Beispiel hierfür stellt die Umstellung im bargeldlosen Zahlungsverkehr auf das SEPA-Verfahren<sup>33</sup> dar, das die Zahlung mittels BIC und BLZ durch die „europäische Einzugsermächtigung“<sup>34</sup> seit Februar 2014 ersetzt, „um den europäischen Binnenmarkt im Zahlungsverkehr zu fördern“.<sup>35</sup>

Derartige Muss-Projekte stehen also den strategisch wichtigen Projekten gegenüber. Sämtliche Nutzen, die durch Umsetzung aller Kann-Projekte im Portfolio erwartet werden und die den Großteil des Projektportfolios ausmachen sollten, sollen dazu beitragen, die anvisierten strategischen Ziele zu erreichen.

Im Unternehmenscontrolling „laufen die Daten des Rechnungswesen und anderer Quellen zusammen“<sup>36</sup>, es fließen also die für ein übergreifendes Controlling relevanten Daten aus einzelnen Unternehmensbereichen an der Stelle zusammen, an der eine übergeordnete Betrachtung dieser Daten vorgenommen wird, um den Erfolg des unternehmerischen Handelns zu sichern.

Um eine Vereinbarkeit des Projektcontrollings mit dem Unternehmenscontrolling zu schaffen, wird eine Schnittstelle zwischen diesen beiden Bereichen benötigt.<sup>37</sup> So können mittels geeignetem Reporting Informationen aus dem Projektkontext in das Unternehmenscontrolling einfließen.

Auch muss ein Informationsfluss von außen ins Projekt möglich sein, damit das interne Projektcontrolling eine Informationsgrundlage bilden kann, auf der Entscheidungen getroffen werden, die nicht nur die Projektziele,

---

<sup>32</sup> siehe Kargl; Kütz:2007, 22.

<sup>33</sup> vgl. Fiedler:2016, 4.

<sup>34</sup> siehe Bundesverband deutscher Banken e. V.:2012, 13.

<sup>35</sup> siehe Bundesverband deutscher Banken e. V.:2012, 12.

<sup>36</sup> siehe Gabler Wirtschaftslexikon:2018h.

<sup>37</sup> vgl. Abb. 4.

sondern auch die strategischen Ziele mitberücksichtigen. Diese Schnittstelle kann mithilfe einer Projekt-Scorecard realisiert werden.

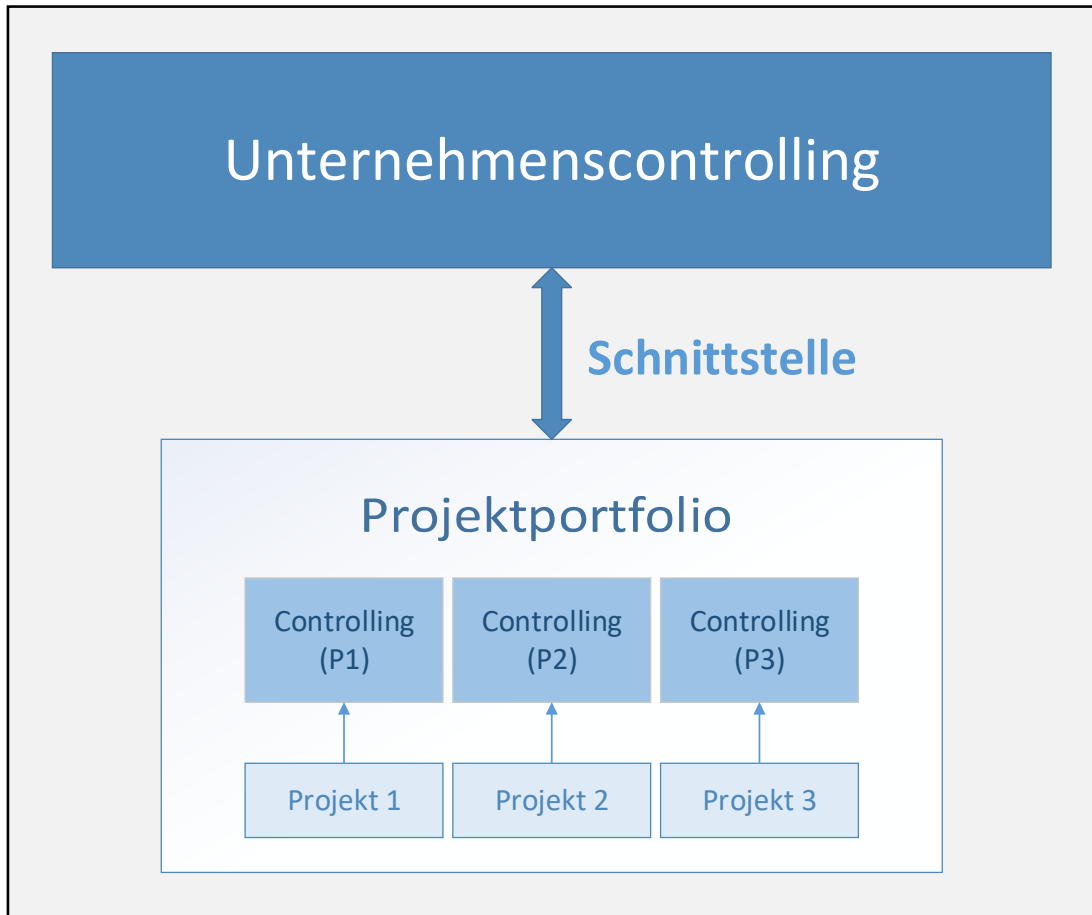


Abbildung 4: Schnittstelle Projekt- & Unternehmenscontrolling (Quelle: Eigene Darstellung).

### 3.1.1 Balanced Scorecard

Die Balanced Scorecard ist ein Instrument, mit dem operatives und strategisches Denken im Unternehmen in Einklang gebracht werden soll. Dabei soll die oftmals bemängelte Lücke zwischen der strategischen Zielsetzung und der Umsetzung der Unternehmensstrategie geschlossen werden.<sup>38</sup>

Die klassische BSC stammt von Kaplan und Norton, welche das Konzept in den 90er Jahren erdachten, als die bis dato eingesetzten Kennzahlensysteme „traditionell auf Finanzaspekte fokussiert und stark vergangenheitsorientiert ausgerichtet“<sup>39</sup> waren. Die BSC bildet somit ein Kennzahlensystem

<sup>38</sup> vgl. Seidl:2011.

<sup>39</sup> siehe Gadatsch:2016, 10.

für das strategische Management, wobei üblicherweise vier unterschiedliche Perspektiven<sup>40</sup> betrachtet werden:

1. Finanzen
2. Kunden
3. Prozesse
4. Lernen und Entwicklung

Die Verwendung von Kennzahlen aus diesen vier Bereichen soll im Management zu einer ausgewogenen Informationsbasis führen, sodass letztlich keine eingeschränkte Sichtweise durch eine starre Fixierung auf rein finanzielle Kennzahlen besteht. Mit einer umfassenderen Betrachtung aus den unterschiedlichen Perspektiven ergeben sich für das Unternehmen zahlreiche Ansatzpunkte für Verbesserungen, um die Ausrichtung der unternehmerischen Aktivitäten strategisch optimal auszurichten.

Während die Dimension *Finanzen* dazu verwendet wird, klassische finanzielle Ziele abzubilden, sollen die übrigen Dimensionen nach Kaplan und Norton Fragen nach den Gesichtspunkten *Kunden*, z. B. „*Wie hoch ist unsere Kundenzufriedenheit?*“, *Prozesse*, z. B. „*Welches Verbesserungspotenzial für die Durchlaufgeschwindigkeiten unserer Prozesse gibt es?*“, sowie *Lernen und Entwicklung*, z. B. „*Wo gibt es Qualifizierungsmängel?*“ beantworten.<sup>41</sup>

Jedoch macht es Sinn, die Dimensionen und ihre zugehörigen Kennzahlen so zu wählen, dass die jeweilige BSC dem individuellen Informationsbedarf gerecht werden kann. Man ist also nicht auf die klassischen vier Dimensionen beschränkt, sondern kann das Konzept der BSC der eigenen Unternehmenssituation anpassen.<sup>42</sup>

Die Wahl der jeweiligen Dimensionen sowie der dazugehörigen Kennzahlen geschieht mithilfe der Analyse von Ursache-Wirkungs-Beziehungen, sodass die Perspektiven der BSC nicht unabhängig voneinander betrachtet werden, sondern vor dem Hintergrund, dass die dahinterstehenden Kennzahlen Einfluss aufeinander haben.<sup>43</sup> So können sich beispielsweise bessere Durchlaufzeiten

---

<sup>40</sup> vgl. *Gabler Wirtschaftslexikon:2018i*.

<sup>41</sup> vgl. *Gadatsch:2016*, 11f.

<sup>42</sup> vgl. *Gadatsch:2016*, 11.

<sup>43</sup> vgl. *Gadatsch:2016*, 11.

positiv auf die Prozesskosten auswirken oder eine gute Kundenzufriedenheit könnte in Zusammenhang mit wirtschaftlichen Kennzahlen gebracht werden.

Anhand der gewünschten Ziele, die in den Dimensionen der BSC abgebildet werden, lassen sich konkrete Maßnahmen ableiten, um die erfassten Ist-Werte der Kennzahlen den Zielwerten anzunähern oder sie sogar zu erreichen.

### 3.1.2 Projekt-Scorecard

Bei der Projekt-Scorecard handelt es sich um eine Adaption der Balanced Scorecard, zugeschnitten auf die Bedürfnisse zur Steuerung eines Projekts.

Sie liefert ein Werkzeug, welches sowohl zur Steuerung von Einzelprojekten, als auch zur Multiprojektsteuerung verwendet werden kann und zugleich in beiden Kontexten ein Mittel zur Orientierung des Projektcontrollings und -managements an der übergeordneten Unternehmensstrategie liefert.

Daher kann die Projekt-Scorecard sowohl im Rahmen des strategischen Managements, als auch im operativen Geschehen des Projektmanagements und des Projektcontrollings Verwendung finden.

Dabei müssen die einzelnen Perspektiven der Scorecard und die zugehörigen Kennzahlen so gewählt werden, dass der Projekterfolg im Sinne der jeweiligen Betrachtungsweise messbar gemacht werden kann. Das Konzept muss jedoch nicht auf vier Perspektiven beschränkt bleiben sondern kann, je nach Bedarf, um weitere Sichtweisen ergänzt werden.

Wie auch bei der BSC werden für die einzelnen Perspektiven der Projekt-Scorecard Ziele festgelegt, deren Erreichungsgrad anhand von den in der Scorecard festgelegten Kennzahlen ermittelt werden soll.

Damit die Projekt-Scorecard sowohl auf Einzelprojekt- als auch auf Multiprojektenebene sinnvoll eingesetzt werden kann, müssen die Zielsetzungen und ihre zugehörigen Kennzahlen an die jeweiligen Projektmanagement- und Controlling-Ziele angepasst werden.

Idealerweise ermöglicht die Projekt-Scorecard dennoch eine Orientierung an der unternehmensweit geltenden BSC, sofern vorhanden, sodass das Projektgeschäft sich hinreichend an der Unternehmensstrategie, die ja in der BSC mithilfe von Kennzahlen abgebildet werden soll, orientiert.

Der Aufbau einer spezifischen Projekt-Scorecard sollte sich also daran orientieren, welche Projektgegenstände zum einen dokumentiert und messbar gemacht werden sollen, zum anderen aber auch gesteuert werden können und sollen. Das Design von Kennzahlensystemen ist daher keine triviale Aufgabe, die mit Muster-Systemen zu lösen ist. Die in dieser Arbeit vorgestellten Scorecards sind daher als Beispiele zur Veranschaulichung des Konzepts der Projekt-Scorecard zu verstehen, welche in dieser Form nicht eins zu eins unkritisch übernommen und in der Praxis angewendet werden können.

### 3.2 Einzelprojekt

Im Zentrum eines Einzelprojekts stehen die konkreten Projektziele, die erreicht werden sollen. Das Projektmanagement plant, steuert und überwacht hierzu die Parameter *Kosten*, *Leistung* und *Zeit* gemäß dem auf das Projekt bezogenen Magischen Dreieck und wird vom Projektcontrolling dabei entsprechend unterstützt. Demzufolge werden die angewandten Methoden zur Planung, Steuerung und Überwachung an den Lebenszyklusphasen des Projektes, also den Projektphasen, ausgerichtet.<sup>44</sup>

Bei der Bildung der Projekt-Scorecard für das Einzelprojektcontrolling gilt es vor allem, die Parameter *Leistung*, *Finanzen*<sup>45</sup> und *Zeit* in Kennzahlen abzubilden, um den Projektfortschritt damit möglichst gut erfassen zu können. Ist der aktuelle Projektstatus messbar, kann dies vom Projektcontrolling und Projektmanagement für eine effektive Steuerung genutzt werden, da sie etwaige Engpässe und Risiken kennen.

Zudem können die in der Scorecard verdichteten Informationen zum Zwecke des Reportings verwendet werden, um damit eine wirtschaftliche Begründung für die Weiterführung des entsprechenden Projektes an die übergeordneten Entscheidungsträger liefern zu können.

Je nach gewähltem Vorgehensmodell werden unterschiedliche Parameter verfolgt, sodass die Dimensionen der Projekt-Scorecard davon abhängen,

---

<sup>44</sup> siehe *Fiedler:2016*, 12.

<sup>45</sup> Statt *Kosten* aus dem Magischen Dreieck zu übernehmen wurde der Parameter *Finanzen* gewählt, weil dies in Anbetracht möglicher wichtiger Kennzahlen, wie z. B. der *Rentabilitätserwartung*, angemessen erscheint.

was gemessen werden muss. Da bei klassischen Vorgehensmodellen alle drei Parameter des magischen Dreiecks im Zentrum stehen, in der agilen Welt jedoch üblicherweise mit festgelegten Zeitfenstern, sogenannten *Timeboxen*, gearbeitet wird, sodass die *Zeit*-Dimension nicht explizit verfolgt werden muss, sind die Kennzahlen von Projekt-Scorecards eben sehr situationsspezifisch festzulegen. Den Unterschieden zwischen klassischen und agilen Ansätzen widmet sich Abschnitt 3.4. Dort wird auch das Konzept der *Timebox* näher erläutert.

Zu den Dimensionen *Leistung*, *Finanzen* und *Zeit* können zudem weitere kritische, den Erfolg des Projekts betreffende Attribute hinzugefügt werden, um diese zu überwachen und ggf. steuernd einzugreifen. Besonders agile Projekte sind kritisch von der Kommunikation zwischen Projekt-Kunden und Projekt-Team abhängig, sodass in einem solchen Projekt die *Kommunikation* eine weitere Dimension bilden könnte. Auch eine *Kunden*-Sichtweise könnte auf Ebene des Einzelprojekts hilfreich sein.

Eine Projekt-Scorecard in agilem Kontext könnte also z. B. folgende Fragen behandeln:

- Werden zum Projektabschluss alle Kundenanforderungen erfüllt sein?<sup>46</sup>
- Ist das Projekt teurer als erwartet?<sup>47</sup>
- Funktioniert die Zusammenarbeit zwischen Team und Kunden?<sup>48</sup>
- Findet ein permanenter Informationsaustausch im Team statt?<sup>49</sup>

---

<sup>46</sup> vgl. *Tabelle 2*, Leistung

<sup>47</sup> vgl. *Tabelle 2*, Finanzen

<sup>48</sup> vgl. *Tabelle 2*, Kunde

<sup>49</sup> vgl. *Tabelle 2*, Kommunikation

### 3.2.1 Beispielhafte Projekt-Scorecard

Tabelle 2: Projekt-Scorecard vom 01.03.2018 (Quelle: Eigene Darstellung in Anlehnung an Fiedler:2016, 60)

| PERSPEKTIVE          | ZIEL                         | KENNZAHL                               | ISTWERT           | ZIELWERT         |
|----------------------|------------------------------|--|-------------------|------------------|
| <i>Leistung</i>      | Hohe Velocity                | Story Points (SP) / Iteration          | 13 SP / Iteration | 25 SP/ Iteration |
|                      | Geringe Fehlerquote          | Nicht best. Tests                      | 10 %              | 0 %              |
|                      | Gute Testabdeckung           | Testfälle / User Stories               | 3 / 5             | 5 / 5            |
|                      | Erfüllte Systemanforderungen | Abgeschlossene Product Backlog Items   | 10                | 80               |
| <i>Finanzen</i>      | Rentabilitätserwartung       | Gewinn / Kapital                       | 70%               | 150%             |
|                      | Budgetausschöpfung           | Budgetaus-schöpfungsgrad in %          | 25 %              | 20 %             |
| <i>Kunde</i>         | Hohe Kundenzufriedenheit     | Eskalations-meetings                   | 1                 | 0                |
|                      | Abgenommene Inkremente?      | Releases                               | 2                 | 2                |
| <i>Kommunikation</i> | Meeting-Treue                | gehaltene Meetings / geplante Meetings | 5 / 6 Meetings    | 6 / 6 Meetings   |

### 3.2.2 Vorteile der Projekt-Scorecard

Der Einsatz einer solchen Projekt-Scorecard auf Einzelprojekt-Ebene könnte folgende Vorteile mit sich bringen:

1. *Übersicht des Projektfortschritts:* Sind alle wichtigen Kennzahlen zum Projektfortschritt abgebildet, kann dieser durch kontinuierliche Aktualisierung stets „auf einen Blick“ überprüft werden. Damit ist die Projekt-Scorecard besonders in agilen Projekten gut einsetzbar, um sie z. B. an hochfrequentierten Orten des Projektgeschehens gut sichtbar zu platzieren (etwa am Treffpunkt des Daily Standup Meetings<sup>50</sup>).
2. *Erleichterung des Reportings:* Im Sinne der Wiederverwendbarkeit könnte die Projekt-Scorecard im Berichtswesen zum Einsatz kommen, um dem Empfänger (z. B. dem übergeordneten Programmcontrolling) alle für ihn

<sup>50</sup> Das *Daily Standup* ist ein tägliches kurzes Meeting, um den Projektstatus und Probleme zu erfassen. In Abschnitt 3.4 wird dieses Konzept anhand des *Daily Scrum* beschrieben.



relevanten Informationen in leicht verständlicher Form zukommen zu lassen.

3. *Ursache-Wirkung*: Bei Konflikten mit nicht erreichten Zielwerten könnte eine Analyse der Scorecard unter Berücksichtigung der bekannten und im Vorfeld analysierten Ursache-Wirkungs-Beziehungen Zusammenhänge aufdecken.
4. *Wiederverwendbarkeit*: Zwar ist jedes Projekt per definitionem einzigartig, dennoch lassen sich ähnliche Projekte bis zu einem gewissen Grad miteinander vergleichen. Hier könnten Projekt-Scorecards von bereits abgeschlossenen Projekten zum Beispiel im Projektauswahlprozess helfen, wenn zwischen mehreren ähnlichen Projekten entschieden werden muss. Auch zu Schulungs- und Lehrzwecken, z. B. in einem firmeninternen Wiki, könnten Projekt-Scorecards als Beispiele für gute oder weniger gute Projekte verwendet werden, etwa in Form von Best Practices/Worst Practices<sup>51</sup> und Lessons Learned.

### 3.3 Multiprojektmanagement

Die übergeordnete Planung, Steuerung und Überwachung eines Projektbündels, wie etwa ein Projektprogramm oder ein Projektportfolio, wird durch Multiprojektmanagement umgesetzt.<sup>52</sup>

Bei einem Projektprogramm handelt es sich um ein Bündel von Projekten, welche in einem engen Zusammenhang zueinander stehen: Die einzelnen Projekte des Programms streben gemeinsame Ziele an, die nur im Bündel erreicht werden können.<sup>53</sup> Dies bedeutet, dass der Erfolg bzw. Misserfolg eines einzelnen Projekts im Programm sich in hohem Maße auf das gesamte Projektbündel auswirkt, was zum Beispiel in einer Chancen-Risiken-Betrachtung, bei der die Chancen den Risiken gegenübergestellt werden (z. B. als Teil einer SWOT-Analyse<sup>54</sup>), sichtbar wird.

---

<sup>51</sup> Während eine Best Practice die beste (bekannte) Methode beschreibt, um ein bestimmtes Problem zu lösen, bezeichnet eine Worst Practice das genaue Gegenteil davon (vgl. *InLoox GmbH:2018d*).

<sup>52</sup> vgl. *Seidl:2011*, 9.

<sup>53</sup> vgl. *Seidl:2011*, 10.

<sup>54</sup> vgl. *Gabler Wirtschaftslexikon:2018j*.

Aus diesem Grund weisen die Einzelprojekte eines Projektprogramms starke Abhängigkeiten untereinander auf. Eine derartige Abhängigkeit besteht zum Beispiel, wenn *Projekt B* Ergebnisse benötigt, die in *Projekt A* entstehen. Oftmals greifen Projekte eines Programms zudem auf dieselben Ressourcen zu, zum Beispiel dieselben Mitarbeiter. Um Engpässe zu vermeiden, müssen die einzelnen Projektplanungen dementsprechend aufeinander abgestimmt sein. Da sich allerdings nicht alle Abhängigkeiten bereits im Vorfeld der Projektumsetzungen feststellen lassen, sondern sich z. T. erst im weiteren Verlauf herausbilden, muss das Multiprojektmanagement dies im kontinuierlichen, übergeordneten Controlling berücksichtigen.<sup>55</sup>

Nachdem alle Projekte des Programms abgeschlossen sind, gilt auch das Projektprogramm als abgeschlossen.

Auch unter einem Projektportfolio versteht man eine Gruppe von Einzelprojekten, die auf übergeordneter Ebene einer gemeinsamen Koordination unterliegen.<sup>56</sup>

Anders als beim Projektprogramm ist das Projektportfolio jedoch nicht zwangsweise zeitlich begrenzt: sobald Kapazitäten für neue Projekte frei werden, können diese ins Portfolio aufgenommen werden. Diese Abgrenzung findet sich z. B. nach Seidl<sup>57</sup> in einer Aufteilung nach permanenten (Projektportfolio-) und zeitlich befristeten (Projektprogramm-)Aufgaben wieder.

Ressourcen sind auch im Projektportfolio begrenzt, weswegen beim Multiprojektmanagement darauf geachtet werden muss, dass Engpässe verhindert werden.

Unterschiede in der Herangehensweise von klassischem Projektmanagement und Multiprojektmanagement ergeben sich vor allem aus den verschiedenen Perspektiven: Während das Projektmanagement direkt im Projektgeschehen agiert und sich dementsprechend von innen, also dem Projektkern, nach außen, dem nahen Umfeld des Projektes, richtet (Bottom-Up-Ansatz), verschafft sich

---

<sup>55</sup> vgl. Seidl:2011, 139.

<sup>56</sup> vgl. Broy; Kuhrmann:2013, 28.

<sup>57</sup> vgl. Seidl:2011, 149.

das Multiprojektmanagement einen Überblick über mehrere gebündelte Einzelprojekte und handelt daher nach einem Top-Down-Prinzip.<sup>58</sup>

Diese Top-Down-Sichtweise ist nötig, da gleichzeitig laufende Projekte auf dieselben Ressourcen zugreifen können, z. B. Mitarbeiter, und dies sorgfältig koordiniert werden muss<sup>59</sup>, damit keine Ressourcenengpässe entstehen, die dann zu Verzögerungen in einzelnen Projekten führen.

Das Multiprojektmanagement schafft somit einerseits eine projektübergreifende Koordination und kann damit im Sinne strategischer Unternehmensziele handeln, andererseits kann es aber auch dazu beitragen, das Projektmanagement der einzelnen Projekte zu entlasten.

Steht beim Projektmanagement des Einzelprojekts vor allem die Fortschrittsüberwachung innerhalb des Projekts im Fokus, so müssen aus der Multiprojekt-Sichtweise die Fortschritte mehrerer parallel laufender Projekte im Blick behalten werden. Die Scorecard aus der Multiprojekt-Sicht könnte dementsprechend dazu genutzt werden, um zeitliche und inhaltliche Abhängigkeiten unter den einzelnen Projekten zu überwachen. Zum Beispiel führen Verzögerungen einzelner Projekte zur Verzögerung des Programmabschlusses.

Auch eine *Kommunikations*-Dimension kann sinnvoll sein, wenn etwa ein Projekt-Programm mit kritisch voneinander abhängigen Einzelprojekten kontrolliert werden muss. Wenn beispielsweise mehrere Projekte auf dieselben Mitarbeiter zugreifen und zeitliche Verzögerungen auftreten, können Risiken mithilfe einer entsprechenden Kommunikationsschnittstelle zwischen den Einzelprojekten verringert werden, da diese eine reibungslose Kommunikation und damit eine bessere Organisation erlauben sollte.

Auf der Multiprojekt-Ebene könnte eine Projekt-Scorecard also z. B. helfen, folgende Fragen zu beantworten:

- Liegt *Projekt A* in der Zeit, um seine Projektergebnisse pünktlich an *Projekt B* übergeben zu können?<sup>60</sup>

---

<sup>58</sup> vgl. Seidl, 2.

<sup>59</sup> vgl. Kargl; Kütz:2007, 56.

<sup>60</sup> vgl. Tabelle 3, Zeit

- Ist *Projekt A* im Budget?<sup>61</sup>
- Welche Leistungen von *Projekt A* werden in anderen Projekten benötigt?<sup>62</sup>
- (Wo) Gibt es Konflikte zwischen den einzelnen Projekten?<sup>63</sup>

### 3.3.1 Beispielhafte Projekt-Scorecard

Während auf Einzelprojekt-Ebene projektinterne Kennzahlen eine Rolle spielen, können aus der Multiprojekt-Sichtweise sowohl Kennzahlen der Einzelprojekte, als auch gebündelte Kennzahlen wichtig sein, wie etwa, um den Gesamtfortschritt eines Projektprogramms erfassen zu können. Dies soll nachfolgendes Beispiel veranschaulichen.

*Tabelle 3: Multiprojekt-Scorecard vom 01.03.2018 (Quelle: Eigene Darstellung in Anlehnung an Fiedler:2016, 60)*

| PERSPEKTIVE     | ZIEL                              | KENNZAHL   | ISTWERT                      |                             | ZIELWERT |          |
|-----------------|-----------------------------------|--|------------------------------|-----------------------------|----------|----------|
| <i>Zeit</i>     | Hohe Termintreue                  | Pünktl. erreichte Meilensteine / Alle Meilensteine | 80 %                         |                             | 100 %    |          |
|                 | Fortschritt aller Projekte        | Erreichte Meilensteine gesamt                      | 3/10                         |                             | 4/10     |          |
| <i>Finanzen</i> | Rentabilitäts-<br>erwartung       | Gewinn / Kapital<br>Projekt A, B                   | A<br>70%                     | B<br>10%                    | A<br>80% | B<br>5%  |
|                 | Wirtschaftlich-<br>keitserwartung | Ist- / Sollaufwand<br>Projekt A, B                 | 10.00<br>0 /<br>40.00<br>0 € | 5.00<br>0 /<br>20.0<br>00 € | <1       | <1       |
|                 | Budgetaus-<br>schöpfung           | Budgetausschöp-<br>fungsgrad in %<br>Projekt A, B  | A<br>20%                     | B<br>5%                     | A<br>15% | B<br>10% |
| <i>Kunde</i>    | Hohe Kundenzufrie-<br>denheit     | Anzahl<br>Eskalationsmeet-<br>ings Projekt A, B    | 1                            | 0                           | 0        | 0        |

<sup>61</sup> vgl. *Tabelle 3*, Finanzen

<sup>62</sup> vgl. *Tabelle 3*, Abhängigkeiten

<sup>63</sup> vgl. *Tabelle 3*, Kommunikation

|                       |   |                                      |  |                                   |
|-----------------------|---|--------------------------------------|--|-----------------------------------|
| <i>Kommunikation</i>  | Reibungslose Kommunikation zw. Projekten    | Anzahl gemeinsamer Meetings          | 3                                      | 3                                 |
|                       | Keine Konflikte zw. Projekten               | Anzahl Eskalationsmeetings           | 1                                      | 0                                 |
|                       | Gutes Wissensmanagement                     | Dokumentation (z. B. Best Practices) | 2 Dokumente                            | 4 Dokumente                       |
| <i>Abhängigkeiten</i> | Pünktliche Lieferung von Projektergebnissen | Zeitlich-inhaltliche Abhängigkeit    | A: 1/2 aller Ergebnisse an B geliefert | A: Alle Ergebnisse an B geliefert |

### 3.3.2 Vorteile Scorecard

1. *Vergleichbarkeit:* Durch die Scorecards der Einzelprojekte sind diese miteinander gut vergleichbar. Möglich wäre es zudem, wie *Tabelle 3* veranschaulicht, eine Scorecard auf Portfolio-Ebene zu entwerfen, welche zusätzliche Kennzahlen enthält, die etwa die Abhängigkeiten zwischen den einzelnen Projekten messen sollen.
2. *Fundament für Entscheidungen:* Auf Grundlage der Daten durch Einzelprojekt-Scorecards und Portfolio-Scorecard könnte die Entscheidungsfindung vereinfacht werden. Mögliche Fragen sind zum Beispiel: Ist das Portfolio in sich stimmig? Wäre es besser, *Projekt X* abubrechen und aus dem Portfolio zu nehmen? Hat *Projekt Z* einen neuen, wichtigen Nutzen aufgedeckt, den wir mit einem neuen Projekt-Programm realisieren könnten?
3. *Ursache-Wirkung:* Bei Konflikten mit nicht erreichten Zielwerten könnte eine Analyse der Scorecard unter Berücksichtigung der bekannten und im Vorfeld analysierten Ursache-Wirkungs-Beziehungen Zusammenhänge aufdecken. Die Konflikt-Analyse könnte dann etwa bei der Entscheidungs- und Maßnahmenfindung helfen, eine Korrektur von fehlerhaften Vorgängen vorzunehmen und letzten Endes die Zielwerte zu erreichen.
4. *Fortschrittsüberwachung:* Hinsichtlich der Fortschrittsüberwachung eines Projektbündels kann beispielsweise überprüft werden, ob der geplante Abschlusstermin eines Programms nach aktuellen Informationen der Scorecard erreicht werden kann. Auch Fragen

bezüglich der strategischen Ausrichtung eines Projektbündels könnten geklärt werden, etwa, ob das Portfolio hinreichend auf die Deckung strategischer Ziele ausgerichtet ist.

### 3.4 Vorgehensmodelle in der IT

*„Ein Vorgehensmodell beschreibt systematische, ingenieurmäßige und quantifizierbare Vorgehensweisen, um Aufgaben einer bestimmten Klasse wiederholbar zu lösen.“<sup>64</sup>*

Im Projektkontext bestimmt das Modell also die Art und Weise, wie ein Projekt umgesetzt werden soll. Dabei liefert es einen Organisationsrahmen für den gesamten Projektlebenszyklus und gibt Antworten auf die folgenden Fragen des Projekts:<sup>65</sup>

- Was?
- Wer?
- Womit?
- Wann?
- Wie?

Hierbei beschreibt das *Was?*, welche Projekt-Artefakte entstehen müssen (z. B. „Projektplan, Modelle, Code, Berichte, Anwenderdokumentation“)<sup>66</sup>, *Wer?* beschreibt Rollen und ihre jeweiligen Verantwortungen (z. B. „Projektleiter, Entwickler, Tester“)<sup>67</sup>, *Womit?* liefert die benötigten Werkzeuge und Methoden („(Quasi-)Standards, Werkzeuge, etc.“)<sup>68</sup>, *Wann?* das spezifische Ablaufmodell, nach welchem die einzelnen Projektphasen abzuwickeln sind und *Wie?* umfasst diejenigen Tätigkeiten, die erledigt werden müssen, um die Artefakte zu realisieren (z. B. „„Erstelle UML-Modell für Anforderungen.“, „Schreibe Unit Tests.“)<sup>69</sup>.

---

<sup>64</sup> siehe Broy; Kuhrmann:2013, 86.

<sup>65</sup> vgl. Broy; Kuhrmann:2013, 87.

<sup>66</sup> siehe Broy; Kuhrmann:2013, 87.

<sup>67</sup> siehe Broy; Kuhrmann:2013, 87.

<sup>68</sup> siehe Broy; Kuhrmann:2013, 87.

<sup>69</sup> siehe Broy; Kuhrmann:2013, 87.

Folglich könnten einige Ergebnisse aus der Anwendung der W-Fragen lauten:

- *Projektleiter* erstellt und pflegt Projektplan.
- *Entwickler A* schreibt Unit Tests.
- *Entwickler B* erstellt UML-Modell für Anforderungen.
- *Tester A* führt Unit Tests durch.
- *Tester B* führt Integrationstests durch.
- *Tester C* führt Systemtests durch.

Der Einsatz von Vorgehensmodellen im Allgemeinen liefert jedoch einige Vor- und Nachteile, die kritisch zu würdigen sind.

Positive Effekte für das Projektmanagement können durch die mögliche Standardisierung entstehen, wenn bestimmte Projekttypen mit denselben Vorgehensmodellen umgesetzt werden. Dadurch kann zudem eine gewisse Vergleichbarkeit der Einzelprojekte geschaffen werden, was die übergeordnete Steuerung von Projekten erleichtern und verbessern kann.

Auch die Vorgehensmodelle selbst können dadurch dauerhaft auf den spezifischen Unternehmenskontext angepasst werden und die Schulung von Mitarbeitern ermöglicht es, aus abgeschlossenen Projekten gesammeltes Wissen weiterzugeben, damit dies in künftigen Projektvorhaben eingesetzt werden kann. Um diese positiven Effekte, die durch die Nutzung von Vorgehensmodellen entstehen, zu erzielen, muss allerdings ein ausreichendes Verständnis über die Vorgehensmodelle vorherrschen, das von allen im Projektkontext agierenden Parteien geteilt wird.

Der vom Modell vorgegebene Rahmen für anfallende Projektaktivitäten kann also die Organisation eines Projekts erleichtern. Jedoch besteht permanent das Risiko, dass sich am Projekt beteiligte Parteien zu sehr am Vorgehensmodell orientieren und situationsbedingt notwendige Abweichungen von diesem nicht durchgeführt werden.

Daher lässt sich schlussfolgern, dass ein starres Festhalten an Vorgehensmodellen unvorteilhaft ist, weil jedes Projekt in einem anderen Kontext steht. Situationsspezifische Anpassungen sollten also generell möglich sein.

Das sogenannte Tailoring<sup>70</sup> erlaubt es, ein Vorgehensmodell an eine spezifische Projektsituation anzupassen.

In den nachfolgenden Unterkapiteln werden nun einige gängige Vorgehensmodelle vorgestellt, wobei zwischen klassischen Vorgehensmodellen und sogenannten agilen Ansätzen unterschieden wird. Dabei bauen die Vorgehensweisen auf unterschiedlichen Grundprinzipien auf, die sich sowohl in klassischen, als auch in agilen Ansätzen in unterschiedlicher Ausprägung wiederfinden lassen: lineare, iterative sowie inkrementelle Vorgehen.

#### 3.4.1 Klassische Vorgehensmodelle

Mit dem Anspruch, „*Software nach Ingenieur-Prinzipien zu entwickeln*“<sup>71</sup> und damit dem Entwicklungsprozess eine klare Struktur zu verleihen, entstand die naheliegende Idee, dass ein Projekt in aufeinanderfolgenden Phasen abgewickelt werden sollte. Dies stellt die Grundlage für viele klassische Vorgehensmodelle dar. Das Wasserfallmodell bildet mit seiner linearen Struktur die einfachste Variante, diese Phasen zu durchlaufen.

Auch das Spiralmodell setzt auf Basis dieser Idee auf.

Das V-Modell bzw. das V-Modell XT liefert ebenfalls ein relativ statisches Konstrukt, dessen Phasenablauf dadurch gekennzeichnet ist, dass zu jeder Planungs- und Umsetzungsphase eine Validierung und Verifikation in Form von Testfällen gehört,<sup>72</sup> woraus sich das typische V-Muster ableitet.

---

<sup>70</sup> vgl. Broy; Kuhrmann:2013, 88.

<sup>71</sup> siehe Höhn; Höppner:2008, 571.

<sup>72</sup> vgl. Lent:2013, 43.



### 3.4.1.1 Wasserfallmodell & Phasenmodell

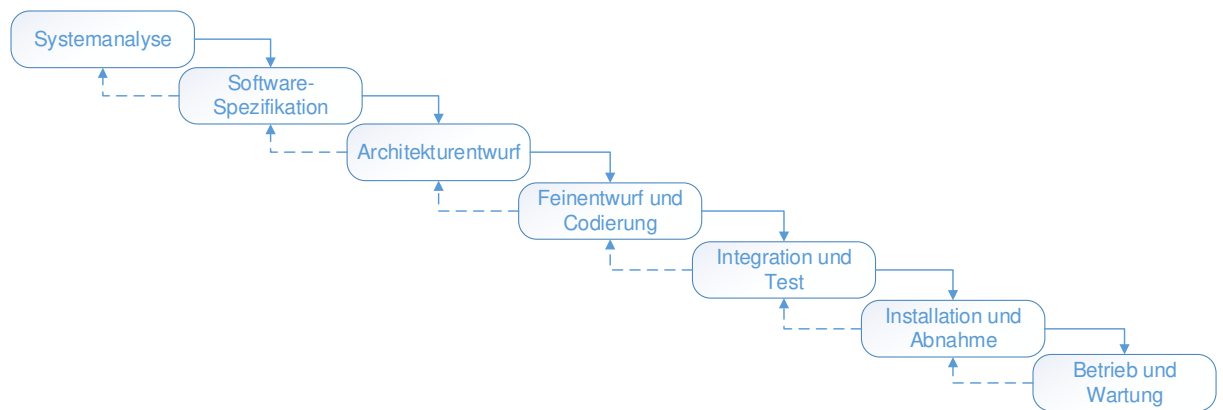


Abbildung 5: Wasserfallmodell (Quelle: Eigene Darstellung in Anlehnung an Rau:2016, 7).

Kerngedanke von linearen Vorgehensmodellen ist die Einteilung des Softwareentwicklungsprozesses in unterschiedliche Phasen, die aufeinanderfolgend abgearbeitet werden. Ein solches Vorgehen findet sich im sogenannten Wasserfallmodell, das ebenfalls auch als Phasenmodell bezeichnet wird, wieder<sup>73</sup>. Als die Softwareentwicklung in den 1960er Jahren noch in ihren Kinderschuhen steckte, wobei nach Rau eine „‘Code&Fix’-Mentalität“<sup>74</sup> vorherrschte, war es naheliegend, dem Prozess der Softwareentwicklung eine klare Struktur zu geben in Form von definierten Phasen, welche die notwendigen Tätigkeiten enthalten, die nacheinander abgearbeitet werden müssen, um ein funktionierendes Softwareprodukt zu entwickeln.<sup>75</sup>

Dieser Prozess weist eine lineare Struktur auf, wobei jedoch nach jeder abgeschlossenen Phase Rückkopplungen zur vorherigen möglich sind.

Der Wasserfall lässt sich zudem mit Meilensteinen ergänzen, welche die Übergänge zwischen den einzelnen Phasen markieren. Ist eine Phase des Prozesses beendet, wird mittels festgelegter Kriterien überprüft, ob die jeweilige Phase erfolgreich abgeschlossen wurde und zur nächsten Phase übergegangen werden kann.<sup>76</sup>

Mögliche Risiken ergeben sich aus der Linearität selbst: So werden Fehler, die zu Beginn des Prozesses entstehen, etwa in der Planung, oft sehr spät

<sup>73</sup> vgl. Abb. 5.

<sup>74</sup> siehe Rau:2016, 7.

<sup>75</sup> vgl. Broy; Kuhrmann:2013, 89.

<sup>76</sup> vgl. Rau:2016, 7.

entdeckt, weil die eigentliche Codierung erst viel später stattfindet und zudem erst danach getestet wird.<sup>77</sup>

Ein derartig lineares Vorgehen macht also nur Sinn, wenn bereits in der Planungsphase ein sehr genaues Abbild dessen konstruiert werden kann, was am Ende eines Projekts entstanden sein soll. Änderungen führen in diesem Modell zu hohem Aufwand, weshalb sie sinnvollerweise vermieden werden.

#### 3.4.1.2 Spiralmodell

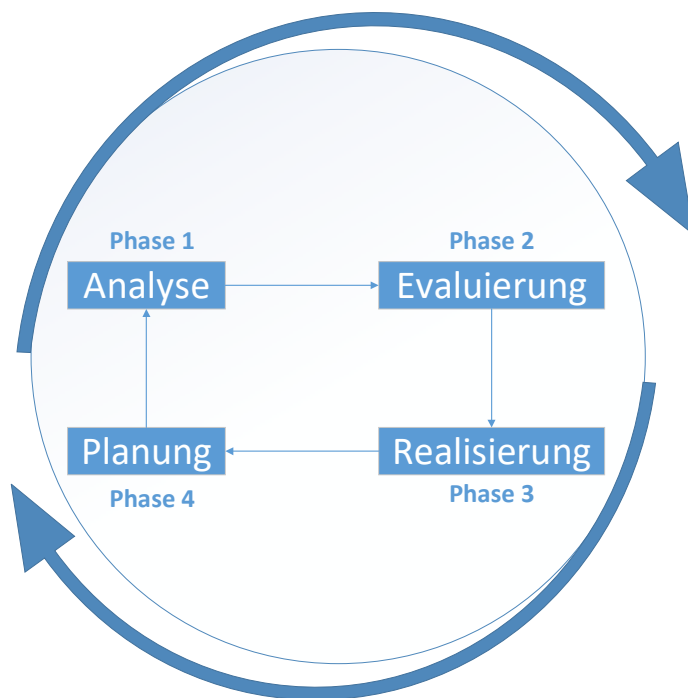


Abbildung 6: Spiralmodell (Quelle: Eigene Darstellung in Anlehnung an Broy; Kuhrmann:2013,91f).

Das Spiralmodell kann als Weiterentwicklung des klassischen Wasserfalls verstanden werden, wobei der Entwicklungsprozess in aufeinanderfolgenden Iterationen stattfindet. Jede Iteration durchläuft dieselben Phasen<sup>78</sup> mit dem Ziel, am Ende ein neues Inkrement, also Teilstück des Produkts, zu haben, welches neue Funktionalitäten der Software liefert.<sup>79</sup>

*Phase 1:* Zu Beginn einer neuen Iteration findet die Zielanalyse statt, in der die Ziele festgelegt werden, die in dieser Iteration erreicht werden sollen.

<sup>77</sup> vgl. Broy; Kuhrmann:2013, 91.

<sup>78</sup> vgl. Abb. 6.

<sup>79</sup> vgl. Hanser:2010, 5.

*Phase 2:* Im nächsten Schritt sollen mögliche Risiken gefunden und Alternativen evaluiert, sowie mögliche Prototypen geplant werden.

*Phase 3:* Die Prototypen werden in der Realisierungsphase umgesetzt und getestet.

*Phase 4:* Den letzten Schritt bildet die Planung des kommenden Zyklus.

Somit entsteht das Produkt Stück für Stück und der Prozess erlaubt es, dass Projektziele im fortschreitenden Projekt zumindest bis zu einem gewissen Grad änder- und erweiterbar bleiben. Damit fällt das Risiko von Fehlplanungen geringer aus als beim klassischen Wasserfallmodell. Auch die Nutzung von Prototypen wirkt diesem Risiko entgegen.

Diese zyklische Vorgehensweise des Spiralmodells erinnert an das altbewährte PDCA-Schema des Demingkreises mit den vier Schritten

*Plan – Do – Check – Act*, dessen Ursprung in der Qualitätssicherung anzusiedeln ist und der als Regelkreis im Lean Development im Kontinuierlichen Verbesserungsprozess (KVP) Anwendung findet.<sup>80</sup> Agile Modelle haben den Denkansatz der schrittweisen Entwicklung übernommen und noch weiter ausgeprägt, was in Abschnitt 3.4.2 näher beschrieben wird.

#### 3.4.1.3 V-Modell XT

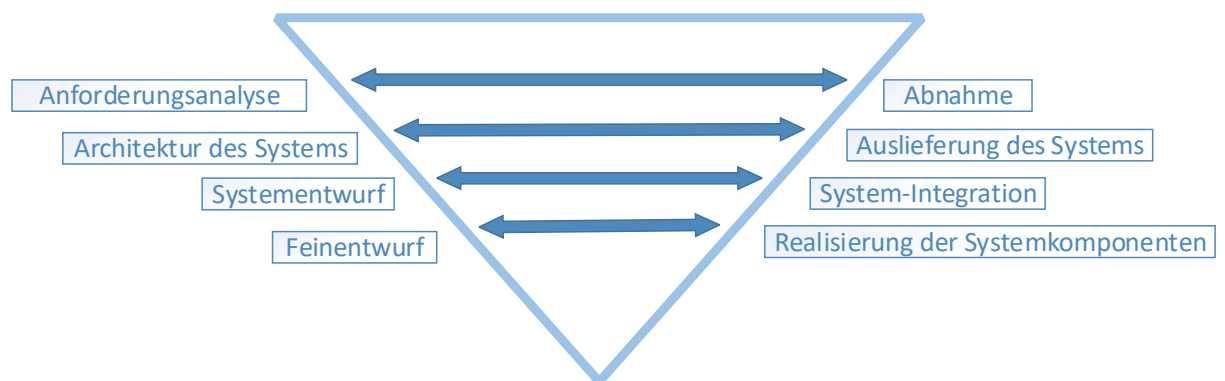


Abbildung 7: V-Modell (Quelle: Eigene Darstellung in Anlehnung an Höhn; Höppner:2008, 17).

Wie bereits angedeutet, durchläuft ein Projekt nach dem V-Modell ebenfalls eine Abfolge von Phasen. Dabei spielt das Konzept der *Verifikation und Validierung* eine zentrale Rolle: Während Tätigkeiten, die der Verifikation

<sup>80</sup> vgl. Dombrowski:2015, 34.

zugeordnet werden können, dafür sorgen, dass das zu erstellende Produkt spezifiziert wird, etwa zu Beginn durch die Anforderungsanalyse<sup>81</sup>, wird mit der Validierung sichergestellt, dass das Produkt die festgelegten Projektziele trifft. Die erfolgreiche Abnahme stellt in dem Fall den letzten Schritt der Validierung dar. Sprich: Die Verifikation, und damit alle Aufgaben auf der linken Seite des V, soll sicherstellen, dass ein funktionierendes Produkt entsteht, die Validierung mit den Aufgaben auf der rechten V-Seite hingegen soll gewährleisten, dass dieses Produkt jene Funktionen erfüllt, die benötigt werden.<sup>82</sup> Verifikation und Validierung werden also stufenweise durchgeführt, wobei jedem Verifikationsschritt ein Validierungsschritt zugeordnet ist (z. B. *Feinentwurf – Realisierung der Systemkomponenten*).

In seiner ersten Variante wurde das V-Modell bereits 1979 von B. Boehm veröffentlicht.<sup>83</sup> Seitdem wurde das Vorgehensmodell stetig weiterentwickelt und konnte 2005 durch das V-Modell XT (eXtreme Tailoring) ersetzt werden, welches seit 2017 in der Version 2.1 vorliegt. Besonders in Deutschland und Österreich findet das V-Modell XT häufig Anwendung und ist zudem seit 2005 für alle IT-Projekte des Bundes verbindlich,<sup>84</sup> wobei bereits die Vorgänger-Modelle seit den 1990er-Jahren als „*Standard für die Entwicklung von IT-Systemen der Bundesbehörden im zivilen und militärischen Bereich*“<sup>85</sup> genutzt wurden. Dies verdeutlicht, dass das V-Modell historisch von stark regulierten Umfeldern geprägt und somit dort besonders geeignet ist.<sup>86</sup>

Dennoch suggeriert die Erweiterung *eXtreme Tailoring*, dass das aktuelle Vorgehensmodell eine gewisse Anpassbarkeit aufweist, sodass es auch außerhalb kritischer Projektumfelder<sup>87</sup> Anwendung finden kann.

Dabei wird der Tailoring-Ansatz mittels Projekttypen, Vorgehensbausteinen

---

<sup>81</sup> vgl. *Abb. 7*.

<sup>82</sup> vgl. *Höhn; Höppner:2008*, 17.

<sup>83</sup> vgl. *Höhn; Höppner:2008*, 574.

<sup>84</sup> vgl. *Höhn; Höppner:2008*, Vorwort.

<sup>85</sup> siehe *Höhn; Höppner:2008*, 577.

<sup>86</sup> vgl. *Der Beauftragte der Bundesregierung für Informationstechnik:2018a*.

<sup>87</sup> Unter einem Projektumfeld versteht man den spezifischen Kontext, in welchem das Projekt umgesetzt wird (vgl. *InLoox GmbH:2018c*).

und Projektdurchführungsstrategien umgesetzt.<sup>88</sup> Unterstützt werden die drei folgenden Projekttypen:<sup>89</sup>

- Systementwicklungsprojekt eines Auftraggebers
- Systementwicklungsprojekt eines Auftragnehmers
- Einführung und Pflege eines organisationsspezifischen Vorgehensmodells

Unter einem Vorgehensbaustein werden im V-Modell XT die für die Lösung einer konkreten Aufgabe notwendigen Ergebnisse und Zwischenergebnisse, die als Produkte bezeichnet werden, sowie die zugehörigen Aktivitäten und Rollen verstanden. Ein Vorgehensbaustein klärt somit die Projektfragen *Was*, *Wer?* und *Wie*, auf die bereits im Abschnitt 3.4 eingegangen wurde.<sup>90</sup>

Das Konzept der Vorgehensbausteine erlaubt dem V-Modell XT eine gewisse Modularität, sodass dieses projektspezifisch mithilfe dieser Bausteine angepasst werden kann.

Projektführungsstrategien schließlich sollen Antwort auf die Projektfrage nach dem *Wann?* liefern, indem sie den zeitlichen und inhaltlichen Ablauf des Projektes festlegen.<sup>91</sup> Hier wird mit sogenannten Entscheidungspunkten gearbeitet, die als Meilensteine fungieren: Ein Projektabschnitt kann abgeschlossen werden, sobald der nachfolgende Entscheidungspunkt erreicht und der Erfolg des Projektabschnittes festgestellt werden konnte. Dabei ist für jeden Entscheidungspunkt festgelegt, welche Ergebnisse und Zwischenergebnisse, respektive Produkte, fertiggestellt, sowie welche Qualitätskriterien erfüllt sein müssen.<sup>92</sup>

Der Projekttyp gibt entsprechend vor, welche Vorgehensbausteine und welche Projektführungsstrategien zum Einsatz kommen können.<sup>93</sup> Zum Beispiel sind für den Projekttypen *Systementwicklungsprojekt eines Auftragnehmers* folgende Strategien vorgesehen:

---

<sup>88</sup> vgl. Höhn; Höppner:2008, 6.

<sup>89</sup> siehe Höhn; Höppner:2008, 5.

<sup>90</sup> vgl. Höhn; Höppner:2008, 6f.

<sup>91</sup> vgl. Höhn; Höppner:2008, 7.

<sup>92</sup> vgl. Höhn; Höppner:2008,, 7.

<sup>93</sup> vgl. Höhn; Höppner:2008, 8.

- „*Inkrementelle Projektdurchführungsstrategie (AN)*,“
- „*Komponentenbasierte Projektdurchführungsstrategie (AN)*,“
- „*Agile Projektdurchführungsstrategie (AN)*,“
- „*Wartung und Pflege (AN)*.“<sup>94</sup>

Hierbei zeigt sich, dass das V-Modell XT bereits agile Entwicklung im Rahmen von iterativem Vorgehen erlaubt, welches in den Ablauf des V-Modells für entsprechende Projekte integrierbar ist.<sup>95</sup> Auch agile Methoden wie Scrum lassen sich in die Zwischenräume der vorgegebenen Entscheidungspunkte integrieren.<sup>96</sup> Inwiefern eine Integration agiler Methoden Sinn macht, ist im Einzelfall zu entscheiden und hängt von der spezifischen Projektsituation ab, die nicht nur den Projekttypen umfasst, sondern viele weitere Faktoren, auf die an dieser Stelle nicht weiter eingegangen wird. Im Vergleich mit den in Kapitel 3.4.2 beschriebenen Methoden fällt jedoch auf, dass der Rahmen möglicher Agilität im V-Modell XT durch seine Vorgaben recht begrenzt ist.

Der sogenannte V-Modell XT-Kern soll eine einheitliche Qualitätsbasis des Managements für alle Projekte schaffen, die mit diesem Vorgehensmodell umgesetzt werden. Dabei umfasst der V-Modell XT-Kern die folgenden verpflichtenden Vorgehensbausteine:

- „*Projektmanagement*“
- „*Qualitätssicherung*“
- „*Konfigurationsmanagement*“
- „*Problem- und Änderungsmanagement*“<sup>97</sup>

Zusätzlich stellt das V-Modell XT für jeden Projekttyp optionale Vorgehensbausteine zur Verfügung.<sup>98</sup>

In einem stark regulierten Umfeld, in dem das korrekte Befolgen von Vorgaben erfolgsentscheidend für das Projekt ist, wie zum Beispiel im Kontext

---

<sup>94</sup> siehe Höhn; Höppner:2008, 183.

<sup>95</sup> vgl. *Der Beauftragte der Bundesregierung für Informationstechnik:2018b*, 390 Stichwort „Entwicklung, inkrementell“.

<sup>96</sup> vgl. *Kiesling:2011*, 57f.

<sup>97</sup> siehe Höhn; Höppner:2008, 11.

<sup>98</sup> vgl. Höhn; Höppner:2008, 15.

der Pharmaindustrie oder in Bundesbehörden, scheint das V-Modell XT ein geeignetes Vorgehensmodell zu sein, um einerseits Projekten einen standardisierten Aufbau und Ablauf zu verleihen, andererseits jedoch die Möglichkeit der Anpassung zu gewährleisten.

Jedoch macht bereits dieser oberflächliche Einblick in das V-Modell XT deutlich, dass ein hohes Maß an Wissen gefordert ist im Umgang mit diesem Vorgehensmodell, da dieses sehr detailliert und umfangreich dokumentiert ist.

### 3.4.2 Agile Ansätze

Wie der kurze Einblick in klassische Vorgehensmodelle, der beispielhaft am Wasserfall- bzw. Phasenmodell, dem Spiralmodell und dem V-Modell XT vollzogen wurde, verdeutlicht, sind die Vorgaben bezüglich des Ablaufs der Projektphasen und deren Struktur unterschiedlich ausgeprägt. Während das Spiralmodell und das V-Modell XT iterative und inkrementelle Vorgehensweisen erlauben, ist dies beim Wasserfallmodell kaum möglich, da die eigentliche Produktentwicklung erst gestartet wird, nachdem die Planung als abgeschlossen gilt.

Dennoch haben diese Vorgehensmodelle Gemeinsamkeiten, welche die Entwickler des Agilen Manifestes dazu veranlasst haben, dieses aufzusetzen: Eine flexible Dokumentation, etwa mittels Wikis oder Dokumenten-Management-Systemen<sup>99</sup>, die eine fortlaufende Aktualisierung ermöglicht und die kontinuierliche Entwicklung und Verfeinerung der Produktvision widerspiegelt, widerspricht dem Gedanken von einem klar strukturierten Phasenablauf. Jene klassischen Vorgehensmodelle werden von den Entwicklern des Agilen Manifestes als schwergewichtig<sup>100</sup> und daher als eher ungeeignet für typische Software-Entwicklungsprojekte empfunden, bei denen funktionierende Software, welche die Bedürfnisse der Kunden und Anwender erfüllen soll, im Zentrum steht.

Häufige Änderungen erfahren in stark reglementierten Umfeldern oftmals wenig Akzeptanz, sodass die in der Praxis gewählten Vorgehensweisen versuchen, diese weitestgehend zu vermeiden. Demgegenüber wird in den *Agilen Werten*

---

<sup>99</sup> vgl. *Hanser:2010*, 10.

<sup>100</sup> siehe *Hanser:2010*, 9.

die Relation festgehalten, dass „*Reagieren auf Veränderung mehr als das Befolgen eines Plans*“ zählt, wie bereits in Abschnitt 2.5 dargelegt wurde. Agile Ansätze sollen also eine gewisse Offenheit für Änderungen integrieren und sogar fördern, da deren Vertreter dies im Kontext der Softwareentwicklung als sinnvoll erachten.

Der zentrale Gedanke einer kontinuierlichen Verbesserung und Ausgestaltung spiegelt sich daher in den beiden Agilen Ansätzen wider, die nachfolgend vorgestellt werden: Scrum und Kanban. Darüber hinaus existieren zahlreiche weitere agile Methoden, z. B. Extreme Programming, sowie verschiedene Mischformen. In dieser Arbeit werden jedoch nur Scrum und Kanban betrachtet, da sie zum einen leicht verständlich sind und dabei typische agile Vorgehensweisen beinhalten, zum anderen gut für eine Gegenüberstellung geeignet sind. Insbesondere auch deshalb, weil Kanban ursprünglich nicht aus der agilen Projektwelt kommt, sondern dem Lean Development entstammt, während Scrum einer der klassischen großen Vertreter agiler Methoden ist.

#### 3.4.2.1 Scrum

Scrum bezeichnet eine agile Projektmanagementmethode<sup>101</sup>, wobei ein Scrum-Projekt iterativ und vor allem inkrementell umgesetzt wird. Zentral bei Scrum ist die Fokussierung auf wichtige Tätigkeiten und das Vermeiden von solchen, die keinen oder zu geringen Wert besitzen und daher aus Managementsicht Verschwendung darstellen. Hier findet sich eine Parallele zum Lean Product Development wieder.<sup>102</sup>

Stattdessen sollen, wie in allen agilen Methoden üblich, wertschöpfende Tätigkeiten im Mittelpunkt stehen, um daraus Nutzen für den Projektkunden zu erzeugen. Poppendieck spricht dabei von den 7 Prinzipien des Lean Thinking<sup>103</sup>, die ein Werkzeug im „Agile Customer’s Toolkit“ bilden:

- 1 „*Eliminate waste – only add value, not inventory.*“
- 2 „*Amplify learning – iterate.*“
- 3 „*Decide as late as possible – defer commitment.*“

---

<sup>101</sup> vgl. *Hanser:2010*, 61.

<sup>102</sup> vgl. *Hanser:2010*, 61.

<sup>103</sup> Das *Lean Thinking* hat seinen Ursprung im Lean Development und bildet die Basis für schlanke Produktion (vgl. *Dombrowski:2015*, 33). Diese Denkweise lässt sich jedoch in Einklang mit der agilen bringen, wie in den folgenden Abschnitten gezeigt wird.



- 4 „*Deliver as fast as possible – Pull value and eliminate delay.*“
- 5 „*Empower the team – train, trust, and lead.*“
- 6 „*Build integrity in – both customer perceived and conceptual.*“
- 7 „*See the whole – avoid sub-optimizing.*“<sup>104</sup>

Diese von Tom und Mary Poppendieck aufgestellten Prinzipien können teilweise bei Scrum wiedergefunden werden. Statt einer spezifikationsorientierten Vorgehensweise, bei der das zu erstellende Produkt bereits im Detail beschrieben wird, bevor die Umsetzung beginnt, gibt es in Scrum kurze Planungshorizonte, ähnlich wie im Spiralmodell, wobei das Scrum-Team sich nur darauf konzentriert, was für die aktuelle Iteration von Belang ist. Diese Iterationen werden in Scrum als *Sprint* bezeichnet und umfassen üblicherweise eine Zeitspanne zwischen einer und vier Wochen, wobei die Sprintlänge innerhalb eines Projektes immer gleich lang bleibt.

Hier wird das Prinzip der *Time Box* angewendet: Diese Time Box bildet ein vordefiniertes Zeitfenster, innerhalb dessen ein Stück funktionierende Software erstellt werden soll.<sup>105</sup> Dies ist notwendig, da der limitierende Faktor bei Scrum grundsätzlich die Zeit ist, was bedeutet, dass das Projektende bekannt ist, jedoch nicht das Projektergebnis. Ziel beim Timeboxing ist es, eine möglichst hohe Effizienz zu erlangen. Damit bildet die Zeit neben den Kosten eine feste Größe im magischen Dreieck. Die Leistung dagegen ist ein variabler Parameter.

Wichtig bei dieser Technik ist, dass bei Erreichen des Sprint-Endes die Iteration tatsächlich abgeschlossen wird, auch wenn nicht alle Sprintziele erreicht werden konnten. Gegebenenfalls müssen diese mit in den nächsten Sprint genommen und dann zuerst abgearbeitet werden.

Sämtliche Dokumentation soll in Scrum-Projekten so ausführlich wie nötig, jedoch so kurz wie möglich gehalten werden. Sie hat keinen statischen Charakter, sondern wird im Lauf der Projekte kontinuierlich überarbeitet und weiterentwickelt. Damit der Kunde seiner Mitverantwortung am Projekterfolg

---

<sup>104</sup> siehe *Poppendieck:2003*, 3f.

<sup>105</sup> vgl. *Schwaber; Sutherland:2016*, 8.

gerecht werden kann, sollte die Dokumentation für alle Projektbeteiligten transparent und leicht verständlich sein. Um in gemeinsamer Arbeit die richtige Produktvision finden und umsetzen zu können, wird dem Kunden eine aktive Rolle im Projektgeschehen zugedacht. Dies kann jedoch in der Praxis leicht zu Problemen führen, wenn die Kundenseite ihre Rolle missversteht und ihre Verpflichtungen (absichtlich oder unabsichtlich) nicht erfüllt. Dies verdeutlicht, dass der Erfolg agiler Methoden maßgeblich davon abhängt, dass ihre Anwender agiles Denken dem Agilen Manifest folgend pflegen.

Beispielhaft wird an dieser Stelle kurz die Technik des *User Story Mapping*<sup>106</sup> vorgestellt. Dies ist eine Visualisierungsmethode für die Darstellung der Systemstruktur eines zu entwickelnden Produkts. Bei Scrum werden die Systemanforderungen in kleine Teile heruntergebrochen. Diese finden sich als *Backlog Items* im *Product Backlog* wieder. Optisch abgebildet werden die Systemanforderungen jedoch in Form von *User Stories* auf *Story Cards*. Eine User Story stellt eine möglichst knapp und simpel formulierte Systemanforderung dar.<sup>107</sup> Durch das Herunterbrechen der Systemanforderungen wird es möglich, komplexe Probleme stückweise zu lösen. Beim Story Mapping werden die Story Cards so angeordnet, „dass diese Anordnung der Karten einen Überblick über die Funktionalitäten des zu realisierenden Systems gibt.“<sup>108</sup> Dabei werden zusammengehörige User Stories, die in ihrer Gesamtheit z. B. den Ablauf einer Nutzeraktivität beschreiben, gruppiert unter sogenannten *Epics*. Damit lassen sich etwa die einzelnen Schritte des Checkout-Prozesses in einem Online-Shop visualisieren, die der Nutzer durchlaufen muss, um die Bestellung abzuschicken, wobei der Checkout-Prozess das Epic und die einzelnen Schritte die User Stories darstellen würden.

Ziel des User Story Mapping ist es, ein gemeinsames Verständnis zwischen allen Projektbeteiligten sowie eine klare Übersicht über das Zielprodukt zu erzielen. Dies soll dafür sorgen, dass die in einen logischen Zusammenhang

---

<sup>106</sup> vgl. Abb. 8.

<sup>107</sup> vgl. Goll; Hommel:2015, 38.

<sup>108</sup> siehe Goll; Hommel:2015, 158.

gebrachten Systemanforderungen in sinnvoller Reihenfolge entwickelt werden, sodass das System am Ende nicht der eigentlichen Vision widerspricht.<sup>109</sup>

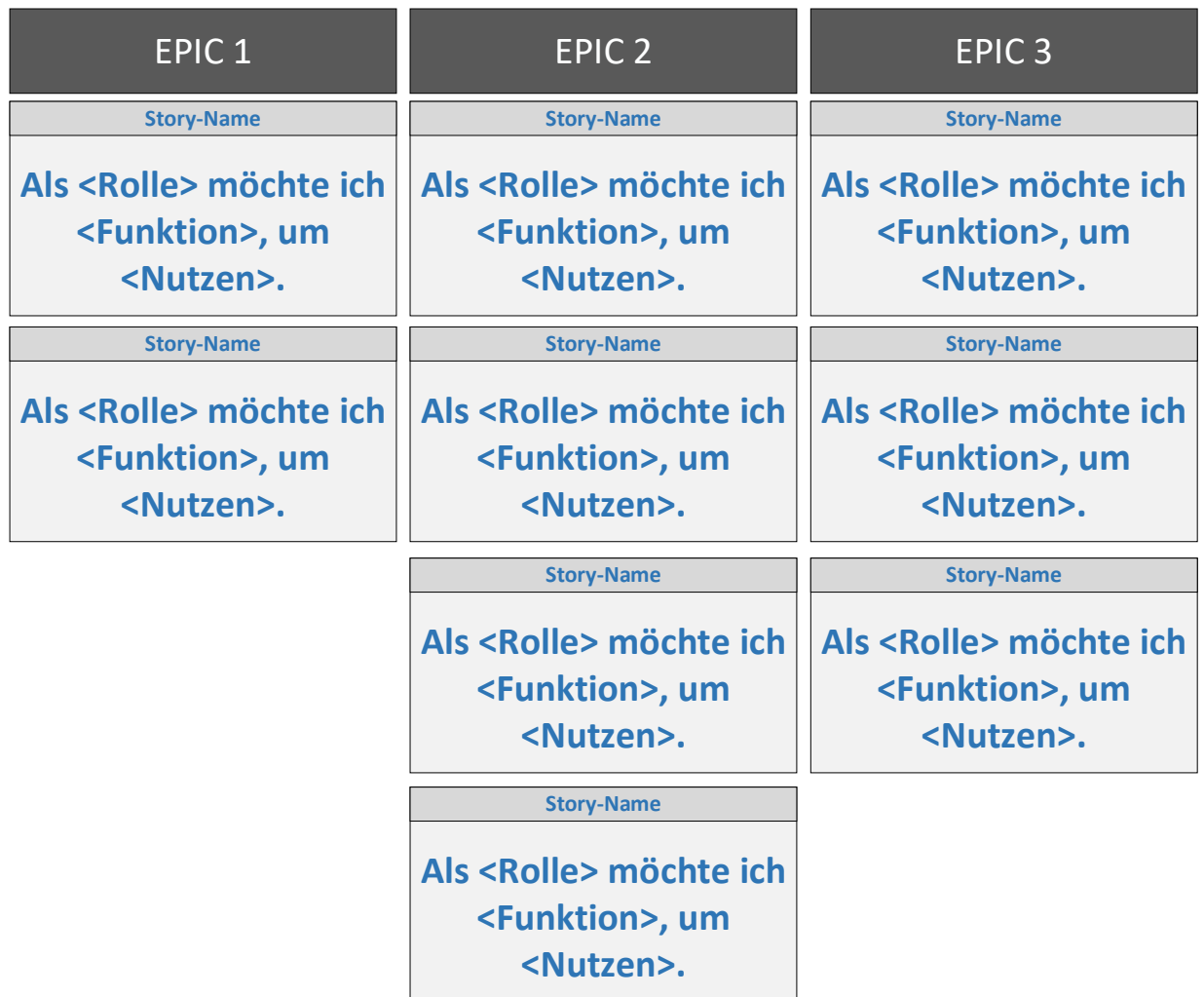


Abbildung 8: Beispiel Story Map (Quelle: Eigene Darstellung in Anlehnung an Goll; Hommel:2015, S. 39).

Das Rahmenwerk von Scrum, entwickelt von Ken Schwaber und Jeff Sutherland, umfasst *Rollen, Ereignisse, Artefakte* und einige Werte, Prinzipien und Regeln, welche das Funktionieren von Scrum sicherstellen sollen.<sup>110</sup>

*Transparenz, Überprüfung und Anpassung* gelten dem Scrum Guide zufolge als die drei Säulen, welche die Scrum-Theorie tragen: Diese Theorie beruft sich auf das Schöpfen von Wissen aus Erfahrung und der Entscheidungsfindung auf Grundlage von Bekanntem.<sup>111</sup> Letzteres spiegelt sich u.a. in den verkürzten

<sup>109</sup> vgl. Goll; Hommel:2015, 158.

<sup>110</sup> vgl. Schwaber; Sutherland:2016, 3.

<sup>111</sup> siehe Schwaber; Sutherland:2016, 3f.

Planungshorizonten wider, wobei Detailplanung nur für den aktuellen Sprint stattfindet.

*Transparenz, Überprüfung und Anpassung* können mithilfe der nachfolgend kurz erläuterten Techniken und Werkzeuge von Scrum „gebaut“ werden.

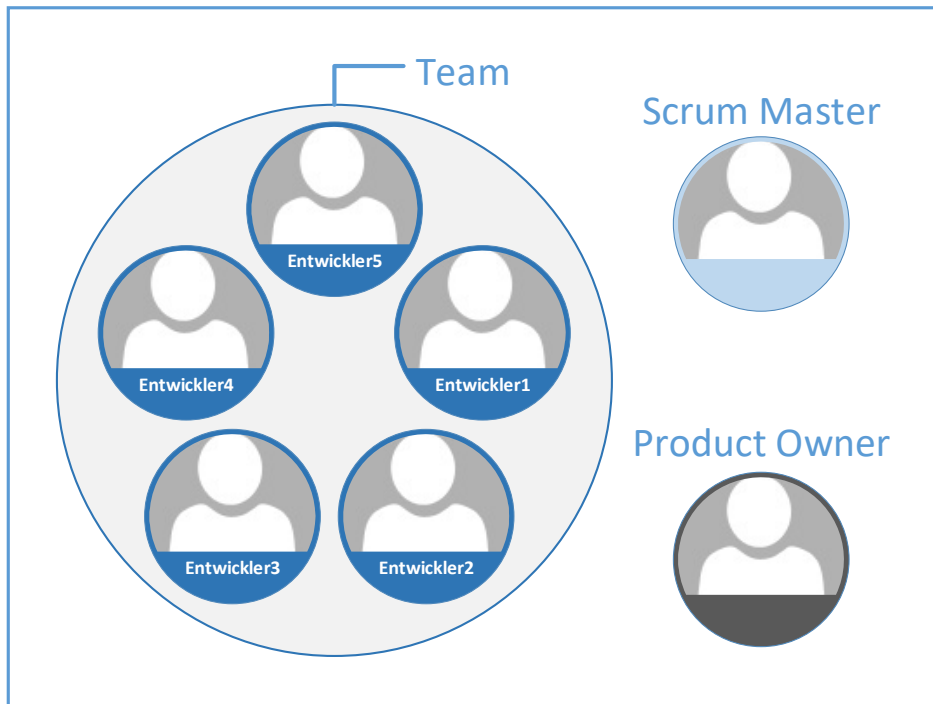


Abbildung 9: Rollen bei Scrum (Quelle: Eigene Darstellung).

Das Scrum Team setzt sich aus den folgenden Rollen zusammen: *Entwickler*, *Product Owner* und *Scrum Master*.<sup>112</sup> Ein Projektmanager ist nicht vorgesehen, stattdessen gilt das Prinzip der Selbstorganisation, was das Team leisten muss. Dies soll zu einer verbesserten Produktivität führen unter der Annahme, dass die Entwickler selbst ihre Arbeit am besten planen können.<sup>113</sup>

Der Product Owner trägt die Verantwortung für das im Projekt zu entwickelnde Produkt. Kernaufgabe des Product Owners ist die Pflege des *Product Backlogs*, welches eines der Artefakte in Scrum darstellt und was die Beschreibung aller infrage kommenden Produktfunktionen beinhaltet.<sup>114</sup> Das Product Backlog kann somit als eine, in eine bestimmte Rangfolge gebrachte, Auflistung von Funktionalitäten verstanden werden. Die Priorisierung der Backlog-Einträge

<sup>112</sup> vgl. Abb. 10.

<sup>113</sup> vgl. Schwaber; Sutherland:2016, 5.

<sup>114</sup> vgl. Schwaber; Sutherland:2016, 5.

obliegt dem Product Owner und gibt die Reihenfolge vor, nach welcher die Funktionalitäten entwickelt werden müssen.

Der Scrum Master nimmt im Idealfall die Rolle eines Koordinators ein, wobei seine primäre Funktion darin besteht dafür zu sorgen, dass alle Projektmitglieder Scrum verstehen und die Scrum-Spielregeln einhalten. Zudem sorgt er dafür, dass das Scrum-Team problemlos arbeiten kann. Hindernisse, die dem Team dabei im Weg stehen, sollen vom Scrum Master beseitigt werden. Kommunikation ist dabei das wichtigste Werkzeug des Scrum Masters, sodass es nicht verwunderlich ist, dass er die Rolle des Moderators übernimmt in den sogenannten Scrum-Meetings, die nachfolgend noch genauer erläutert werden.

Auch dem Product Owner steht der Scrum Master unterstützend zur Seite.<sup>115</sup>

Das Entwickler-Team ist für die Erstellung des Produkts verantwortlich. Die Verantwortung für die Ergebnisse trägt das Team gemeinsam.<sup>116</sup>

Sprint und die zugehörigen Scrum-Meetings werden als Ereignisse definiert. Dabei ist jedes dieser Ereignisse in eine bestimmte Time Box gebettet.<sup>117</sup> Dies soll dafür sorgen, dass z. B. Meetings nicht unnötig in die Länge gezogen werden und die zur Verfügung stehende Zeit produktiv genutzt wird.

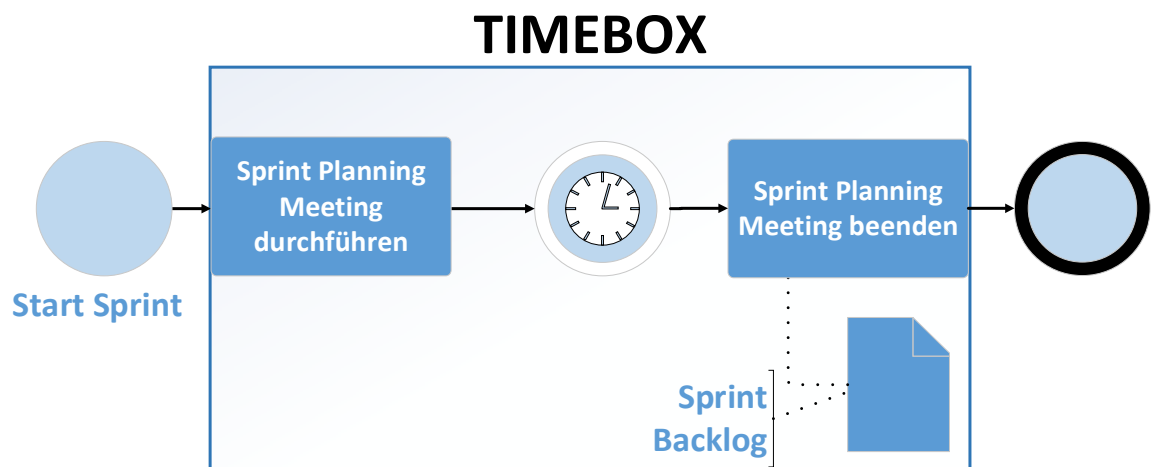


Abbildung 10: Timeboxing am Beispiel Sprint Planning Meeting (Quelle: Eigene Darstellung).

<sup>115</sup> vgl. Schwaber; Sutherland:2016, 5f.

<sup>116</sup> vgl. Schwaber; Sutherland:2016, 6.

<sup>117</sup> vgl. Abb. 5.

Zu Beginn eines neuen Sprints steht die Sprint Planung an, die im *Sprint Planning Meeting* durchgeführt wird. Im Zentrum der Sprint Planung steht die Erstellung des *Sprint Backlogs*, der ein weiteres definiertes Artefakt darstellt. Dafür müssen Entwickler-Team und Product Owner möglichst gemeinsam entscheiden, welche Einträge des Product Backlog in den Sprint Backlog übernommen werden sollen.<sup>118</sup>

Das Ende eines Sprints markiert das *Sprint Review*. Hier wird das im Sprint erstellte Inkrement, ein weiteres Artefakt, dem Product Owner und möglicherweise weiteren Stakeholdern vorgeführt. Der Product Owner muss letztlich entscheiden, ob er das Inkrement abnehmen und die im Sprint Backlog enthaltenen Einträge als „Done“ markiert werden können. Zudem soll eine Rekapitulation darüber stattfinden, welche technischen Probleme es ggf. gab und wie diese gelöst werden konnten. Außerdem wird das Product Backlog bei Bedarf aktualisiert, also erweitert und überarbeitet. Im Sinne des Projektcontrollings sollen zudem „*Zeitplan, Budget, die potentiellen Eigenschaften sowie die Markterwartungen für das nächste zu erwartende Produkt-Release überprüft*“ werden.<sup>119</sup>

Zuletzt wird in der *Sprint Retrospektive* gemeinsam von Scrum Master und Entwicklern analysiert, wie der abgeschlossene Sprint verlaufen ist. Dabei steht die Frage nach Verbesserungen in Bezug auf die Arbeitsweise des Teams im Vordergrund und wie diese Verbesserungen umgesetzt werden sollen.<sup>120</sup>

Der *Daily Scrum* ist ein täglich stattfindendes, 15-minütiges Meeting, das möglichst immer zur selben Zeit am selben Ort abgehalten wird. Dabei sind üblicherweise nur das Entwickler-Team und der Scrum Master anwesend, es sei denn, weitere Personen werden explizit eingeladen. Hierbei werden von den einzelnen Teammitgliedern die folgenden drei Fragen beantwortet:

- 1 Was habe ich seit dem letzten Daily Scrum getan?
- 2 Was möchte ich bis zum nächsten Daily Scrum erreicht haben?

---

<sup>118</sup> vgl. Goll; Hommel:2015, 94.

<sup>119</sup> siehe Schwaber; Sutherland:2016, 13.

<sup>120</sup> vgl. Schwaber; Sutherland:2016, 13.

### 3 Welche Hindernisse halten mich davon ab, meine Aufgaben zu erfüllen?<sup>121</sup>

Scrum ist also in der Theorie schnell verstanden, denn der offizielle Scrum Guide umfasst nur 19 DIN A4 Seiten. Das heißt, dass Scrum relativ einfach gelernt werden kann, jedoch wäre es wohl vermessen zu glauben, dass der Erfolg von Scrum nur davon abhängt, die Dokumentation zu kennen. Gerade dann, wenn Scrum in einem Umfeld eingeführt werden soll, das bisher auf klassische Vorgehensmodelle spezialisiert war, können schnell Probleme auftreten. Erfolgsentscheidend sind u.a. die folgenden Faktoren:

- *Dynamik*: Dokumente, wie z. B. Backlogs, sind nicht statisch und Änderungen gehören zum Prozess dazu
- *Informationsfluss*: Alle kennen den aktuellen Projektstatus
- *Verständnis*: Die Regeln werden beachtet und die Zuständigkeiten jeder Rolle eingehalten
- *Selbstorganisation*: Diese erfordert entsprechendes Know-How aller Entwickler und die nötige Disziplin. Ebenso müssen sie die Bereitschaft zur Reflexion eigener Verhaltens- und Arbeitsweisen aufbringen.

Der Erfolg hängt demzufolge stark von den Individuen ab, die das Projekt abwickeln. Darüber hinaus müssen die Strukturen des jeweiligen Unternehmens eine offene Kommunikations- und Fehlerkultur erlauben.

#### 3.4.2.2 Kanban & Lean Development

Kanban ist eine Methode aus der Produktion, die ursprünglich von Taiichi Ohno entwickelt wurde, um Produktionsprozesse langfristig zu verbessern und zu verschlanken.<sup>122</sup> Die Produktionsprozesse sind hierbei nicht statisch, sondern folgen einem evolutionären Verbesserungsschema, in welchem kleinste Veränderungen zu einer kontinuierlichen Verbesserung führen sollen. Demzufolge werden Prozesse im Kanban nicht in kurzer Zeit umfassenden Veränderungsmaßnahmen unterzogen. Stattdessen gilt es, den Ist-Zustand

---

<sup>121</sup> vgl. Goll; Hommel:2015, 98.

<sup>122</sup> vgl. Dombrowski:2015, 84.

eines Prozesses zu erfassen und davon ausgehend Verbesserungen einzuführen, ohne den ursprünglichen Prozess zu verwerfen.<sup>123</sup>

Damit ist Kanban strenggenommen kein agiles Vorgehensmodell, findet jedoch auch außerhalb der Produktion im Change Management<sup>124</sup> Anwendung.

David J. Anderson entwickelte zudem ein Kanban-System für die Softwareentwicklung, welches ebenfalls in Projektsituationen zum Einsatz kommen kann.<sup>125</sup> Da es hierbei viele Überschneidungen mit den in der agilen Projektwelt vorherrschenden Denkweisen gibt, lässt sich dieses Kanban-System gut darin einbetten. Dabei stehen die nachfolgend aufgelisteten und kurz erläuterten sechs Prinzipien im Zentrum:<sup>126</sup>

1. *Arbeit sichtbar machen*: Mithilfe eines Kanban-Boards können sowohl die einzelnen Aufgaben als auch der Arbeitsfluss eines Prozesses visualisiert werden. Die Erstellung des Kanban-Boards obliegt dem Kanban-Team, das aus denjenigen Personen besteht, die letzten Endes die Arbeiten im Prozess oder Projekt verrichten. Eine mögliche und simple Darstellung ist die Abbildung der einzelnen Arbeitstypen und die Zuordnung ihrer zugehörigen Tätigkeiten in Tabellenform.<sup>127</sup> Dabei stellt die Tabelle den Prozessfortschritt der einzelnen Aufgaben dar. In der *Input Queue* befinden sich Aufgaben, die weiter bearbeitet werden sollen, sobald die notwendigen Kapazitäten dazu freigeworden sind. Analog dazu werden in der Spalte *Bereit für Release* diejenigen Aufgaben gesammelt, die fertig sind und auf einen Release warten.<sup>128</sup>

Tabelle 4: Beispiel eines Kanban-Boards in Tabellenform (Quelle: Eigene Darstellung in Anlehnung an Leopold; Kaltenecker:2013, 27).

| <b>Input Queue</b> | <b>Analyse</b> | <b>Entwicklung</b> | <b>Test</b> | <b>Bereit für Release</b> |
|--------------------|----------------|--------------------|-------------|---------------------------|
|                    |                |                    |             |                           |
|                    |                |                    |             |                           |

<sup>123</sup> vgl. Leopold; Kaltenecker:2013, 15.

<sup>124</sup> vgl. Gabler Wirtschaftslexikon:2018k.

<sup>125</sup> vgl. Leopold; Kaltenecker:2013, 12.

<sup>126</sup> vgl. Leopold; Kaltenecker:2013, 17ff.

<sup>127</sup> vgl. Tabelle 4.

<sup>128</sup> vgl. Leopold; Kaltenecker:2013, 27.



Die Visualisierung mithilfe eines Kanban-Boards ermöglicht eine schnelle Erfassung des aktuellen Projektstatus sowie das Erkennen von Zusammenhängen, sodass etwa auf einen Blick Engpässe deutlich werden. Das setzt jedoch voraus, dass die gewählte Visualisierungsform am Kanban-Board leicht verständlich ist und ein gemeinsames Verständnis darüber vorherrscht.

2. *Work in Progress (WiP) wird limitiert*: Um das Kanban-System nicht zu überlasten, sollen die Aufgaben, die gleichzeitig abgearbeitet werden, limitiert werden. Das führt dazu, dass Projektmitglieder z. B. höchstens zwei Arbeitspakete parallel bearbeiten können. Das soll zu kürzeren Durchlaufzeiten führen, weil produktiver und störungsfreier an wenigen Aufgaben gearbeitet und diese somit zügiger abgeschlossen werden können.<sup>129</sup> Limitiert werden die Aufgaben z. B. durch eine bestimmte Anzahl Karten, wobei auf jeder Karte eine Aufgabe kurz beschrieben wird – „für Messungen wichtige Informationen wie zum Beispiel Start- und Fertigstellungszeitpunkte, Bearbeitungsdauer in den einzelnen Prozessschritten oder Ursachen für Blockaden“<sup>130</sup> können darauf ebenfalls vermerkt werden. Die Karten werden am Kanban-Board angebracht und durchlaufen dieses gemäß ihres Bearbeitungsstatus.<sup>131</sup>
3. *Flow Management*: Die Steuerung des Arbeitsflusses zielt darauf ab, Engpässe und Blockaden zu verhindern bzw. diese ggf. aufzudecken und zu beseitigen. Dabei werden die vom Kanban-Team identifizierten Arbeitstypen (in einem Softwareentwicklungsprozess z. B. *Analyse*, *Entwicklung* und *Test*) und Serviceklassen dazu verwendet, den anfallenden Aufgaben Service Level Agreements zuzuordnen, die festlegen, in welchem Zeitraum und nach welchen Qualitätskriterien diese Arbeiten verrichtet werden müssen. Die SLAs geben also Aufschluss über die Priorität einer Arbeit.<sup>132</sup> Anders als etwa im Wasserfallmodell wird der Arbeitsfluss nicht nach dem Push-, sondern dem Pull-Prinzip verrichtet: Fertige Aufgaben werden nicht einfach in den nächsten Prozessschritt geschoben, sondern werden vom

---

<sup>129</sup> vgl. Leopold; Kaltenecker:2013, 18f.

<sup>130</sup> siehe Leopold; Kaltenecker:2013, 36.

<sup>131</sup> vgl. Leopold; Kaltenecker:2013, 36.

<sup>132</sup> vgl. Leopold; Kaltenecker:2013, 19f.

nachgelagerten Prozessschritt dann gezogen, wenn er die Kapazitäten dazu hat.<sup>133</sup> So warten zum Beispiel dringende Aufgaben in der Input Queue darauf, im folgenden Prozessabschnitt bearbeitet zu werden.

4. *Explizite Prozessregeln schaffen*: Prozessregeln sollen die Arbeitsweise des Kanban-Teams und ggf. zugehöriger Stakeholder festhalten. Es gilt allerdings, dass Regeln nicht statisch sind, sondern angepasst werden müssen, sobald festgestellt wird, dass sie nicht mehr zielführend sind.<sup>134</sup> Damit diese Regeln Anwendung finden können, müssen sie jedoch in allgemein verständlicher Form kommuniziert und bestenfalls auch visualisiert werden.
5. *Feedback-Mechanismen*: Feedback-Mechanismen sind notwendig, um regelmäßig Zustände und Prozesse reflektieren zu können und dementsprechende Verbesserungen abzuleiten. Dies wird u.a. durch Daily Standup Meetings, also täglichen kurzen Zusammentreffen zum Austausch des aktuellen (Projekt-)Status, ähnlich dem Daily Scrum, sowie durch regelmäßige Retrospektiven und Reviews umgesetzt.<sup>135</sup>
6. *Gemeinschaftliche Verbesserungen*: Die Verbesserung von Arbeitsabläufen von denjenigen Menschen vornehmen zu lassen, welche die jeweiligen Tätigkeiten verrichten und dementsprechend wissen, welche Schwachstellen in den Prozessen vorliegen, hat den Vorteil, dass die damit einhergehenden Veränderungen besser akzeptiert werden. Dabei schreibt Kanban jedoch nicht vor, wie diese Verbesserungen zustande kommen sollen.<sup>136</sup>

Da die Kanban-Methode nicht explizit vorgibt, wie diese sechs Prinzipien konkret umzusetzen sind, ist sie dazu geeignet, sie auf verschiedenartige Projektsituationen anzupassen.

Wie auch bei Scrum ist die Kommunikation eines der Schlüsselwerkzeuge, mit dem der Erfolg dieses Ansatzes steht und fällt. Der Projektstatus muss konsequent transparent gehalten werden, sodass das Kanban-System nicht über- oder gar unterlastet wird und kontinuierlicher Fortschritt erreicht werden

---

<sup>133</sup> vgl. Leopold; Kaltenecker:2013, 29f.

<sup>134</sup> vgl. Leopold; Kaltenecker:2013, 20f.

<sup>135</sup> vgl. Leopold; Kaltenecker:2013, 21.

<sup>136</sup> vgl. Leopold; Kaltenecker:2013, 21f.

kann. Eine offene Fehlerkultur ist hierbei notwendig, um die Arbeitsmethoden immer wieder aufs Neue zu reflektieren und ggf. anzupassen. Auch Selbstorganisation ist eine wichtige Fähigkeit, mit welcher das Kanban-Team ausgestattet sein muss. Nur so ist das Team in der Lage, eigene Arbeitsweisen auf Schwachstellen hin zu analysieren, Verbesserungen auszuarbeiten und den Arbeitsfluss aufrecht zu erhalten.

#### 3.4.2.3 Vergleich von Scrum & Kanban

Beide Ansätze, sowohl Scrum als auch Kanban, können den Agilen Methoden zugeordnet werden, auch wenn dies in Bezug auf Kanban umstritten ist.<sup>137</sup>

Beide Methoden setzen auf das Einhalten einfacher Prinzipien, wie etwa das Transparentmachen von Work Flow und (Projekt-)Fortschritt. Viele Techniken beider Methoden spiegeln die Kerngedanken des Agilen Manifestes wieder, so etwa die ausgeprägte Meetingkultur, die in beiden Ansätzen möglichst offen und zielführend ausgestaltet sein sollte. Aufgrund der in dieser Arbeit beschriebenen Eigenschaften des Kanban-Systems ist es dazu geeignet, in der agilen Projektwelt Anwendung zu finden.

In einigen Punkten ist die Verwandtschaft beider Methoden zum Lean Development offensichtlich: Im Zentrum dabei steht der Antrieb, (Projekt-)Arbeit effizient zu gestalten, indem das Endziel in kleine Teile zerlegt wird. Diese Vorgehensweise soll die Lösung komplexer Aufgaben ermöglichen, wobei dabei so wenig wie möglich Verschwendung betrieben wird.

Beide Methoden sind besonders dann geeignet, wenn die benötigten Funktionalitäten des Endprodukts zu Projektbeginn nicht vollständig bekannt sind. Während Scrum die schrittweise Lösungsfindung durch zeitlich beschränkte Iterationen nach dem Timeboxing-Prinzip umsetzt, ist bei Kanban der limitierende Faktor stets die Arbeit (WiP), die über einen evolutionären Ansatz kontinuierlich erledigt wird.

Kontinuierliche Verbesserung wollen Vertreter beider Methoden erreichen, sodass die Fähigkeit zur Reflexion sowie Kritik- und Fehlertoleranz kritische Eigenschaften sind, welche die Teammitglieder mitbringen und/oder erlernen müssen. Auch Transparenz des Arbeitsprozesses ist notwendig, weshalb die

---

<sup>137</sup> vgl. *Leopold; Kaltenecker:2013*, 14.

Kommunikation einen weiteren kritischen Erfolgsfaktor beider Verfahren darstellt. Daher ist es nicht verwunderlich, dass die Visualisierung des Arbeitsflusses sowohl bei Kanban durch das Kanban-Board, als auch bei Scrum durch das User Story Mapping für den Erfolg beider Methoden herausragend ist.<sup>138</sup>

In der Praxis werden Teile von Kanban und Scrum auch kombiniert und auf individuelle Projektsituationen angepasst.<sup>139</sup>

---

<sup>138</sup> vgl. *Goll; Hommel:2015*, 130.

<sup>139</sup> vgl. *cPrime:2018*.

## 4 Controlling in den unterschiedlichen Projektwelten

---

Nachdem in Abschnitt 3 die Grundlage dafür geschaffen wurde, Anforderungen an das Controlling in den beiden unterschiedlichen Projektwelten – der „klassischen“ spezifikationsorientierten und der agilen – zu stellen, können diese nun nachfolgend definiert und im Anschluss daran analysiert und verglichen werden. Dabei wird der Begriff der *spezifikationsorientierten Projektwelt* in dieser Arbeit dafür verwendet, eine (wertfreie) Abgrenzung zu schaffen vom *agilen Projektumfeld*, indem Projekte, die nicht agil sind, als spezifikationsorientiert definiert werden. Da Spezifikationen auch in agilen Projekten kein Fremdwort sind, eine starke Fokussierung im Projektgeschehen darauf gemäß des Agilen Manifestes jedoch unerwünscht ist<sup>140</sup>, bietet sich die Spezifikationsorientierung als Abgrenzungsmerkmal in dieser Arbeit an.

Das Controlling wird in unterschiedlichen Projektsituationen entsprechend den spezifischen Umständen und Anforderungen gehandhabt. D.h. je nach Art und Umfang eines Projekts und seiner Bedeutung für das Unternehmen, der Unternehmensphilosophie, Personalverfügbarkeit und weiterer Faktoren fällt das Controlling mehr oder weniger umfangreich aus. So ist es beispielsweise in bestimmten, weltweit agierenden Konzernen wie etwa der IBM, nicht unüblich, Controlling-Funktionen in kleineren Projekten dem Projektmanager zu übertragen statt einen hausinternen Controller einzusetzen.<sup>141</sup> Fließt das Projektcontrolling in den Aufgabenbereich des Projektmanagers mit ein, dann fallen die Projektmanagement- und die Controllingperspektive zusammen, während ein eigenständiger Bereich für Projektcontrolling den Schwerpunkt z. B. auf die Unterstützung des Projektmanagements (etwa bei der Ziel- und Entscheidungsfindung oder bei der Projektplanung<sup>142</sup>) setzen kann. Natürlich können auch wichtige Stakeholder, z. B. der Kunde des Projekts, Einfluss auf das Design des Controllings und möglicher Methoden haben. Demnach erscheint es wenig hilfreich, ein pauschales Projektcontrolling beschreiben zu wollen, weder für Projekte in spezifikationsorientiertem, noch in agilem Kontext.

---

<sup>140</sup> vgl. 2.5.1 *Agile Softwareentwicklung*, 13f.

<sup>141</sup> Das gewählte Beispiel entstammt persönlichen Erfahrungen des Autors, die im Projektkontext im Rahmen eines Studien-Pflichtpraktikums im Jahr 2013 gesammelt wurden.

<sup>142</sup> vgl. Kargl; Kütz:2007, 1.

Dennoch lassen sich Vorgehensweisen skizzieren, die typisch sind für die jeweilige Projektwelt und auf ihre spezifischen Anforderungen zurückzuführen sind.

Davon ausgehend, dass die Vorgehensmodelle beider Projektwelten ihre Daseinsberechtigung haben, können typische Probleme, die zum Scheitern eines Projektes führen, ebenso in beiden Welten auftreten. Diese im Vorfeld zu bedenken kann dabei helfen, den für eine spezifische Projektsituation am besten geeigneten Ansatz zu finden. Die nachfolgende Tabelle soll dabei beispielhaft anhand der Perspektiven *Kommunikation*, *Dokumentation* und *Steuerung* aufzeigen, welche kritischen Faktoren dort zu einem gescheiterten Projekt führen können, wobei die skizzierten Problembeschreibungen und mögliche, daraus resultierende Folgen lediglich eine Auswahl zur Veranschaulichung liefern sollen.

Tabelle 5: Typische Gründe für Projektscheitern (Quelle: Eigene Darstellung).

| PERSPEKTIVE   | PROBLEM   | MÖGLICHE FOLGEN  |
|---------------|---|--|
| Kommunikation | Träge Kommunikationsweise   | <ul style="list-style-type: none"> <li>• Probleme/Hindernisse werden zu spät erkannt</li> <li>• Vertrauensverlust zw. Projektbeteiligten</li> </ul>  |
|               | Kommunikation findet nicht oder unzureichend statt (z. B. teamintern) | <ul style="list-style-type: none"> <li>• Kundenwünsche werden nicht erkannt</li> <li>• Probleme/Hindernisse werden nicht oder zu spät erkannt</li> <li>• Vertrauensverlust zw. Projektbeteiligten</li> </ul>                                   |
|               | Keine Offenheit/Fehlertoleranz  | <ul style="list-style-type: none"> <li>• Vertrauensverlust zw. Projektbeteiligten</li> <li>• Probleme/Hindernisse werden nicht oder zu spät erkannt</li> <li>• Motivation der Projektbeteiligten sinkt</li> <li>• Schuldzuweisungen</li> </ul> |

|               |   |  |
|---------------|---|--|
| Dokumentation | Keine saubere Dokumentation, z. B. unvollständig, widersprüchlich, nicht aktuell. | <ul style="list-style-type: none"> <li>• Unterschiedliche Erwartungshaltungen zw. Projektbeteiligten</li> <li>• Kundenwünsche werden nicht umgesetzt</li> <li>• Kommunikationsprobleme</li> <li>• Probleme/Hindernisse werden zu spät erkannt</li> </ul> |
| Steuerung     | Unbrauchbare Aufwandsschätzungen  | <ul style="list-style-type: none"> <li>• Termine können nicht fristgerecht eingehalten werden</li> <li>• Kosten übersteigen Nutzen</li> <li>• Personalmängel durch falsche Ressourcenplanung</li> </ul>  |
|               | Keine angemessene Fortschrittskontrolle   | <ul style="list-style-type: none"> <li>• Probleme/Hindernisse werden zu spät erkannt</li> <li>• Planung kann nicht eingehalten werden</li> </ul>   |

## 4.1 Controlling in der spezifikationsorientierten Projektwelt

### 4.1.1 Kommunikation & Dokumentation

Die Grundlage für die verbindliche Zielvereinbarung zwischen Auftraggeber und -nehmer wird in der Regel gebildet durch die Einheit aus Lasten- und Pflichtenheft.<sup>143</sup> Während das Lastenheft den Projektauftrag aus Kundensicht in einer Anforderungsbeschreibung abbilden soll, bildet das Pflichtenheft die Umsetzungsvorstellungen des Auftragnehmers ab in Form eines Grobkonzeptes.

Das Lastenheft beschreibt damit das *WAS* (=Ziel) des Projektes, das Pflichtenheft komplementär das *WIE* (=Weg zum Ziel) und damit diejenigen Leistungen, welche im Rahmen des Projekts vom Auftragnehmer zu erbringen sind.<sup>144,145</sup>

<sup>143</sup> vgl. *Abb. 11*.

<sup>144</sup> vgl. *Montag:2012a*.

<sup>145</sup> vgl. *Montag:2012b*.



Abbildung 11: Lasten- und Pflichtenheft als Grundlage für Zielvereinbarung (Quelle: Eigene Darstellung).

Ist ein spezifikationsorientierter Ansatz gewählt, dann hat der Auftraggeber entweder bereits zu Beginn eine präzise Vorstellung davon, was der Projektoutput leisten muss und ist dazu in der Lage, diese Vorstellung hinreichend mithilfe des Lastenhefts zu beschreiben<sup>146</sup>, oder das trifft nur bedingt zu und die Anforderungen an das Produkt müssen in mehreren Iterationen erarbeitet und umgesetzt werden.<sup>147</sup> Spätestens am Ende jedoch liegen eine detaillierte Beschreibung aller Komponenten sowie die Liste der zu erfüllenden Abnahmekriterien vor in Form einer Spezifikation.<sup>148</sup> In agilen Kontexten darf die Dokumentation bzw. Spezifikation ebenso wenig fehlen, jedoch wird eine ausufernde Dokumentation, dem Agilen Manifest folgend, zu Gunsten funktionierender Software nicht angestrebt. Da im Prinzip in jeder Iteration mit einer Änderung der Systemanforderungen gerechnet werden muss, würde das sonst zu einem hohen Aufwand bei der Aktualisierung des Anforderungskatalogs im Pflichtenheft oder der Spezifikation führen.

Andererseits ist bei dem in der spezifikationsorientierten Projektwelt vorherrschenden dokumentenlastigen<sup>149</sup> Planungsstil eine gewisse Anpassbarkeit vonnöten. Keine Projektplanung ist exakt und jedes Projekt kann durch Unvorhersehbarkeiten notwendige Änderungen erfahren.<sup>150</sup> Damit können

<sup>146</sup> vgl. 3.4.1.1. Wasserfallmodell & Phasenmodell.

<sup>147</sup> vgl. 3.4.1.2. Spiralmodell.

<sup>148</sup> vgl. *Der Beauftragte der Bundesregierung für Informationstechnik:2018c*.

<sup>149</sup> vgl. *Hanser:2010*, 9.

<sup>150</sup> vgl. *Fiedler:2016*, 66.



auch Dokumente, die im Projekt entstehen, während seiner Laufzeit nicht unveränderbar sein.<sup>151</sup> Ein Nachteil ergibt sich daraus insbesondere dann, wenn bereits zu Beginn eine detaillierte Beschreibung des Produkts angefertigt wird: Die fortlaufende Pflege der Dokumente kann demzufolge erheblichen Aufwand erzeugen durch häufige und umfassende Änderungen.

#### 4.1.2 Steuerung

Die Planung durch das Projektmanagement erfolgt bereits zu Beginn sehr detailliert, wobei das gewählte Vorgehensmodell den konkreten Planungshorizont vorgibt. So wird z. B. bei iterativen Vorgehen, ähnlich wie im agilen Kontext, eine Detailplanung zu Beginn eines neuen Zyklus benötigt, sodass spätere Iterationen noch nicht detailliert geplant werden müssen. Ziel der Projektplanung ist es dabei, möglichst genaue Aussagen treffen zu können hinsichtlich der zu erwartenden Kosten sowie die Ressourcenaufwände abzuschätzen. Kosten und Zeit müssen gegen den Projektfortschritt gelegt werden und entsprechend gesteuert werden können.

Um das Vorankommen im Projekt sichtbar zu machen, werden u.a. Meilensteine für wichtige Etappenziele in den Planungsprozess eingebunden. Diese definieren Kontrollpunkte zwischen oder während laufender Projektphasen. Anhand festgelegter Kriterien muss dann geprüft werden, ob die Ergebnisse aus der beendeten Phase den Anforderungen entsprechen und somit die nächste Phase beginnen kann.<sup>152</sup> Sie haben einen festen Termin und geben dadurch ebenso Aufschluss über das zeitliche Vorankommen im Projekt. Dadurch kann der Einsatz von Meilensteinen der Zeitplanung Struktur verleihen, indem wichtige Termine greifbar festgehalten werden. Meilensteine können zudem als Entscheidungspunkte betrachtet werden, die zum Umdenken in der Planung und einem zugehörigen Richtungswechsel des Projektgeschehens führen können (z. B. weil bei Erreichen von *Meilenstein X* bereits mehr Budget verbraucht wurde als geplant).

Während der Projektfortschritt im Hinblick auf die verstrichene Zeit kontinuierlich verfolgt und visualisiert werden kann, müssen die Kosten der zukünftig noch

---

<sup>151</sup> vgl. *Steimle:2007*, 55.

<sup>152</sup> vgl. *Fiedler:2016*, 73.

benötigten Ressourcen abgeschätzt und geplant werden.<sup>153</sup> Besonders im Rahmen von Festpreisprojekten gilt es, möglichst realistisch zu planen, da dem Auftraggeber ein genau definiertes Projektergebnis zu einem fixen Preis und einem vereinbarten Termin zugesichert wird.<sup>154</sup>

Dreh- und Angelpunkt von Projektplanung und -controlling ist der Projektstrukturplan (PSP), der die Gesamtheit aller Tätigkeiten umfasst welche notwendig sind, um die vereinbarten Projektziele zu erreichen. Visualisiert wird dabei durch eine hierarchische Baumstruktur, welche bis auf Ebene der einzelnen Tätigkeiten oder wenigstens Arbeitspakete heruntergebrochen wird.<sup>155,156</sup>

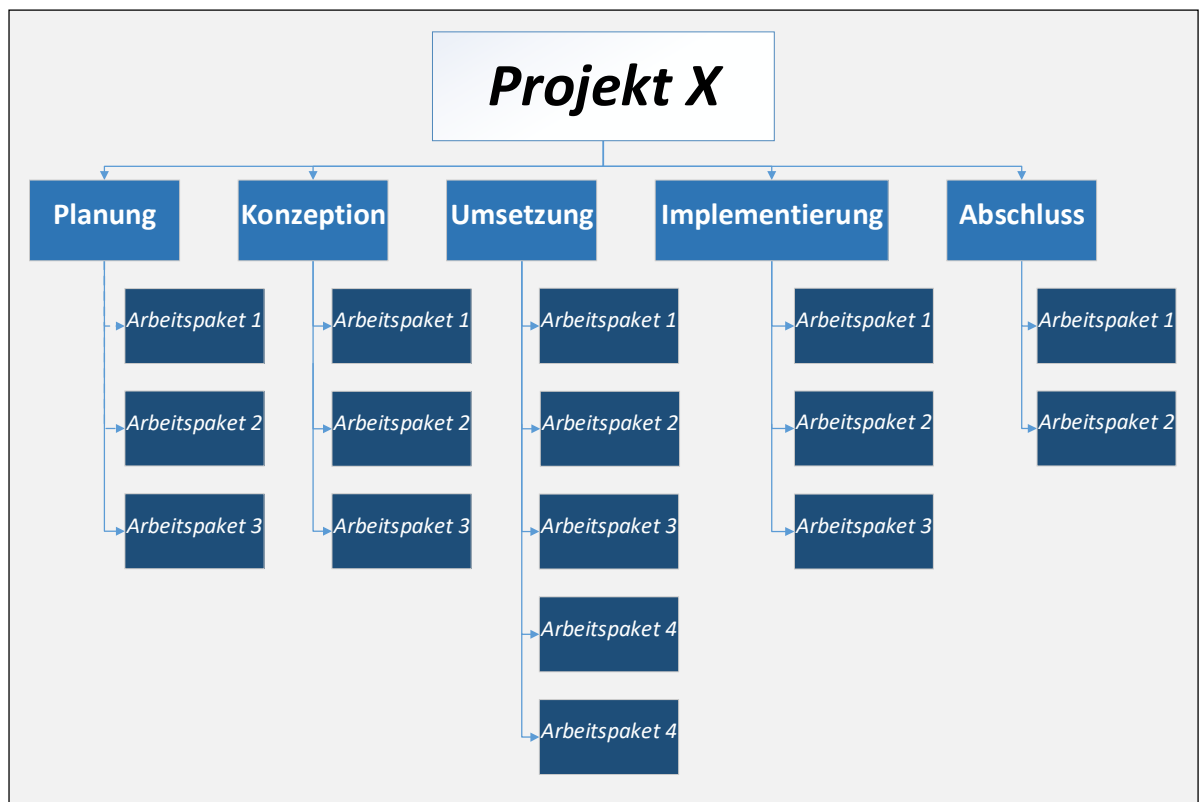


Abbildung 12: Beispiel Projektstrukturplan (Quelle: Eigene Darstellung in Anlehnung an Young:2007, 132).

Zwar wird das Entwickler-Team bei der Detailplanung im Rahmen der Arbeitspaket-Definition miteinbezogen, jedoch obliegt es den für das Projektmanagement und -controlling Verantwortlichen, das Projekt auf Kurs

<sup>153</sup> vgl. Fiedler:2016, 69.

<sup>154</sup> vgl. Steimle:2007, 41.

<sup>155</sup> vgl. Abb. 12.

<sup>156</sup> siehe Fiedler:2016, 73.

zu halten durch „*detaillierte Planung der Termine, Aufwände, Ressourcen und Kosten*“.<sup>157</sup> Die Verantwortung für die Aufwandsschätzungen der einzelnen Arbeitspakete trägt somit das Projektmanagement.<sup>158</sup> Den Verantwortlichen stehen hierzu viele Methoden zur Verfügung, um zu einer Schätzung zu gelangen, z. B.:

- **Einzelschätzung**<sup>159</sup>: Schätzung wird von einem Experten durchgeführt
- **Mehrfachbefragung**<sup>160</sup>: Schätzung wird von mehreren Experten durchgeführt
- **Analogiemethode**<sup>161</sup>: Schätzung auf Grundlage vergleichbarer Projekte

In der Praxis kann es zu Ungenauigkeiten bzgl. der Schätzwerte kommen. Dabei ist häufig die Einbeziehung von mehr oder weniger großen Zeitpuffern die Ursache. Dies kann unbewusst (aus Unkenntnis heraus oder aufgrund einer mangelhaften Spezifikation), aber auch bewusst erfolgen. Bei Letzterem muss genau analysiert werden, worin die Gründe bestehen. Diese können vom Schutz der eigenen Person bzw. der eigenen Leistung bis hin zu Sabotage (Verhinderung oder Beschädigung des Projektes) reichen. Eine gute Möglichkeit bietet daher der Vergleich mit ähnlichen, bereits abgeschlossenen Projekten. Der Vergleich mit bereits abgeschlossenen Projekten setzt jedoch voraus, dass das Projektwissen gesammelt und zukünftigen Projekten zugänglich gemacht wird.<sup>162</sup>

Das bedeutet letztlich, dass die Ergebnisse einer Aufwandsschätzung kritisch behandelt werden sollten. Fiedler schlägt Schätzworkshops vor, in denen bezüglich der Aufwandsschätzung geprüft wird, „*ob sie plausibel, realistisch und vollständig ist*“.<sup>163</sup>

Bei der Einbeziehung unabhängiger (Schätz-)Experten kann es zu Ungenauigkeiten kommen (z. B. weil die individuellen Fähigkeiten der Projekt-Entwickler nicht berücksichtigt werden). Daher kann es sinnvoll sein, sowohl

---

<sup>157</sup> siehe Fiedler:2016, 74.

<sup>158</sup> vgl. Fiedler:2016, 78.

<sup>159</sup> siehe Fiedler:2016, 78f.

<sup>160</sup> siehe Fiedler:2016, 80.

<sup>161</sup> siehe Fiedler:2016, 80.

<sup>162</sup> vgl. Fiedler:2016, 80.

<sup>163</sup> siehe Fiedler:2016, 84.

abhängige als auch unabhängige Experten zu befragen, wobei dies den Aufwand des Schätzverfahrens vergrößert.

Bei projekterfahrenen Unternehmen, die ihre Erfahrungen bündeln und diese entsprechend aufbereiten, sodass neue Projekte darauf zugreifen und von dem bereits vorhandenen Wissen profitieren können, kann die Aufwandsschätzung nach den hier vorgestellten Methoden adäquate Ergebnisse liefern. Z. B. kann bei der Einführung und dem Customizing von Standardsoftware auf vergleichbare Projekte zurückgegriffen werden und ähnliche Arbeitspakete können identifiziert werden, deren damalige Schätzung dann zur aktuellen Schätzung herangezogen werden kann.

Soll jedoch neuartige Software für den Kunden entwickelt werden, wobei dadurch möglicherweise zu Projektbeginn noch viele Unklarheiten herrschen, etwa bei der Frage nach möglichen Technologien, die zum Einsatz kommen sollen, dann kann es sein, dass die Aufwandsschätzungen hohe Ungenauigkeiten aufweisen, weil sie die unbekanntenen Faktoren schlichtweg nicht einbeziehen können. Dann könnte es möglicherweise sinnvoll sein, die Entwickler zuerst recherchieren zu lassen, welche Technologien überhaupt infrage kommen und sie dann Schätzungen vornehmen zu lassen unter Einbeziehung der notwendigen Einarbeitungszeit in die neuen Technologien.

Fehler in der Aufwandsschätzung wirken sich nicht zuletzt auch auf die Termin- und Ressourcenplanung aus und damit ebenso auf die Planung der Projektkosten.

Kurzum: Aufwandsschätzungen und die daraus abgeleiteten Planungen können Projekten mit klarer Vision und klar definierten Zielen, welche sich nicht grundlegend verändern können, einen Leitfaden geben. Ein sorgfältiges Anforderungsmanagement bzw. Change Management hilft, bei sogenannten *Moving Targets*, d.h. sich veränderten Anforderungen bzw. Projektzielen, die Übersicht zu behalten und den Erfolg des Projektes nicht zu gefährden. Allerdings ist dies nur in "vernünftigen" Rahmen handhabbar.<sup>164</sup>

---

<sup>164</sup> vgl. Pavlik:2018, Abschn. „Warum IT-Projekte scheitern“.

## 4.2 Controlling in der agilen Projektwelt

### 4.2.1 Kommunikation

Die Anforderungen an die Kommunikation in einem agilen Projekt unterscheiden sich z.T. in hohem Ausmaß von der klassischen Kommunikation in der spezifikationsorientierten Projektwelt, wobei, wie im Abschnitt 4.1 ausgeführt, die Abstimmung und Dokumentation der Systemanforderungen zwischen Auftraggeber und -nehmer entscheidend ist für das Finden der Produktvision. Dabei ist sie in beiden Welten ein kritischer Faktor für den Erfolg von Projekten.<sup>165</sup>

Systemanforderungen, die im Rahmen der Absprache zwischen dem agilen Team und dem Kunden aufgestellt werden, können in simpler Form dokumentiert werden mithilfe von User Stories<sup>166</sup> auf Story Cards. Eine einfache Dokumentation an dieser Stelle bringt den Vorteil, dass die Kommunikation beider Parteien hinsichtlich der Systemanforderungen erleichtert wird und Missverständnisse vermieden werden können. Die Verwendung von User Stories ermöglicht es zudem, System-Funktionalitäten aus Anwendersicht zu betrachten, sprich, die Frage zu beantworten: *Was kann der Anwender mit dem System tun, wenn Funktion X implementiert wurde?*

Fundierte technische Kenntnisse, wie sie z. B. beim Erstellen von Lasten- und Pflichtenheft notwendig sind<sup>167</sup>, sind aus der Anwender-Perspektive nicht zwingend notwendig, sodass der Inhalt der von Team und Kunden erstellten User Stories prinzipiell auch von denjenigen Projektbeteiligten verstanden werden kann, die keine IT-Experten sind. Damit lassen sich die Story Cards unterschiedlichen Zwecken anpassen, indem diejenigen Informationen abgebildet werden, die im entsprechenden Informationskontext von Belang sind (z. B. kann sich der Informationsbedarf beim Reporting unterscheiden von demjenigen im Rahmen einer Sprint-Planung).

---

<sup>165</sup> vgl. Fiedler:2016, 7.

<sup>166</sup> vgl. Abb.13.

<sup>167</sup> vgl. yeebase media GmbH:2016, Abschn. „Lastenheft vs. Pflichtenheft“.

| Story-Nr.  | Story-Name | Story-Points |
|--|------------|--------------|
| <p>Als &lt;Rolle&gt; möchte ich<br/>&lt;Funktion&gt;, um &lt;Nutzen&gt;.</p> |            |              |
| Status   |            | Priorität    |

Abbildung 13: Beispiel Story Card (Quelle: Eigene Darstellung in Anlehnung an Goll; Hommel:2015, S. 39).

Nicht alle Anforderungen können auf Kunden- oder Anwenderwünschen basieren. So sind etwa Anforderungen an die Systeminfrastruktur, also das gesamte Setup, abhängig von den Anforderungen, die es umzusetzen gilt. Die Frage ist quasi: „*Welche Systeminfrastruktur wird hier benötigt, um die Anforderungen umzusetzen (Datenbankserver, Failover Server, Webserver, etc.)?*“. Auch Softwarearchitekturen und Frameworks spielen hier eine Rolle – diese interessieren den Anwender jedoch in der Regel nicht.

Technische Infrastrukturen werden häufig mit Blockdiagrammen visualisiert, um übersichtliche Darstellungen der vernetzten Hard- und Software zu erzeugen.

Bestimmte Systemanforderungen müssen also bei der Anwendung von Story Cards ausgeklammert und auf „klassischem“ Weg festgehalten werden. Auch die Dokumentation zu den eigentlichen Liefergegenständen, die der Kunde benötigt, um mit den Endergebnissen im laufenden Betrieb zu arbeiten, bleibt im agilen Projekt nicht aus, sodass auch hier nicht auf Betriebs- handbücher, Architekturübersichten, etc. verzichtet werden kann. Damit muss auch in agilen Projekten teilweise detailliert dokumentiert werden, weil sich nicht alles nach agilen Prinzipien lösen lässt.

Die eigentliche Projektdokumentation, die sich mit Aspekten wie dem Aufgabenvorrat, Abhängigkeiten zwischen Arbeitspaketen, etc. beschäftigt, kann mithilfe von Techniken wie dem User Story Mapping durchaus vereinfacht werden. Im Rahmen der bestehenden Systemanforderungen berufen sich die Entwickler beim Umsetzen der Funktionalitäten daher nicht auf ein ausführliches Lasten- und Pflichtenheft, weshalb die Ergebnisse zeitnah getestet und mit der Kundenseite besprochen werden sollten, um sicherzustellen, dass sie auch den Kundenwünschen entsprechen (z. B. durch Review Meetings mit dem Product Owner). Die Mitarbeit des Kunden ist demzufolge essentiell, damit seine Anforderungen auch korrekt an das Entwickler-Team kommuniziert werden. Wenn die Zusammenarbeit zwischen Entwickler und Kundenseite nicht fortlaufend funktioniert, besteht das Risiko, dass das Projektergebnis den Kunden nicht zufriedenstellt. Sind die Systemanforderungen zu Beginn noch unscharf, unvollständig oder gar nicht vorhanden, müssen sie gemeinschaftlich vom Team und dem Kunden entwickelt werden. Der Katalog der fachlichen Systemanforderungen, welche die Systemfunktionalitäten aus Anwendersicht darstellen, wie etwa das Product Backlog bei Scrum, bleibt jedoch nicht statisch, sondern wird kontinuierlich überarbeitet.<sup>168</sup> Ziel ist es daher, den Kunden an der Planung einzelner Projektabschnitte, also der Iterationen, zu beteiligen und das Nachkommen seiner Mitverantwortung bei der Projektplanung und -umsetzung einzufordern.

Das Etablieren von regelmäßigen Feedbackzeitpunkten als Kontrollmechanismus kann dabei die Basis schaffen, auf der die notwendige Zusammenarbeit möglich wird.<sup>169</sup>

---

<sup>168</sup> vgl. 3.4.2.1 Scrum, 47 .

<sup>169</sup> vgl. Abb. 14.

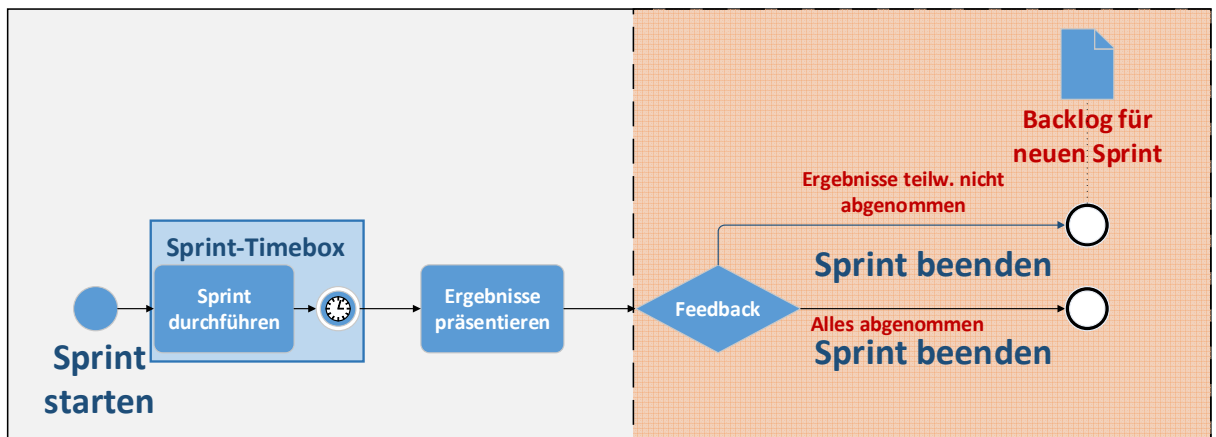


Abbildung 14: Feedback am Beispiel Sprint Review Meeting (Quelle: Eigene Darstellung).

Meetings sind für Projekte grundsätzlich ein notwendiges Werkzeug, um Informationen unter Projektbeteiligten und involvierten Interessengruppen auszutauschen und ggf. Feedback einzuholen. Im agilen Projektkontext dienen die Meetings zudem in hohem Maße der Strukturierung des Arbeitsflusses bzw. der Iterationen. Dabei existieren für unterschiedliche Zwecke unterschiedliche Meeting-Typen, wobei jeder Meeting-Typ klar definiert ist: Zeitpunkt und Zeitrahmen (Timebox), Inhalte (z. B. Planung der nächsten Iteration) und Teilnehmer des Meetings (z. B. Entwickler, Kunde) sind vorgegeben.

Um den Informationsfluss innerhalb des Teams konstant zu halten, wird der aktuelle Projektstatus sowie damit verbundene Hindernisse täglich besprochen in Form von kurzen Meetings (z. B. dem Daily Scrum), bei denen jedes Teammitglied seinen aktuellen Arbeitsstand kurz erläutert<sup>170</sup>.

Dadurch sollen Probleme frühzeitig erkannt und beseitigt werden können.

Für die Iterationsplanung werden im agilen Kontext Planungsmeetings abgehalten, bei denen zwischen Auftraggeber- und Auftragnehmerseite geklärt wird, welche Inhalte in der kommenden Iteration umgesetzt werden sollen. Analog dazu kann die Abnahme während eines Review-Meetings erfolgen. Nach dem Ende einer Iteration kann zudem Feedback erzeugt werden, indem die vergangene Iteration noch einmal analysiert wird hinsichtlich der Frage nach Verbesserungsmöglichkeiten für künftige Iterationen.

<sup>170</sup> vgl. 3.4.2.1 Scrum, 47f.



Dabei rekapitulieren die Meetingteilnehmer z. B. etablierte Verhaltens- und Vorgehensweisen und suchen nach Methoden, diese zu verbessern.<sup>171</sup>

Die Etablierung von verschiedenen Meeting-Typen kann zum Aufbau einer sinnvollen Meeting-Kultur genutzt werden, in der regelmäßig Feedback eingefordert wird, um eine funktionierende Zusammenarbeit zwischen Team und Kunden zu ermöglichen. Die Zeitpunkte und Inhalte der jeweiligen Meetings sind grundsätzlich vorgegeben, können aber natürlich im Sinne eines größtmöglichen Projektfortschrittes<sup>172</sup> angepasst werden. Als Werkzeug zur Projektsteuerung verhindern die Meetings zudem, dass die Länge der Timebox der aktuellen Iteration verändert werden kann: So kann das Team nicht einfach den Sprint um eine Woche verlängern, weil nicht alle Iterationsziele erreicht wurden (was dem Gedanken des regelmäßigen Ausliefern von Ergebnissen widerspricht). Auch die vorgegebenen Zeitfenster der Meetings, der Time-Boxing-Methode folgend, geben der agilen Projektorganisation Struktur und sollen verhindern, dass der Fokus im Meeting sich verschiebt und unnötig Zeit verschwendet wird. Sofern die Teammitglieder dieser Philosophie folgen, können Meetings im Rahmen eines agilen Projekts zu einem effektiven sowie effizienten Informationsfluss mit regelmäßigem Feedback an der richtigen Stelle führen und somit zum Projekterfolg beitragen.

Ein weiterer kritischer Faktor entsteht aus der Selbstorganisation des agilen Teams. Während Dietmar Prudix die Bedeutung einer klaren Regelung von Aufgaben und klaren Kommunikationswegen betont, kann dies in agilem Kontext durch die Selbstorganisation der Entwickler-Teams und der gemeinsamen Verantwortung bezüglich ausgelieferter Software allerdings problematisch sein.<sup>173</sup> In einem Scrum-Team werden alle Individuen, unabhängig ihrer Fähigkeiten, als *Entwickler* bezeichnet und nicht etwa als *UI- oder Datenbankspezialist*, *Tester* oder *Architekt*. Das Team muss daher hinreichend interdisziplinär<sup>174</sup> sein, um sämtliche Aufgaben selbstverantwortlich lösen zu können, übernimmt aber nach außen hin (z. B. dem Product Owner gegenüber) die Verantwortung gemeinschaftlich. Dementsprechend wichtig

---

<sup>171</sup> Gemäß einer Retrospektive, wie sie in 3.4.2.1 *Scrum*, 47 beschrieben ist.

<sup>172</sup> vgl. *McManus:2004*, 9.

<sup>173</sup> vgl. *Prudix:2016*, 35.

<sup>174</sup> vgl. *Maximini:2018*, 193.

ist es, dass sich die Teammitglieder untereinander verständigen können, unabhängig von ihren spezifischen Kenntnissen. Auch ist es wichtig, dass die Kommunikationskultur in einem agilen Projekt das richtige Maß an Fehlertoleranz und Offenheit für Änderungen entwickeln kann. Schließlich können sich Projektziele jederzeit ändern und auch Arbeitsweisen sowie aufgestellte Projekt-Spielregeln müssen leicht veränderbar sein, um zeitnah auf neue Erkenntnisse oder Ereignisse reagieren zu können (was z. B. in einem stark regulierten Umfeld nicht gelten sollte). Reflexionsvermögen und der Mut zu Veränderungen sind wichtige Eigenschaften, die eine „Fehler- und No-Blame-Kultur“ nach Dombrowski auszeichnen.<sup>175</sup>

Der Austausch unter den Entwicklern muss daher kontinuierlich stattfinden, weswegen sich beispielsweise Daily Scrums bewährt haben. Generell gilt die teaminterne Kommunikation als besonders kritischer Faktor für agile Projekte, weshalb der Team-Raum als wichtiger Schauplatz gemeinsamen Arbeitens verstanden werden kann.<sup>176</sup> Sind die Mitglieder agiler Projektteams örtlich verstreut, fehlt die Möglichkeit eines physischen Team-Raums, sodass das gemeinsame Zusammenarbeiten virtuell stattfinden und entsprechend technisch unterstützt sein muss.<sup>177</sup> Da jedoch in dieser Hinsicht die Kommunikation eingeschränkt wird, ist im Einzelfall kritisch zu bewerten, ob eine virtuelle Vorgehensweise die Grundgedanken des Agilen Vorgehens ad absurdum führen würde.

Während agile Vorgehensmodelle wie Scrum grundsätzlich durch Übersichtlichkeit und Einfachheit bestechen und somit von den Projektmitgliedern in der Theorie schnell erlernt werden können, muss dagegen betont werden, dass eine offene Kommunikationskultur, wie sie hier beschrieben ist, nicht spontan zustande kommen kann, sondern mühsam aufgebaut werden muss, und zwar kontinuierlich, dem Gedanken des KVP folgend. Fraglich ist daher, inwiefern diese in einem Umfeld gedeihen kann, dessen Grundsatzwerte den Agilen Werten widersprechen, z. B. in einem nichtagil geprägten

---

<sup>175</sup> vgl. *Dombrowski:2015*, 76.

<sup>176</sup> vgl. *Maximini:2018*, 14f.

<sup>177</sup> vgl. *InLoox GmbH:2018b*.

Unternehmen, das obendrein keine oder nur geringe Toleranz für Fehler aufweist.

#### 4.2.2 Dokumentation

Auch die Projektdokumentation eines agilen Projekts weist einige Unterschiede auf zur klassischen Projektdokumentation. Auf das Wesentliche beschränkt, sollten alle Informationen in möglichst einfach verständlicher Sprache festgehalten werden.

Statt beispielsweise mittels ausführlicher Lasten- und Pflichtenhefte werden die Systemanforderungen in einfach strukturierten Katalogen, wie dem bereits erläuterten Product Backlog aus Scrum, erfasst und priorisiert. Da die Systemanforderungen aus Nutzersicht beschrieben werden, können diese ohne großen Zusatzaufwand zu Visualisierungszwecken verwendet werden, zum Beispiel in Form von User Story Cards an einer User Story Map.<sup>178</sup> Das ermöglicht den Betrachtern des Boards, die Systemanforderungen im Zusammenhang wahrzunehmen und eine gemeinsame Produktvision vor Augen zu haben (z. B. während der Planung des kommenden Sprints).

Da sich die Projektziele im Verlauf eines Projekts ändern können, ist es ohnehin sinnvoll, den aktuellen Projektstatus regelmäßig zu visualisieren und allen Projektbeteiligten zugänglich zu machen. Hindernisse und Engpässe können somit zeitnah erkannt werden. Im Rahmen des Projektreportings<sup>179</sup> können solche Visualisierungen daher Wiederverwendung finden.<sup>180</sup>

Eine zu Projektbeginn recht grobe Beschreibung der Systemanforderungen kann nach der Methode "Enough Design Up Front" im Lauf des Projekts erweitert und verfeinert werden, sodass zum Projektabschluss eine detaillierte Beschreibung der Anforderungen vorliegt, die mit dem Endergebnis verglichen werden kann.<sup>181</sup> Im Rahmen einer Story Map findet sich dieses Prinzip wieder im späteren Aufbrechen einer großen User Story (Epic) in kleinere User Stories.<sup>182</sup>

---

<sup>178</sup> vgl. 3.4.2.1 Scrum, 43f.

<sup>179</sup> vgl. domendos:2018.

<sup>180</sup> vgl. Cohn:2016.

<sup>181</sup> vgl. Goll; Hommel:2015, 104.

<sup>182</sup> vgl. Abb. 15.

|   |                |           |
|---|----------------|-----------|
| EPIC-Nr.  | EPIC: Checkout | 10 SP     |
| <p>Als &lt;Kunde&gt; möchte ich &lt;eine Bestellung aufgeben&gt;, um &lt;meine Waren zu erhalten&gt;.</p> |                |           |
| Status  |                | Priorität |

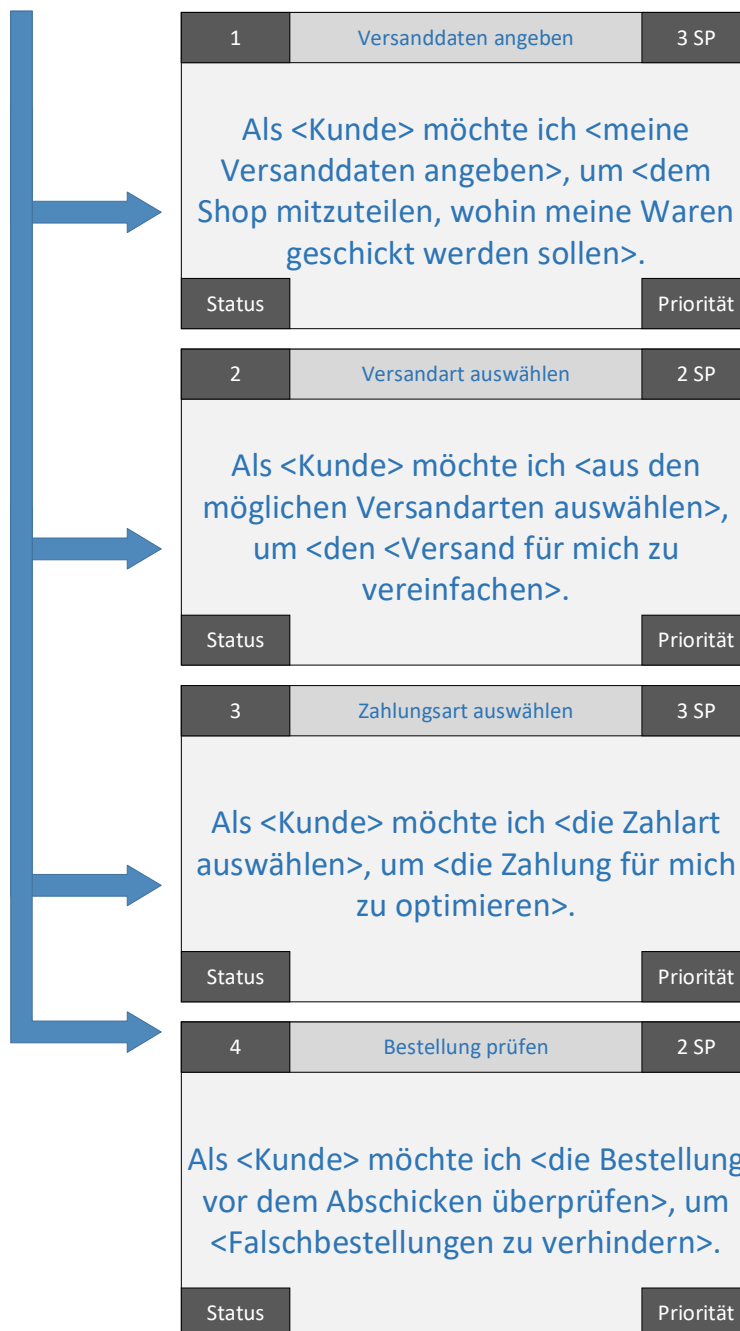


Abbildung 15: Visualisierung einer Epic-User-Story mittels Story Cards (Quelle: Eigene Darstellung in Anlehnung an Goll; Hommel:2015, S. 39).

Im Gegensatz zur klassischen Projektdokumentation muss in agilen Projekten zwar nicht übermäßig detailliert dokumentiert werden (z. B. durch Lasten- und Pflichtenheft), allerdings erfordert die Projektdokumentation in agilen Projekten sorgfältige Pflege aufgrund ihres dynamischen Charakters durch Aktualisierungs-, Synchronisierungs-, und Versionierungsaufgaben, die in agilen Projekten entsprechend häufig anfallen.<sup>183</sup> Aus diesem Grund bietet es sich hier an, die entsprechenden Projektdokumente offen zu halten und weniger statisch, als es im spezifikationsorientierten Projektleben Sinn hat.

Dies umzusetzen erfordert von allen Projektbeteiligten das nötige Verständnis für die Wichtigkeit der Dokumentation, weshalb sie neben der Kommunikationskultur als ein weiterer kritischer Erfolgsfaktor für agiles Vorgehen verstanden werden sollte, besonders vor dem Hintergrund des Agilen Manifestes, welches nicht missinterpretiert werden sollte: Die Aussage „*Funktionierende Software mehr als umfassende Dokumentation*“<sup>184</sup> darf also nicht dazu führen, dass die Dokumentation des Projektgeschehens vernachlässigt wird, sondern stattdessen im Rahmen der hohen Dynamik agiler Projekte entsprechend einfach gehalten wird, z. B. durch:

- Vorlagen, z. B. für Meetingprotokolle, User Stories etc.
- Dokumentationsregeln, z. B. für Benennungsschemata
- Versionspflege, z. B. für das Product Backlog

#### 4.2.3 Steuerung

Zur Messung und Visualisierung des Projektfortschritts können teamintern z. B. verschiedene Burndown-Charts sinnvoll genutzt werden. Beispielsweise ein Burndown-Chart für den aktuellen Sprint,<sup>185</sup> der regelmäßig, zum Beispiel beim Daily Scrum, aktualisiert wird.<sup>186</sup>

*Abbildung 16* zeigt dazu zwei Verläufe: Während bei der Planung von einem linearen Verlauf ausgegangen wird (grau), was bedeuten würde, dass das Projektteam eine konstante Geschwindigkeit, also Velocity, beibehalten kann und damit die Inhalte des Sprint Backlogs gleichmäßig abgearbeitet werden,

---

<sup>183</sup> vgl. *Rau:2016*, 24.

<sup>184</sup> siehe *the Agile Manifesto authors:2018a*.

<sup>185</sup> vgl. *Abb. 16*.

<sup>186</sup> vgl. *Goll; Hommel:2015*, 97.

zeigt sich im realen Verlauf (rot), dass die Arbeitsgeschwindigkeit vom geplanten Verlauf variiert. Verzögerungen könnten z. B. auf Probleme und Hindernisse zurückzuführen sein, die das Team zeitweise verlangsamen lässt. Dagegen kann eine (vermeintlich) höhere Arbeitsgeschwindigkeit etwa durch zu hohe Schätzungen des Arbeitsaufwandes oder einfach durch im Verlauf des Projektes getroffene Vereinfachungen entstehen.

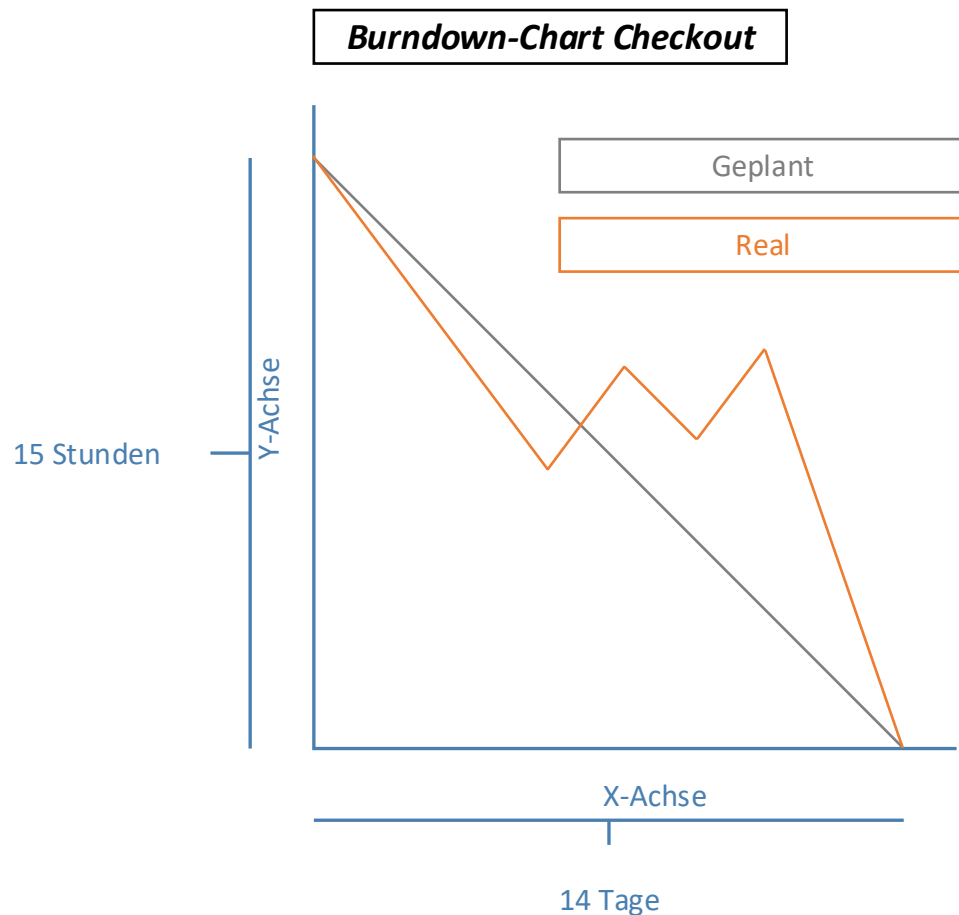


Abbildung 16: Beispiel eines Burndown-Charts für die Epic-Story „Checkout“ (Quelle: Eigene Darstellung in Anlehnung an itemis AG:2018).

Noch detaillierter könnte der Arbeitsfortschritt gemessen werden, wenn alle User Stories des aktuellen Sprints einzeln in einem Burndown-Chart abgebildet werden würden. Dabei würden die noch zu erledigenden Aufgaben innerhalb der jeweiligen User Stories abgebildet in Abhängigkeit von der noch verbleibenden Zeit. Damit lässt sich z. B. nach Scrum abschätzen, ob bis zum Sprint-Ende alle User Stories des Sprint Backlogs umgesetzt werden können.

Auch übergeordnet muss das agile Projekt gesteuert werden. Projektleiter im klassischen Sinne sind dort zwar nicht vorgesehen, würden sie dem

Selbstverantwortungsanspruch des Teams widersprechen. Die Funktionen des Projektleiters<sup>187</sup> dürfen aber nicht einfach wegfallen, sondern müssen an anderer Stelle im Projekt integriert werden, damit Projektplanung und -umsetzung funktionieren, d.h. die Projektziele erreicht werden. Während gemäß *Abbildung 17* der Projektmanager dafür sorgt, dass die einzelnen Teammitglieder ihren Aufgaben nachkommen, damit das gewünschte Produkt entsteht und die Dimensionen *Zeit*, *Kosten* sowie *Leistung* steuert, muss sich das Team im agilen Projekt<sup>188</sup> selbst organisieren, sodass am Ende das Produkt entsteht, das der Product Owner haben möchte, wie es *Abbildung 18* verdeutlichen soll. Damit muss der Product Owner auch steuernd handeln, um sicherzustellen, dass das Team in die richtige Richtung arbeitet.

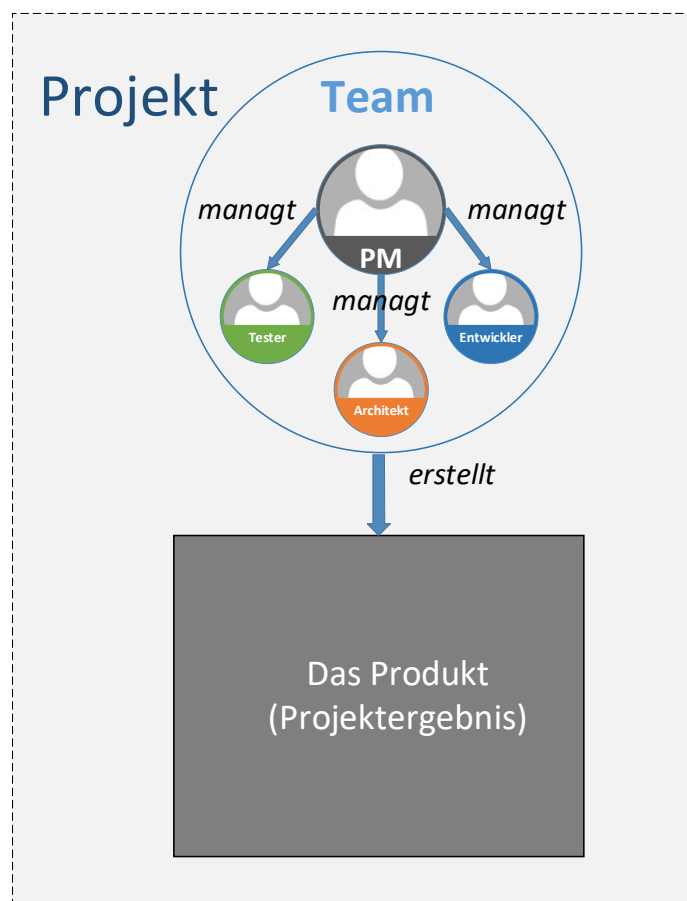


Abbildung 17: Verantwortung in klassischer Projektsituation (Quelle: Eigene Darstellung).

<sup>187</sup> vgl. Abb. 17.

<sup>188</sup> vgl. Abb. 18.

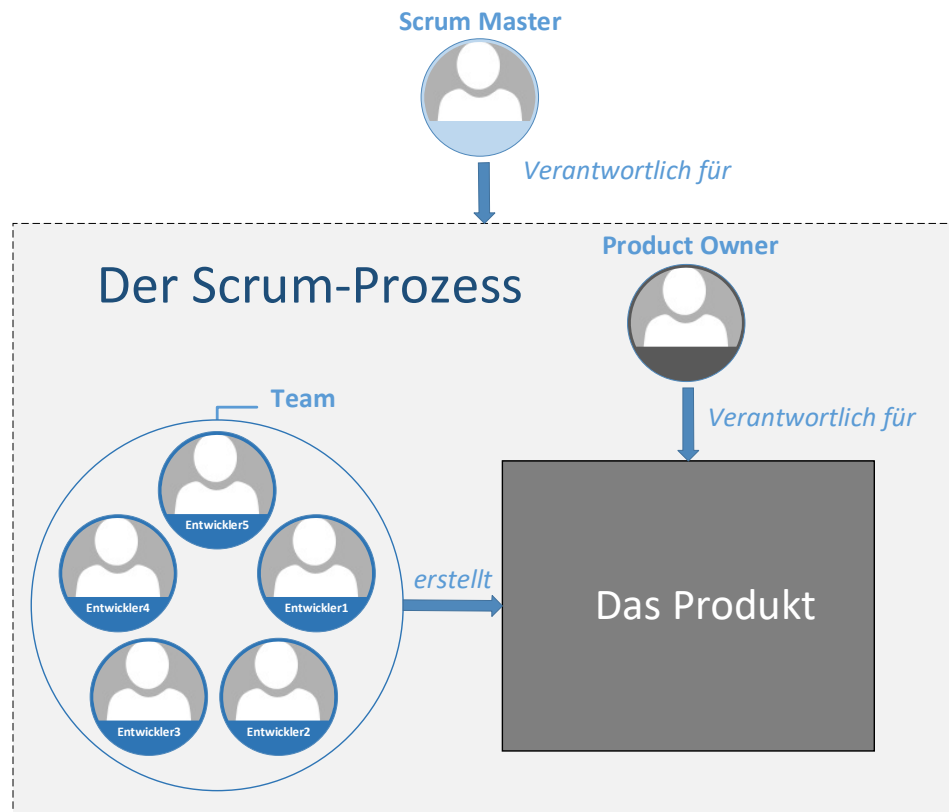


Abbildung 18: Verantwortung in Scrum (Quelle: Eigene Darstellung).

So ergänzen sich im Scrum-Kontext die einzelnen Verantwortungsbereiche, die Scrum Master, Product Owner und das Team übernehmen:

Der Scrum Master zeigt sich verantwortlich für den Scrum-Prozess und sorgt dafür, dass alle Beteiligten gemäß Scrum-Regeln vorgehen, er beseitigt Hindernisse für das Entwickler-Team und fungiert als Vermittler zwischen den Stakeholdern.

Der Product Owner hingegen übernimmt die Verantwortung für das Produkt: Er sorgt dafür, dass das Team das richtige Produkt entwickeln kann, indem er die Systemanforderungen im Product Backlog sinnvoll priorisiert und fortlaufend aktuell hält. Damit ist der Product Owner an der Steuerung des Teams beteiligt, indem er die Projektziele, insbesondere dann, wenn sie sich geändert haben, mit dem Product Backlog abgleicht. Werden Änderungswünsche oder neue Einträge für das Backlog an ihn herangetragen, so muss er entscheiden, ob das Backlog verändert werden muss. Auch den Geschäftswert des Produkts (aus Kundensicht) sollte der Product Owner im Laufe des Projekts im Auge behalten, indem die bereits gelieferten Inkremente entsprechend bewertet



werden.<sup>189</sup> Alternativ zur Priorisierung mittels Product Backlog bietet das Kanban-System die Möglichkeit, Systemanforderungen in Service-Klassen einzuteilen, um die Differenzierung von Arbeitspaketen z. B. nach Risiken vorzunehmen.<sup>190</sup> Dabei könnte der Produktverantwortliche, dem Vorschlag von Leopold und Kaltenecker folgend, spezifische Merkmale definieren, nach denen die einzelnen Service-Klassen unterschieden werden sollen: z. B. *Kundenzufriedenheit* und *Systemstabilität*. Arbeitspakete, die für mehr Systemstabilität sorgen, fallen dann beispielsweise in die Klasse *Muss-Funktionen*, weil der Produktverantwortliche das Risiko von Systemausfällen als kritischer ansieht als eine niedrigere Kundenzufriedenheit, die sich möglicherweise ergibt, wenn nicht alle *Kann-Funktionen* ausgeliefert werden. Für das Team würde das bedeuten, dass zuerst alle Aufgaben umgesetzt werden, die für Systemstabilität sorgen, priorisiert nach Höhe des Risikos bei Nichtumsetzung, bevor die übrigen Funktionalitäten entwickelt werden können.

Die Organisation der projektzielbezogenen Arbeiten, etwa die Analyse der Systemanforderungen, das Entwickeln von Funktionalitäten sowie das Testen derselben, obliegt ausschließlich dem Entwicklerteam. Eine wesentliche Aufgabe dabei ist die angemessene Limitierung der Arbeitspakete je Zyklus, um den Entwicklungsprozess nicht zu überfrachten und eine dem Zyklus entsprechende Anzahl an Funktionalitäten zu entwickeln. Gemäß Scrum limitiert das Entwickler-Team die Sprint-Ziele mithilfe des priorisierten Backlogs, indem es so viele Aufgaben ins Sprint Backlog übernimmt, wie es nach eigener Einschätzung innerhalb des Sprints umsetzen kann (unter Zuhilfenahme der geschätzten Story Points der einzelnen User Stories – diese im Team geschätzten relativen Zahlenwerte werden zur Messung der Velocity verwendet<sup>191</sup>). Eine ähnliche Limitierungsmethode liefert die WiP-Technik aus dem bereits erläuterten Kanban-System.

Die innere Steuerung des Projekts muss also von allen beteiligten Parteien gemeinsam übernommen werden: Der übergeordnete Koordinator (bei Scrum der Scrum Master) schafft das störungsfreie Arbeitsumfeld, welches das Team benötigt, um seine Verpflichtung erfüllen und Selbstverantwortung übernehmen

---

<sup>189</sup> vgl. Maximini:2018, 103f.

<sup>190</sup> vgl. Leopold; Kaltenecker:2013, 53f.

<sup>191</sup> vgl. Maximini:2018, 203.

zu können, während der Produktverantwortliche (Product Owner) dafür Sorge trägt, dass das Team das richtige Produkt entwickelt.

Um eine effiziente Interaktion zwischen Scrum Master, Product Owner und dem Team zu gewährleisten, bietet es sich an, dass alle Parteien zumindest zeitweise am selben Ort arbeiten. Sind alle Projektmitglieder untereinander erreichbar, können auftretende Probleme oft schneller gelöst werden. Besonders bei Entwickler-Teams bewährt sich dieser Vorteil, wenn sie über einen eigenen Team-Raum verfügen, der sowohl Möglichkeiten für gemeinsames Arbeiten, als auch für Stillarbeiten liefert.<sup>192</sup>

Indem die Projektbeteiligten iterative Selbststeuerungsmechanismen entwickeln (z. B. Qualitätszirkel mit regelmäßigen Analysen der eigenen Arbeitsweise zur Offenlegung von Missständen<sup>193</sup>), welche sich am PDCA-Kreis orientieren, kann der Projektfortschritt kontinuierlich überwacht und reflektiert werden, ohne dass permanent ein Projektleiter von außen auf das Projekt einwirken muss. Jedoch sollte hierbei angemerkt werden, dass die Selbststeuerung kritischer wird, je größer das Projektteam ist, weshalb dieses Prinzip nicht auf alle Projektsituationen ohne Weiteres übertragbar ist.

Nachfolgend wird für die Verantwortungsbereiche des Product Owners, des Scrum Masters und des Teams jeweils ein Beispiel genannt, wie ein Steuerungsmechanismus in einem agilen Projekt etabliert werden kann:

- 1 **Product Owner:** Da der Product Owner den Geschäftswert des Produktes ständig bewertet, kann er z. B. nach jedem Sprint entscheiden, ob sich das Projekt immer noch lohnt.
- 2 **Scrum Master:** Auch der Scrum Master kann Vorgehensweisen etablieren, um den Scrum-Prozess zu überwachen, z. B. bei Hindernissen und Problemen, die im Daily Scrum geschildert werden, indem er im Anschluss daran mit den betroffenen Teammitgliedern darüber spricht.
- 3 **Team:** Eine teaminterne regelmäßige Analyse von Burndown-Charts kann dazu führen, dass Engpässe erkannt werden, die aus

---

<sup>192</sup> vgl. Maximini, 14f.

<sup>193</sup> vgl. Wirtschaftslexikon24:2018.

problematischen Aufgabentypen resultieren. Folglich könnte in gemeinsamer Absprache mit dem Kunden eine Entscheidung getroffen werden, ob das Know-How für das Lösen dieser Aufgaben noch in der laufenden Iteration erarbeitet werden soll. Das würde im Zweifel allerdings bedeuten, dass nicht alle ursprünglich geplanten Aufgaben in diesem Sprint abgeschlossen werden könnten und das Sprint Backlog entsprechend geändert werden müsste.

Die erfolgreiche Steuerung des Projektgeschehens setzt demnach voraus, dass der jeweilige Kontext, in dem sich das Projekt befindet, von allen Parteien korrekt erfasst wird und dementsprechende Methoden etabliert werden, wie zum Beispiel Qualitätszirkel.

### 4.3 Vergleich der Anforderungen

Nachfolgend werden die kritischen Anforderungen beider Projektwelten in den Phasen *Analyse & Planung*, sowie *Umsetzung & Abschluss* gegenübergestellt. Danach folgt eine Zusammenfassung typischer Merkmale, die die Projektwelten charakterisieren sowie die Vor- und Nachteile, die sich daraus ergeben.

#### 4.3.1 Analyse & Planung in der spezifikationsorientierten Projektwelt

Da verlässliche Schätzungen in der spezifikationsorientierten Projektwelt bereits zum Projektstart notwendig sind, um eine brauchbare Planung entwickeln zu können, sind die dazu benötigten Informationen essentiell. Folglich müssen Systemanforderungen des Kunden kommuniziert und vom Auftragnehmer korrekt erfasst werden, sodass beide Parteien dieselbe Vorstellung vom Endprodukt haben und sich auf die Projektziele einigen können. Wenn sich die Systemanforderungen im Verlauf des Projekts nicht verändern, ist es zwar nicht notwendig, in zyklischen Abständen erneut darüber zu reden (wie etwa im Sprint Planning Meeting bei Scrum), jedoch muss es dem Kunden möglich sein, diese korrekt an den Auftragnehmer zu vermitteln. Sind beispielsweise Änderungen nötig oder gewünscht, sind effiziente Kommunikation und ein sauberes Change Management notwendig, damit die Planung möglichst zeitnah angepasst werden kann.

Darüber hinaus sollte eine gewisse Informationsgüte, die im Qualitätsmanagement z. B. über Regelungen zur Datenerhebung sichergestellt werden kann, angestrebt werden, um aus Analysen- und Planungsüberlegungen brauchbare Ergebnisse zu erhalten. So könnten Aufwandsschätzungen zu komplexen Arbeitspaketen z. B. von Entwicklern erhoben werden, die bereits Erfahrungen mit ähnlichen Arbeiten sammeln konnten.

#### 4.3.2 Analyse & Planung in der agilen Projektwelt

Demgegenüber stehen die Anforderungen an die *Analyse & Planung* bei agilen Projekten. Die in der agilen Welt üblichen kurzen Planungs- und Umsetzungshorizonte profitieren von einer guten Selbsteinschätzung aller Teammitglieder, da sie selbst die Verantwortung für ihre Arbeit übernehmen. Dementsprechend muss jedes Teammitglied dazu in der Lage sein abzuschätzen, welche Aufgaben es in der zu planenden Iteration erfüllen kann. Da die Verantwortung gemeinschaftlich getragen wird, ist ein starker Teamgeist eine weitere Anforderung, die ein agiles Projektteam entwickeln sollte. Konflikte sollten zwar nicht gesucht, jedoch auch nicht umgangen werden. Eine offene Kommunikationskultur ist unabdingbar für ein langfristig produktives Arbeitsklima, in dem sich jeder Projektbeteiligte engagieren kann und will.

#### 4.3.3 Umsetzung & Abschluss in der spezifikationsorientierten Projektwelt

Die Umsetzung des Projektplans in der spezifikationsorientierten Welt erfordert schlicht das Befolgen des Plans. Da allerdings Projektpläne in der Regel nicht eins zu eins umgesetzt werden können, weil in der Praxis eine Vielzahl unterschiedlicher, z. T. sogar unbekannter Faktoren das Projektgeschehen unmittelbar oder indirekt beeinflussen, sind Planabweichungen und Modifikation der Planung unvermeidbar. Daher muss bei der Umsetzung permanent geprüft respektive gemessen werden, inwiefern das Projekt im Rahmen des Plans vorankommt. Das bedeutet zum einen, dass die verantwortlichen Personen, z. B. Projektmanager und Controller, die Zeit dazu haben müssen, die aussagekräftigen Faktoren mithilfe von entsprechenden Methoden messen und analysieren zu können. Zum anderen verdeutlicht es aber auch die Wichtigkeit einer funktionierenden Kommunikation im Projekt, sodass etwa das Einholen von Informationen von Teammitgliedern reibungslos abläuft,

etwa „*Wie viele Stunden hat Teammitglied X in Arbeitspaket Y bereits investiert und wieviel % der Arbeit sind erledigt?*“.

Planungsänderungen müssen unbedingt rechtzeitig kommuniziert werden. Außerdem sollten *Umsetzung & Abschluss* auch realistisch sein.

#### 4.3.4 Umsetzung & Abschluss in der agilen Projektwelt

Analog erfordert ebenso die Umsetzung agiler Projekte eine reibungslose Kommunikation, sowohl teamintern, als auch zwischen Team und Kunden, um die Iterationsplanung erfolgreich umsetzen zu können. Möglicherweise kann eine unterteilte Planung auf Projekt- sowie Iterationsebene sinnvoll sein, etwa bei Projekten mit langer Laufzeit. Während bei Scrum die Sprints vom Team in Absprache mit dem Product Owner geplant werden, könnte der Product Owner eine übergeordnete Planungsebene einführen, um beispielsweise die Entwicklung des Produkts mit den eigentlichen Zielen (z. B. Vereinfachung unternehmensinterner Buchhaltungsprozesse durch das Produkt) abzugleichen. Somit könnten rechtzeitig wichtige Entscheidungen getroffen werden, beispielsweise, ob das Projekt insgesamt verlängert werden sollte, weil die angestrebte Vereinfachung innerhalb der ursprünglich geplanten Gesamtprojektdauer (Summe aller Sprints) nur noch z.T. realisiert werden kann.

Die Bereitschaft zu Änderungen grundsätzlich sowie die Entscheidung, ob eine Änderung jeweils durchgeführt werden muss, erfordern einen gewissen Mut. Die kontinuierliche Verfeinerung der Planung und damit die Konkretisierung der Projektvision geschieht gemeinschaftlich. Selbst, wenn das Projekt in einer großen definierten Timebox abläuft, kann es theoretisch frühzeitig abgeschlossen werden, ohne dass es ein Projektabbruch wäre. Da regelmäßig Ergebnisse, z. B. lauffähige Software, ausgeliefert wird, ist ein Abschluss zu früheren Zeitpunkten als dem letztmöglichen vielleicht sogar sinnvoll, wenn beispielsweise die Projektvision frühzeitig konkretisiert und entsprechend umgesetzt werden konnte. Analog können natürlich weitere Sprints an das ursprünglich geplante Projektende angehängt werden, wenn beispielsweise der Kunde ein weiteres Release (Inkrement) haben möchte.

#### 4.3.5 Zusammenfassung der Unterschiede

Nachfolgende Tabelle fasst die hier beschriebenen möglichen Unterschiede, die in den beiden Projektwelten vorkommen, für die Bereiche *Kommunikation*, *Dokumentation* und *Steuerung* zusammen.

Tabelle 6: Unterschiede in den Bereichen Kommunikation, Dokumentation & Steuerung.

| <b>PERSPEKTIVE</b>   | <b>Spezifikationsorientierte Projektwelt</b>   | <b>Agile Projektwelt</b>  |
|----------------------|--|---|
| <i>Kommunikation</i> | Eher klare Kommunikationswege durch Rollenbeschreibung & klare Verantwortungsbereiche:<br>z. B. <i>PM für Planung &amp; Steuerung, Controller für Fortschrittmessung &amp; Unterstützung</i> | Offene Kommunikationskultur, in der Verantwortung geteilt wird:<br>z. B. <i>Scrum Master -&gt; Prozess, Product Owner -&gt; Produkt, Team -&gt; Arbeitspakete</i> |
|                      | Fehlertoleranz muss nicht zwangsweise gegeben sein: Eher vorgegebene Vorgehensweisen<br>z. B. <i>in reguliertem Projektumfeld</i>  | Fehlertoleranz als kritisches Merkmal: Eher kreative Vorgehensweisen<br>z. B. <i>in Projekten mit neuen Technologien</i>  |
| <i>Dokumentation</i> | Gefahr von zu aufwändiger Dokumentation  | Gefahr von zu einfacher Dokumentation<br>z. B. <i>bestimmte Dokumente werden einfach nicht gemacht</i>  |
|                      | Eher schwergewichtig<br>z. B. <i>Lastenheft, Pflichtenheft, Spezifikation</i>  | Eher leichtgewichtig<br>z. B. <i>Product Backlog, Sprint Backlog</i>  |
| <i>Steuerung</i>     | PM übernimmt Planung<br>z. B. <i>PSP, Organisation/Visualisierung Aufwandsschätzung</i>  | Team übernimmt z. T. Planung<br>z. B. <i>Sprint Planung</i>   |
|                      | Eher eingeschränkte Selbstorganisation<br>z. B. <i>Zuteilung Arbeitspakete von PM</i>  | Konzept der Selbstorganisation<br>z. B. <i>Daily Scrum: Fortschritts- und Problemanalyse mittels Burndown Charts</i>  |

Die unterschiedlichen Eigenschaften von agiler und klassischer Methodik lassen darauf schließen, dass beide gewisse Vor- und Nachteile aufweisen.

Je nachdem, wie der Kontext eines möglichen Projektvorhabens aussieht, müssen also die auftretenden Vor- und Nachteile hinsichtlich ihrer Konsequenzen bewertet werden, um das richtige Vorgehensmodell zu finden.

Nachfolgend werden stichpunktartig die jeweiligen Vor- und Nachteile aufgeführt.

#### 4.3.5.1 Spezifikationsorientierte Projektwelt

##### Vorteile:

- etablierte Kommunikationswege
- Klare Verantwortung: Fehler können leicht den Verantwortlichen zugeordnet werden
- PM übernimmt Steuerung: Team kann sich auf Inhalt (=Arbeitspakete) konzentrieren
- Umfangreiche Dokumentation als Grundlage für Umsetzung

##### Nachteile:

- Weisungsbefugnisse sorgen für Mehraufwand, z. B. Beschwerdefall:



*statt*



- Eher wenig Raum für Kreativität und Kritikoffenheit / Fehlertoleranz
- Wenig Selbstbestimmung im Team, da eher fremdbestimmt:  
Negative Einflüsse auf das zu entwickelnde System
- Gefahr unnötiger Details in der Dokumentation

#### 4.3.5.2 Agile Projektwelt

##### Vorteile:

- Team organisiert sich selbst: Realistischere Ergebnisse, z. B. bei Aufwandsschätzung, möglich
- Da keine Weisungsbefugnis durch PM, offene Diskussionen möglich
- Raum für Kreativität bei Lösungsfindung

Nachteile:

- Selbstorganisation erfordert Erfahrung und Disziplin jedes Einzelnen
- Keine festgelegten Rollen innerhalb des Entwicklerteams: Gefahr von Machtkämpfen; Verantwortung muss geteilt werden
- Dokumentation kann schnell vernachlässigt werden



## 5 Agile Anwendung in der Praxis

---

### 5.1 Typische Project-Constraints in den beiden Projektwelten

Die in Abschnitt 4 behandelten Unterschiede in den beiden Projektwelten machen deutlich, dass nicht jedes Vorgehensmodell für jeden Projektkontext ideal ist. Mithilfe der Vor- und Nachteile kann abgewogen werden, ob eine agile oder spezifikationsorientierte Vorgehensweise im Einzelfall sinnvoller ist (oder auch eine Kombination aus beidem). Auch die möglichen *Project-Constraints*, also gewisse Zwänge bzw. Rahmenbedingungen, welche das Projekt auf die eine oder andere Weise beeinflussen und mit denen das Projektteam einfach "leben" muss, sollten im Vorfeld berücksichtigt werden. Daher werden nachfolgend einige Project-Constraints angeführt. Der anschließende Teil des 5. Abschnitts beschäftigt sich mit zwei verschiedenen Praxissituationen und stellt Möglichkeiten vor, um agile Projektarbeit zu realisieren:

1. Agile Projektarbeit in einer klassischen Unternehmensstruktur
2. Skalierung von agilen Modellen für Großprojekte

Für die Situation 1 wird die Methode einer Agilen Kapsel erläutert, während für Situation 2 eine Auswahl an Frameworks zur Skalierung von agilen Vorgehensmodellen vorgestellt wird.

#### 5.1.1 Projektspezifische Project-Constraints

*Tabelle 7: Project-Constraint „Komplexität“*

| <b>Hohe Komplexität</b>   | <b>Beispielgründe</b>  |
|---|--|
| Auftraggeber & -nehmer müssen Anforderungen gemeinsam erarbeiten. | <ul style="list-style-type: none"><li>• Unscharfe Anforderungen</li><li>• Viele Änderungen zu erwarten</li><li>• Keine vergleichbaren Projekte</li></ul> |

→ *Agiles Vorgehen mit kurzen Planungshorizonten*

Tabelle 8: Project-Constraint „Projektgröße“

| Projektgröße   | Beispielgründe   |
|--|--|
| Projekt muss in sinnvolle Teilprojekte gegliedert werden, um die daraus entstehende Komplexität <sup>194</sup> beherrschbar zu machen. | <ul style="list-style-type: none"> <li>• Viele Teilprojekte</li> <li>• Viele Subteams</li> </ul> |

→ Manche Teilprojekte agil, manche spezifikationsorientiert umsetzen

Tabelle 9: Project-Constraint „Zeit“

| Zeitdruck           | Beispielgründe  |
|---------------------|---|
| Iteratives Vorgehen | <ul style="list-style-type: none"> <li>• Schnelle Ergebnisse</li> </ul> |

→ Spezifikationsorientierte Ansätze oder agil

Tabelle 10: Project-Constraint „Vertragsart“

| Festpreis                                      | Beispielgründe   |
|--|--|
| Widerspruch zum Agilen Manifest <sup>195</sup> | <ul style="list-style-type: none"> <li>• Definierter Leistungsumfang</li> <li>• Änderungen begrenzt</li> <li>• Viel Dokumentation als Absicherung</li> </ul> |

→ Spezifikationsorientiert umsetzen oder agil mit teilweise festgelegtem Leistungsumfang.

Tabelle 11: Project-Constraint „Kundenerfahrung“

| Kunde unerfahren   | Beispielgründe   |
|--|--|
| Erfahrungsmangel muss kompensiert werden, damit Zusammenarbeit funktioniert. | <ul style="list-style-type: none"> <li>• Wenig Projekterfahrung</li> <li>• Keine Erfahrung mit Agilen Projekten</li> </ul> |

→ Schulung des Kunden

→ Product Owner durch (erfahrenen) Auftragnehmer stellen

Tabelle 12: Project-Constraint „Kommunikation“

| Asynchrone Kommunikation   | Beispielgründe   |
|--|--|
| Findet die projektinterne Kommunikation überwiegend asynchron statt, leidet die Zusammenarbeit der Teammitglieder. | <ul style="list-style-type: none"> <li>• Team ist weit verstreut</li> <li>• Projektmitglieder arbeiten parallel an mehreren Projekten</li> </ul> |

→ Regelmäßige Meetings zur Steuerung des Informationsflusses.<sup>196</sup>

<sup>194</sup> vgl. Maximini:2018, 99.

<sup>195</sup> vgl. 2.5.1 Agile Softwareentwicklung, 13f.

<sup>196</sup> vgl. 4.2.1 Kommunikation, 64f.

## 5.1.2 Unternehmensbezogene Project-Constraints

Tabelle 13: Project-Constraint „PM-Erfahrung“

| <b>Mangelnde Erfahrung mit Projektmanagement</b>                          | <b>Beispielgründe</b>  |
|---|--|
| Erfahrungsmangel muss kompensiert werden, damit Risiken minimiert werden. | <ul style="list-style-type: none"> <li>• Bisher kaum/keine Projekte</li> <li>• Projektmanagement vernachlässigt</li> </ul> |

- *Recherchen & Schulungen, bevor Projektvorgehen gewählt wird*
- *Projektmanagement-Coach mit ins Projekt nehmen*

Tabelle 14: Project-Constraint „Erfahrung mit Vorgehensmodellen“

| <b>Mangelnde Erfahrung mit iterativem Vorgehen / ausschließlich Wasserfall</b>           | <b>Beispielgründe</b>  |
|--|--|
| Erfahrungsmangel muss kompensiert werden, damit iterative Modelle gewählt werden können. | <ul style="list-style-type: none"> <li>• Projekte funktionierten mit Wasserfall</li> <li>• Bisher nur geringe Komplexität</li> </ul> |

- *Recherchen & Schulungen, bevor Projektvorgehen gewählt wird*
- *Coach mit ins Projekt nehmen*

Tabelle 15: Project-Constraint „Projektarbeit“

| <b>Eingefahrene Projektarbeitsweisen</b>  | <b>Beispielgründe</b>   |
|---|---|
| Eingefahrene Verhaltensweisen müssen regelmäßig kritisch reflektiert und ggf. abgebaut werden. <sup>197</sup> | <ul style="list-style-type: none"> <li>• Etablierte Prozesse</li> <li>• Erfolgreiche Verhaltensweisen</li> <li>• Betriebsblindheit</li> </ul> |

- *Auswerten der Projekte & Projektprozesse (z. B.: Wie werden Meetings protokolliert?)*

Tabelle 16: Project-Constraint „Prozesskonflikte“

| <b>Konflikte mit Unternehmensprozessen</b>  | <b>Beispielgründe</b>  |
|---|--|
| Bei Neueinführung von Vorgehensmodellen muss sichergestellt werden, dass diese mit Unternehmensprozessen nicht in Konflikt <sup>198</sup> stehen. | <ul style="list-style-type: none"> <li>• Projekte werden der Linienorganisation unterstellt</li> </ul> |

- *Analyse von Unternehmensprozessen & Vorgehensmodellen*
- *Anpassung der Vorgehensmodelle bzgl. der Konfliktstellen*

<sup>197</sup> vgl. Engstler:2016, 89.

<sup>198</sup> vgl. Maximini:2018, 120.

Tabelle 17: Project-Constraint „Mangel an Fähigkeiten“

| Wenige Spezialisten  | Beispielgründe   |
|--|--|
| Im Vorfeld muss klar sein, welche Kenntnisse benötigt werden und welche Spezialisten dafür infrage kommen. | <ul style="list-style-type: none"> <li>• Bestimmte Kenntnisse werden sehr selten gebraucht</li> <li>• Spezialisten sind teuer</li> </ul> |

- Kleines Spezialistenteam Vollzeit für das Projekt abstellen
- Agiles Vorgehen

Tabelle 18: Project-Constraint „Führungsetage“

| Kein Rückhalt für Neueinführungen   | Beispielgründe  |
|---|---|
| Rückhalt der Führungsetage ist bei Neueinführung eines Vorgehensmodells essentiell. Auch die eigenen Mitarbeiter müssen eine Neueinführung unterstützen, damit sie erfolgreich sein kann. | <ul style="list-style-type: none"> <li>• Wenig Fehlertoleranz</li> <li>• Kaum Offenheit für Neues</li> <li>• Betriebsblindheit</li> </ul> |

- Führungsetage/Mitarbeiter müssen von den Vorteilen einer Neueinführung überzeugt werden

### 5.1.3 Kosten als Project-Constraints

Tabelle 19: Project-Constraint „Kosten der Projektorganisation“

| Einführungskosten  | Beispielgründe  |
|--|---|
| Sind in Zukunft viele Projekte zu erwarten, die mit dem neuen Modell umsetzbar sind, lohnt sich eine Neueinführung möglicherweise. | <ul style="list-style-type: none"> <li>• Neues Vorgehensmodell</li> <li>• Neueinstellungen</li> </ul> |

- Kosten müssen mit zu erwartendem Nutzen verglichen werden

Tabelle 20: Project-Constraint „Schulungskosten“

| Coaching durch Experten   | Beispielgründe   |
|---|--|
| Können die erlernten Fähigkeiten und das Wissen aus dem Coaching zukünftig Einsatz finden, lohnt sich ein entsprechendes Coaching möglicherweise. | <ul style="list-style-type: none"> <li>• Projektmanagement</li> <li>• Einführung Vorgehensmodell</li> <li>• Agiles Denken</li> </ul> |

- Kosten müssen mit zu erwartendem Nutzen verglichen werden

#### 5.1.4 Projekt-Individuen als Project-Constraints

Tabelle 21: Project-Constraint „Rollenverständnis“

| Scrum Master ist kein Projektmanager  | Beispielgründe   |
|---|--|
| Versteht sich der Scrum Master als Projektmanager <sup>199</sup> , so wird die Selbstorganisation im agilen Denken ad absurdum geführt. | <ul style="list-style-type: none"> <li>• Klassische Projektleiter werden als Scrum Master eingesetzt</li> <li>• Keine Erfahrung mit Agilen Ansätzen</li> </ul> |

→ *Einführung von agilen Methoden heißt nicht verstehen. Schulung & Projektbetreuung durch Coach notwendig.*

Tabelle 22: Project-Constraint „Kundenmitwirkungspflicht“

| Kunde sieht sich nicht in der Pflicht   | Beispielgründe   |
|---|--|
| Entzieht sich der Kunde seiner Verantwortung, kann das Projekt nicht ohne Weiteres agil umgesetzt werden. | <ul style="list-style-type: none"> <li>• Kunde hat keine Projekterfahrung</li> <li>• Fehlendes Verständnis für seine Mitverantwortung</li> </ul> |

→ *Product Owner vom Auftragnehmer stellen*

#### 5.1.5 Umgang mit Project-Constraints

Projekte sind schon ihrer Definition nach komplexe Unternehmungen. Daher ist es schon im Vorfeld der konkreten Projektumsetzung sinnvoll zu wissen, welche infrage kommenden Project-Constraints existieren oder existieren könnten. Unter deren Einbeziehung kann dann analysiert werden, was im Kontext des geplanten Projektvorhabens machbar ist und auf welche Weise das Projekt organisiert werden soll, also welche Vorgehensmodelle in Betracht gezogen werden können, welches Preismodell infrage kommt usw. Das Ganze ist nicht zu verwechseln mit einer Machbarkeitsstudie<sup>200</sup>, die bereits vor der Genehmigung eines Projektes durchgeführt wird, um die Chancen und Risiken zu ergründen. Die Frage nach möglichen Constraints kann sich zwar mit den Untersuchungen einer solchen Studie überschneiden, beschränkt sich aber nicht darauf, weil sie im Verlauf des Projekts immer wieder neu gestellt werden muss. So kann beispielsweise das Verhalten von Projekt-Individuen erst im fortgeschrittenen Projektverlauf analysiert und somit möglicherweise ein Problem erkannt werden, etwa, weil der Kunde seiner Verantwortung

<sup>199</sup> vgl. *yeebase media GmbH:2013*, Abschn. „Typische ScrumButs“.

<sup>200</sup> vgl. *Jung:2014*.

nicht ausreichend nachkommt und dem Team für Rückfragen nicht zur Verfügung steht.

## 5.2 Agile Projektarbeit in einer klassischen Unternehmensstruktur

Für Unternehmen, die nur wenig Projektarbeit betreiben, besteht die Möglichkeit, Projekte in einer Art Kapsel in die Unternehmensstruktur einzubetten.<sup>201</sup> Von außen betrachtet ist eine solche Kapsel nur einer von vielen Unternehmensbereichen. Innerhalb der Kapsel jedoch herrschen an den Projektkontext angepasste Regeln, die jedoch die umliegende Unternehmensstruktur nicht beeinträchtigen sollten.

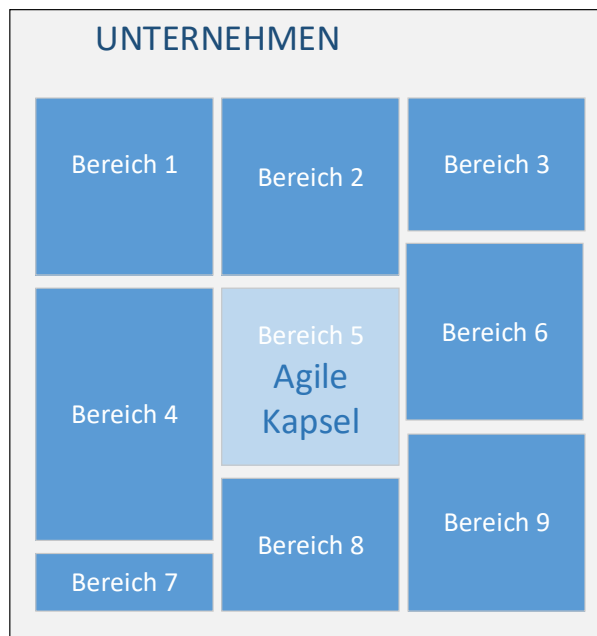


Abbildung 19: Agile Kapsel (Quelle: Eigene Darstellung).

Theoretisch kann eine derartige Kapselstruktur innerhalb eines Unternehmens, das nicht auf Projekte ausgerichtet ist, dazu genutzt werden, die Vorteile von Projekten zu kombinieren mit den Vorteilen der nichtprojektbezogenen Arbeit. Sowohl spezifikationsorientierte, als auch agile Vorgehensweisen wären hier mithilfe von verschiedenen Kapseln denkbar.

Auf eine agile Kapsel beschränkt wäre eine Organisation, die kein reines IT-Unternehmen ist, jedoch immer wieder IT-Projekte durchführen muss (bspw. Hersteller von Imagefilmen), dazu in der Lage, notwendige IT-Projekte

<sup>201</sup> vgl. Abb. 19.

unternehmensintern abzuwickeln parallel zum Tagesgeschäft, z. B. in Form von einem *Scrum Software Studio*<sup>202</sup>. Jedes IT-Projekt, das dann mit Scrum umgesetzt werden soll, würde innerhalb der agilen Kapsel stattfinden.

Damit lassen sich die Projektplanung und -organisation vereinfachen, indem Vorgehensweisen, Prozesse sowie Ressourcen neuen Projekten bereitgestellt und Erfahrungen aus abgeschlossenen Projekten gebündelt werden und einzelne Projekte nicht innerhalb der normalen Unternehmensstruktur um Ressourcen kämpfen müssen und Projekterfahrungen nicht verloren gehen.

Werden nur selten eher kleinere IT-Projekte durchgeführt, ist die agile Kapsel eine Methode, um die Vorteile agiler Ansätze auszureizen, ohne die eher klassisch geprägte Organisation zu stören.

### 5.3 Skalierung von agilen Modellen für Großprojekte

Agile Vorgehensmodelle bauen auf Projektvorhaben auf, die mit einem Team von kleiner Größe umgesetzt werden können. So empfiehlt etwa der Scrum Guide ein Entwicklerteam, welches zwischen drei und neun Teammitglieder umfasst.<sup>203</sup> Für größere Projekte mit mehreren Teams, die nach agilem Vorbild laufen sollen, gibt es die Möglichkeit, Frameworks für eine Skalierung agiler Modelle anzuwenden, wobei man nach Maximini unter Skalierung „den Einsatz mehrerer paralleler Teams, statt eines Einzelnen“<sup>204</sup> versteht. Skaliert wird nach Maximini jedoch nicht das Modell, sondern das Projektumfeld, wobei mehrere Skalierungsansätze zur Auswahl stehen.<sup>205</sup> Mithilfe der Skalierung soll es möglich werden, mit mehreren Teams parallel und agil an einem Projekt zu arbeiten und diese durch übergeordnete Steuerung zu koordinieren. Nachfolgend werden die Skalierungsansätze *SAFe*, *Scrum of Scrums*, *LeSS* sowie *Nexus* vorgestellt und einer kritischen Betrachtung unterzogen.

#### 5.3.1.1 *SAFe*

Das Scaled Agile Framework, kurz *SAFe*, liefert ein relativ komplexes Framework, das laut Hersteller sowohl bei Projekten mit 50 – 125, als auch bei großen Projekten mit mehreren tausend Projektmitgliedern

---

<sup>202</sup> vgl. Maximini:2018, 4ff.

<sup>203</sup> siehe Schwaber; Sutherland:2016, 6.

<sup>204</sup> Maximini:2018, 121.

<sup>205</sup> vgl. Maximini:2018, 121.

funktionieren soll.<sup>206</sup> Das Framework ist im engeren Sinne nicht agil, sondern soll Lean Product Development, Agile Development und Systems Thinking<sup>207</sup> miteinander verbinden.<sup>208</sup>

SAFe umfasst diverse Rollen, Verantwortlichkeiten, Artefakte und Aktivitäten, die auf der Website des Herstellers beschrieben werden.<sup>209</sup> Eine Einführung kann unterschiedlichen Aufwand erzeugen, je nach gewählter Konfiguration des Frameworks, wobei die folgenden vier Framework-Varianten existieren:

- Essential SAFe
- Large Solution SAFe
- Portfolio SAFe
- Full SAFe

Aufgrund der Komplexität dieses Frameworks ist ein höherer Aufwand gefordert als beispielsweise bei der einfachen Skalierung, wie sie *Scrum of Scrums* bietet. Es kann daher als schwergewichtig bezeichnet werden und ist besonders für Großprojekte geeignet.

Für Unternehmen, die bereits einen umfangreichen Erfahrungsschatz besitzen zum Thema agile Projekte, kann die Verwendung eines solchen Frameworks sinnvoll sein, wenn die Skalierungsmöglichkeiten ausgereizt sind und größere Maßstäbe benötigt werden.<sup>210</sup>

---

<sup>206</sup> siehe *Scaled Agile Inc.:2018b*.

<sup>207</sup> Systems Thinking oder Systemdenken kann als das wissenschaftliche Problem verstanden werden, in Systemen zu denken (vgl. *Arnold; Wade:2015*, 670.).

<sup>208</sup> vgl. *Scaled Agile Inc.:2018b*.

<sup>209</sup> vgl. *Scaled Agile Inc.:2018a*.

<sup>210</sup> vgl. *Orientation in Objects GmbH:2018*, Abschn. „Voraussetzung 3: Erfahrung mit agiler Softwareentwicklung“.



### 5.3.1.2 Scrum of Scrums

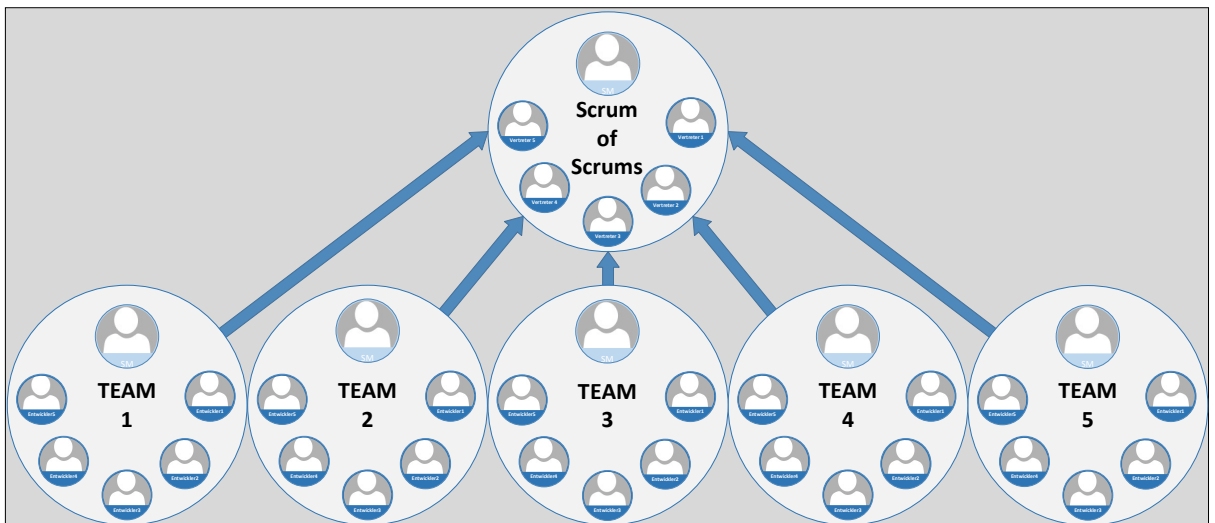


Abbildung 20: Scrum of Scrums (Quelle: Eigene Darstellung in Anlehnung an Lindner:2017).

Für große Scrum-Projekte, die mehr als drei Teams benötigen, gibt es eine relativ simple Skalierungsmöglichkeit - das sogenannte Scrum of Scrums. Beim Scrum of Scrums halten alle Teams ihre Scrum Meetings gemäß dem Scrum Guide ab.<sup>211</sup> Ergänzend dazu findet im Nachgang ein weiteres Meeting statt, bei dem jeweils ein Teamvertreter aus den einzelnen Scrum Teams teilnimmt. Dort werden auf dieser höheren Ebene alle relevanten Ergebnisse aus den Einzelteam-Meetings zusammengetragen. Eine weitere Skalierung zum Scrum of Scrum of Scrums etc. ist analog möglich, birgt jedoch Risiken, da die Beherrschbarkeit durch den zusätzlichen Koordinationsaufwand, der mit jeder weiteren Skalierungsstufe entsteht, sinkt.<sup>212</sup> Maximini empfiehlt für den Scrum of Scrums eine Teamanzahl zwischen drei und fünf,<sup>213</sup> wobei es generell sinnvoll ist, beim Aufbau der „Scrum of...“-Hierarchie den Erfahrungsstand und die vorhandene Disziplin der einzelnen Teams abzuschätzen und entsprechend in die Hierarchieplanung einzubeziehen.

### 5.3.1.3 LeSS

Large Scale Scrum (LeSS) liefert zwei verschiedene Frameworks: LeSS in der einfachen Variante, was für bis zu acht Scrum Teams mit jeweils acht Teammitgliedern ausgelegt ist, sowie LeSS Huge für eine Scrum-Skalierung,

<sup>211</sup> vgl. 3.4.2.1 Scrum, 46ff.

<sup>212</sup> vgl. „Lindner:2017“.

<sup>213</sup> siehe Maximini:2018, 124.

die mehrere tausend Projektmitglieder erlaubt. Ähnlich wie Scrum of Scrums basiert das Framework im Kern auf Scrum.

Im Folgenden werden LeSS und LeSS Huge kurz vorgestellt.

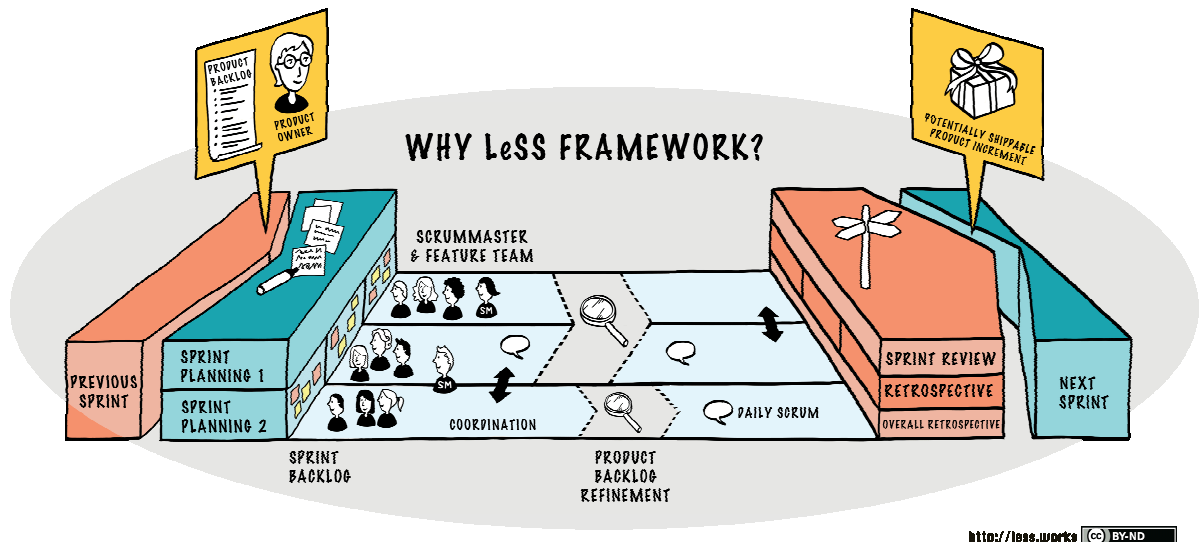


Abbildung 21: LeSS Framework (Quelle: The LeSS Company B.V.:2018a).

LeSS weist nur einige Unterschiede zum originalen Scrum auf: Um die Koordination der höchstens acht Teams zu gewährleisten, werden die Scrum Meetings durch einige veränderte Abläufe angepasst:

Für die Mitglieder aller Teams gibt es nun das *Sprint Planning Part 1* sowie das *Sprint Review*. In Ersterem werden die Product Backlog Einträge auf die einzelnen Teams verteilt. Letzteres spricht für sich selbst. Zusätzlich wird das *Product Backlog Refinement* für alle Teams eingeführt, das zwar schon in der alltäglichen Scrum Praxis (auf Einzelteamebene) existiert, im Scrum Guide aber nicht explizit vorgeschrieben ist. Hier werden die Product Backlog Einträge verfeinert mit dem Ziel, das Backlog auf den kommenden Sprint vorzubereiten, indem z. B. große Stories (Epics) in kleinere User Stories heruntergebrochen werden und den einzelnen Stories Akzeptanzkriterien und Aufwandschätzungen beigefügt werden. Das Product Backlog Refinement soll nach LeSS auf Einzelteamebene („*single-team PBR*“), sowie mit allen Teams („*multi-team PBR*“)<sup>214</sup> stattfinden.

<sup>214</sup> siehe *The LeSS Company B.V.:2018a*.

Schließlich führt LeSS die *Overall Retrospective* ein, ein Meeting mit dem Ziel, die teamübergreifende Projektorganisation zu verbessern, indem Probleme gesammelt und gemeinsam analysiert werden.<sup>215</sup>

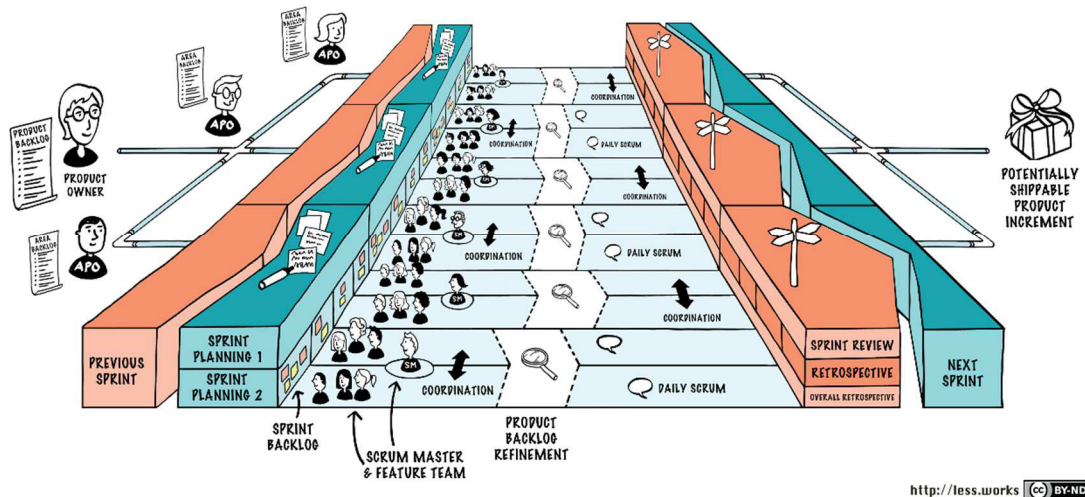


Abbildung 22: LeSS Huge (Quelle: The LeSS Company B.V.:2018b).

Das LeSS Huge Framework baut auf dem von LeSS auf. Der wesentliche Unterschied ergibt sich aus der weiteren Skalierung: Dabei werden die Teams verschiedenen *Requirement Areas*, also Anforderungsbereichen, zugeteilt. Ein solcher Bereich umfasst mehrere Features, wobei jedes Feature von einem Feature Team bearbeitet wird. D.h. ein Anforderungsbereich beschäftigt mehrere Teams. Diesem wird ein *Area Product Owner* zugeordnet, der sich um die Priorisierung und Pflege des *Area Product Backlogs* kümmert – ein detaillierter Ausschnitt des gesamten Product Backlogs. Die Area Product Owner übernehmen in ihrem Anforderungsbereich analog die Aufgaben, die sonst ein herkömmlicher Product Owner hat. Übergeordnet existiert zudem auch ein Product Owner, der gemeinsam mit den Area Product Ownern das *Product Owner Team* bildet.<sup>216</sup>

Sowohl LeSS als auch LeSS Huge liefern ein verhältnismäßig simples Framework, mit dem sich Scrum relativ einfach skalieren lässt. Dass die Abweichungen zum klassischen Scrum unter Verwendung dieser Frameworks nur gering ausfallen ist vorteilhaft für diejenigen Unternehmen, die bereits in kleinen

<sup>215</sup> vgl. The LeSS Company B.V.:2018c.

<sup>216</sup> vgl. The LeSS Company B.V.:2018d.

Projekten mit Scrum gearbeitet haben und für größere Projekte Scrum skalieren möchten.

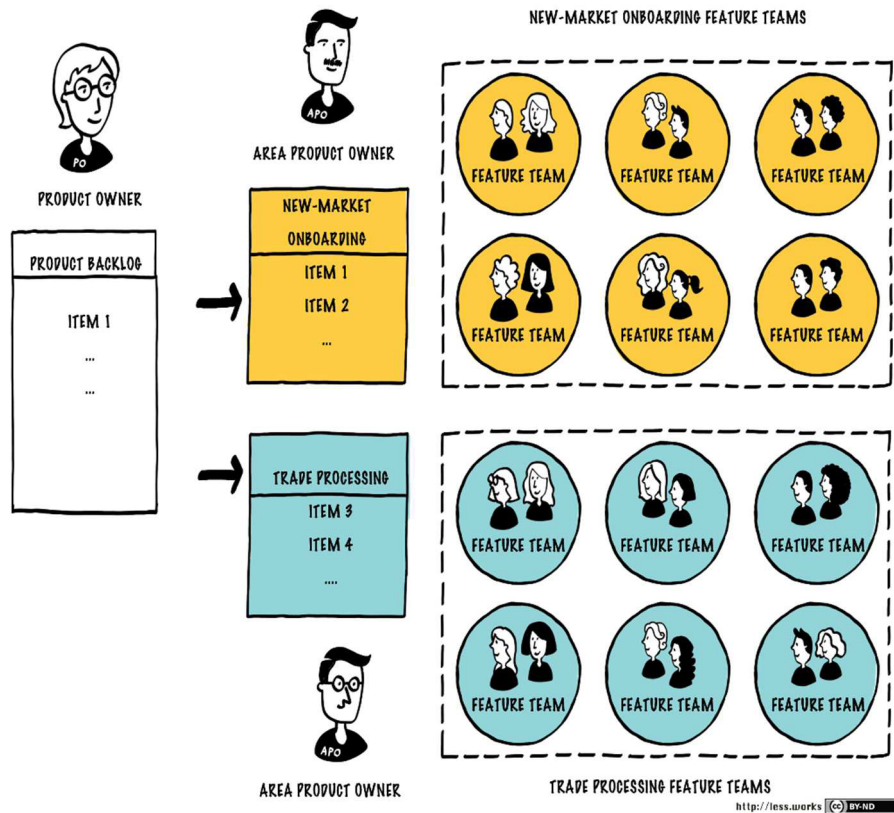


Abbildung 23: Requirement Areas (Quelle: The LeSS Company B.V.:2018d).

### 5.3.1.4 Nexus

## NEXUS™ FRAMEWORK

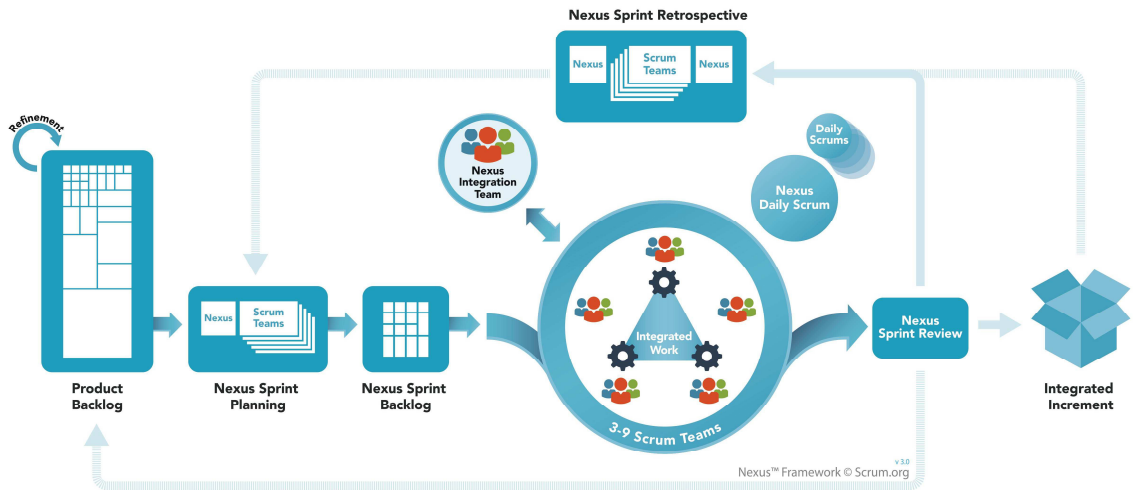


Abbildung 24: Nexus Framework (Quelle: Scrum.org:2018).

Nexus liefert ein weiteres Framework zur Skalierung von Scrum<sup>217</sup>, das von Ken Schwaber, einem der beiden Scrum-Entwickler, und Scrum.org gemeinsam entwickelt wurde. Dabei erweitert der Nexus Guide das originale Scrum-Rahmenwerk derart, dass nun drei bis neun Teams aufgesetzt werden können.

Für die Koordination der Scrum Teams, die Überwachung von Scrum und Nexus (bzw. Sicherstellung, dass die Regeln von Scrum & Nexus eingehalten werden) und die Beseitigung von Integrationsproblemen wurde das *Nexus Integration Team* geschaffen. Seine Hauptverantwortung jedoch besteht darin sicherzustellen, dass bei jedem Sprintende mindestens ein Inkrement vollständig integriert ist. Das Nexus Integration Team setzt sich zusammen aus jeweils einem Scrum Master und einem Product Owner sowie weiteren Teammitgliedern.

Alle Scrum Teams arbeiten an demselben Product Backlog, welcher im Product Backlog Refinement regelmäßig verfeinert wird. Ziel der Verfeinerung ist es, Abhängigkeiten unter den einzelnen Product Backlog Einträgen aufzudecken

<sup>217</sup> vgl. Abb. 24.

und möglichst zu minimieren, sodass es nicht zu Problemen kommt bei der teamübergreifenden Entwicklung von Inkrementen.

Jeder Sprint startet mit dem *Nexus Sprint Planning*, wobei ausgewählte Vertreter aller Teams anwesend sind. Im Anschluss daran erstellt jedes Scrum Team seinen *Nexus Sprint Backlog* im klassischen Sprint Planning.

Der *Nexus Daily Scrum* ist, analog zum Daily Scrum, ein tägliches kurzes Meeting gewählter Vertreter aller Scrum Teams, um den aktuellen Projektstand sowie vorhandene Integrationsprobleme und neu entdeckte Abhängigkeiten zu besprechen. Im Daily Scrum werden die Ergebnisse des Nexus Daily Scrum an die Scrum Teams kommuniziert, sodass Abhängigkeiten beseitigt und Integrationsprobleme gelöst werden können. Im *Nexus Sprint Review* wird das im Sprint erstellte Inkrement vorgestellt sowie das Product Backlog überarbeitet.<sup>218</sup>

Bevor ein neuer Sprint beginnt, wird der aktuelle Sprint in der *Nexus Sprint Retrospektive* unter Anwesenheit von Vertretern aus den Teams sowie in darauf folgenden Sprint Retrospektiven in den einzelnen Teams auf Probleme hin untersucht und Gegenmaßnahmen entwickelt, analog der Vorgehensweise bei der Sprint Retrospektive aus Scrum.<sup>219</sup>

Da Nexus das Regelwerk von Scrum beibehält und das Framework lediglich erweitert wird, sodass mithilfe des Nexus Integration Teams eine Instanz geschaffen wurde, die sich um die Koordination der Teams sowie die Integration von Inkrementen kümmert, sollte eine Skalierung von Scrum problemlos möglich sein, sofern bereits die nötigen Kompetenzen im Umgang mit Scrum vorhanden sind.

### 5.3.2 Skalierung – Vor- und Nachteile

Im Allgemeinen ermöglicht es die Skalierung, dass agile Modelle, die eher auf kleine Projektsituationen zugeschnitten sind, an größere Projektkontexte adaptiert werden können, sodass darin Agilität zumindest teilweise möglich wird. Dabei hängt die agile Ausprägung vom gewählten Framework ab: Während umfangreiche Frameworks wie SAFe durch ihre hohe Komplexität weniger

---

<sup>218</sup> vgl. Schwaber; Scrum.org:2018, 9.

<sup>219</sup> vgl. 3.4.2.1 Scrum, 47.

Spielraum für die Umsetzung der agilen Prinzipien lassen, werden diese hingegen stark unterstützt von Frameworks, die auf einem einfacheren agilen Modell basieren und es erweitern, z. B. Nexus.

Unter Verwendung eines Skalierungsframeworks kann agile Projektarbeit auf unterschiedliche Projektsituationen angepasst werden. Je offener das Framework ist, desto einfacher ist es möglich, Vorgehensweisen und Praktiken aus anderen Vorgehensmodellen zu integrieren, um es auf die spezifische Projektsituation zuzuschneiden.

Zudem kann damit der Vorteil genutzt werden, agiles Vorgehen dort einzusetzen, wo es im speziellen Projektkontext sinnvoll ist, während an anderer Stelle z. B. Methoden aus dem nichtagilen Projektmanagement angewandt werden können.

Nachteilig ist jedoch, dass die Skalierungsframeworks den agilen Modellen ihre Einfachheit nehmen, tendenziell je größer skaliert wird. So kann z. B. in einem Unternehmen, das wenig Erfahrung in agiler Projektarbeit aufweist, eine Skalierung zu Chaos führen, wenn zu viele Teams koordiniert werden müssen.<sup>220</sup> Wissen und Erfahrung im Umgang mit dem jeweiligen Framework sollte zudem nachgehalten werden, was zu zusätzlichem Aufwand führt (z. B. Schulung der Mitarbeiter, Bündelung von Best Practices etc.).

#### 5.4 Entscheidungsfindung für Unternehmen

Die Frage, ob ein bestimmtes Projekt besser agil oder spezifikationsorientiert umgesetzt werden sollte, muss im Einzelfall von den Verantwortlichen getroffen werden. Eine standardisierte Entscheidungshilfe, nach welcher Projekte im wissenschaftlichen Sinne kategorisiert werden und demzufolge das richtige Vorgehen vorgeschlagen werden kann, scheint derzeit nicht zu existieren. Das Fraunhofer Institut zumindest versucht mittels einer Umfrage<sup>221</sup> herauszufinden, welche Art von Projekten deutsche Forschungs- und Entwicklungsunternehmen agil abwickeln. Das legt den Schluss nahe,

---

<sup>220</sup> vgl. *Maximini:2018*, 121.

<sup>221</sup> vgl. *Schimpf:2017*.

dass die Fragen nach dem Vorgehen, ob agil oder nicht, und dem konkreten Vorgehensmodell nicht einfach zu beantworten sind.

Aus diesem Grund stellt die nachfolgende Tabelle allgemeine Projekteigenschaften zusammen und listet beispielhaft Fragen auf, die zur Bewertung herangezogen werden können. Im Rahmen dieser Arbeit soll diese Kategorisierung ein Beispiel dafür liefern, wie Unternehmen die Entscheidungsfindung bzgl. der geeigneten Projekt-Vorgehensweise angehen könnten.

Zur Verdeutlichung werden zwei erdachte Szenarien herangezogen, um mithilfe von *Tabelle 23* Empfehlungen für das Projektvorgehen zu erlangen.

Dazu werden die folgenden Schritte durchlaufen:

1. Projekteigenschaften aus dem Szenario entnehmen
2. Bewertung der Projekteigenschaften vornehmen
3. Empfehlung für das Projektvorgehen geben

Je nachdem, wie detailliert ein beliebiges Szenario beschrieben wird, können dabei mehr oder weniger Eigenschaften herausgefiltert und entsprechend bewertet werden. Je mehr Eigenschaften erkennbar sind, desto realistischer lässt sich abwägen, welche Vorgehensweise am sinnvollsten ist.

*Tabelle 23: Projekteigenschaften zur Bewertung eines möglichen Vorgehens*

| PERSPEKTIVE           | EIGENSCHAFT   | FRAGEN ZUR BEWERTUNG  |
|-----------------------|---|---|
| <i>Projektaufgabe</i> | <b>Komplexität</b> <ul style="list-style-type: none"> <li>▪ Neuartigkeit</li> <li>▪ Größe</li> </ul>  | Wie neuartig ist das Projektvorhaben?<br>Wie groß ist der Projektumfang?<br>Gibt es vergleichbare Projekte bei uns? |
|                       | <b>Risiken</b> <ul style="list-style-type: none"> <li>▪ Anforderungen</li> <li>▪ Moving Targets</li> </ul>  | Sind alle Anforderungen bekannt?<br>Sind viele Moving Targets zu erwarten?  |
|                       | <b>Vertragsregelung</b> <ul style="list-style-type: none"> <li>▪ Liefertermin</li> <li>▪ Liefergegenstand</li> <li>▪ Preismodell</li> </ul>                           | Gibt es einen festen Liefertermin?<br>Was sind die Liefergegenstände?<br>Welches Preismodell soll verwendet werden? |
|                       | <b>Budget</b> <ul style="list-style-type: none"> <li>▪ Schulungen</li> <li>▪ Berater</li> <li>▪ Neueinstellung</li> <li>▪ Outsourcing von Dienstleistungen</li> </ul> | Wie viel Budget ist vorhanden für Schulungszwecke, Berater, Neueinstellungen und Outsourcing?                       |



|                           |  |  |
|---------------------------|--|--|
| <i>Projektumfeld</i>      | <b>Auftraggeber</b> <ul style="list-style-type: none"> <li>▪ Umfeld</li> <li>▪ Projektvorgehen</li> <li>▪ Kommunikation</li> <li>▪ Projektrolle</li> </ul> | <p>In welchem Umfeld ist der AG angesiedelt?<br/> Wie wickelt AG eigene Projekte ab?<br/> Welche Vorgehensmodelle sind erlaubt?<br/> Wie wird Kommunikation mit AG abgewickelt?<br/> Welche Rolle nimmt der AG im Projekt ein?</p>           |
|                           | <b>Auftragnehmer</b> <ul style="list-style-type: none"> <li>▪ Umfeld</li> <li>▪ Projektvorgehen</li> <li>▪ Kommunikation</li> </ul>                        | <p>In welchem Umfeld ist der AN angesiedelt?<br/> Wie wickelt AN eigene Projekte ab?<br/> Welche Vorgehensmodelle sind erlaubt?<br/> Wie wird Kommunikation mit AG abgewickelt?</p>  |
|                           | <b>Stakeholder</b>   | <p>Wer sind die Endanwender und mögliche weitere Stakeholder?</p>  |
| <i>Unternehmenskultur</i> | <b>Projektkultur</b>   | <p>Welche etablierten Projektarbeitsweisen gibt es?<br/> Wie ist die Projektorganisation in die Unternehmensorganisation eingebettet?</p>  |
|                           | <b>Projekt-Wissensmanagement</b> <ul style="list-style-type: none"> <li>▪ Projekterfahrung</li> <li>▪ Projektwissen</li> </ul>                             | <p>Wo und wie wird Projekterfahrung gesammelt und aufbereitet (z. B. Lessons Learned, Best Practices)?<br/> Kann Projektwissen von anderen Teams zukünftig genutzt werden?</p>   |
|                           | <b>Prozessmanagement</b>   | <p>Treten Konflikte zw. Unternehmensprozessen &amp; Projekten auf?<br/> Welche Auswirkungen von Prozessänderungen auf Projekte sind zu erwarten?</p>   |
|                           | <b>Unternehmenswerte</b> <ul style="list-style-type: none"> <li>▪ Fehlertoleranz</li> <li>▪ Offenheit</li> <li>▪ Führungsetage</li> </ul>                  | <p>Gibt es eine ausgeprägte Fehlertoleranz im Unternehmen?<br/> Wie offen ist das Unternehmen für Neuerungen?<br/> Welchen Einfluss hat die Führungsetage auf Projekte?<br/> Wie eigenständig können Projektverantwortliche entscheiden?</p> |
|                           | <b>Unternehmensmerkmale</b> <ul style="list-style-type: none"> <li>▪ Betriebsblindheit</li> <li>▪ Flexibilität</li> <li>▪ Agilität</li> </ul>              | <p>Wie geht das Unternehmen mit Betriebsblindheit um?<br/> Wie flexibel kann das Unternehmen auf Veränderungen reagieren?<br/> Wie stark ist Agilität ausgeprägt?</p>  |

|                                |   |  |
|--------------------------------|---|--|
| <i>Projekt-<br/>Ressourcen</i> | <b>Team</b> <ul style="list-style-type: none"> <li>▪ Teamfähigkeiten</li> <li>▪ Verfügbarkeit</li> </ul>          | Mit welchen Fähigkeiten sind potenzielle Teammitglieder ausgestattet (z. B. Selbstorganisation, Konfliktlösung)?<br>Wie sind sie verfügbar (zeitlich, räumlich)? |
|                                | <b>Projektmanagement</b> <ul style="list-style-type: none"> <li>▪ Fähigkeiten</li> <li>▪ Verfügbarkeit</li> </ul> | Mit welchen PM-Fähigkeiten sind potenzielle Projektmanager ausgestattet?<br>Wie sind sie verfügbar (zeitlich, räumlich)?   |
|                                | <b>Räumlichkeiten &amp; Techn. Ausstattung</b>  | Wo arbeiten die Teams?<br>Welche techn. Ressourcen sind verfügbar?   |

#### 5.4.1 Szenario 1: Neuentwicklung

*Das junge mittelständische Unternehmen Fit&Style wurde 2010 in Deutschland gegründet und hat sich spezialisiert auf die Produktion und den Verkauf von Sportklamotten über den Onlinehandel. Seit einigen Jahren steigt das Interesse rund um Fitnessprodukte rasant, was sich nicht zuletzt an der harten Konkurrenz im Onlinegeschäft bemerkbar macht. Auch die technischen Entwicklungen sind Fit&Style nicht entgangen und so stellt das Unternehmen fest, dass der Onlinehandel über mobile Endgeräte immer wichtiger wird. Aus diesem Grund möchte Fit&Style ein Projekt in Auftrag geben: Eine App für iOS und Android soll entwickelt werden, damit die Kunden auch über Smartphones und Tablets shoppen können. Neben dem Zugang zum bereits bestehenden Online-Shop soll die App weitere Funktionen beinhalten. Gewünscht sind Personalisierungsfunktionen, weil man sich damit eine höhere Kundenbindung verspricht. Welche Möglichkeiten es gibt, weiß Fit&Style jedoch nicht, das muss erst noch herausgefunden werden. Die erste Version der App muss auf jeden Fall rechtzeitig zum Start der Fitnessmesse Fibo bis zum 03.04.2019 veröffentlicht worden sein, damit sie dort präsentiert werden kann. Fit&Style ist, als aufgeschlossenes und relativ junges Unternehmen, mit Projektarbeit vertraut und hat auch schon kleinere Projekte selbst abgewickelt, allerdings war dabei Projektmanagement kaum vorhanden.*

#### 5.4.1.1 Bewertung der Projekteigenschaften

##### Komplexität:

- Neuentwicklung sorgt für hohe Komplexität
- Anforderungen nur teilweise bekannt, daher viele Moving Targets zu erwarten

##### Vertragsregelung:

- Fixer Liefertermin ermöglicht Timebox-Methode

##### Projektmanagement:

- Da PM-Fähigkeit kaum vorhanden, kann Selbstorganisation des Teams hilfreich sein

##### Unternehmensmerkmale:

- Junges und offenes Unternehmen, daher wenig eingefahrene Verhaltensweisen und wenig Betriebsblindheit
- Führungsetage besteht überwiegend aus jungen Managern mit modernem Führungsstil
- Projekte wurden bereits durchgeführt, sodass zumindest eine gewisse Projekterfahrung vorhanden ist

#### 5.4.1.2 Empfehlung auf Basis der Bewertung

Die im Rahmen dieses beschriebenen Szenarios erkennbaren Eigenschaften sprechen für eine agile Umsetzung des Projektvorhabens. Mithilfe einer agilen Vorgehensweise sollte sich die Komplexität des Projektes beherrschbar machen lassen, indem zuerst die groben Anforderungen an die Software herausgearbeitet und kontinuierlich verfeinert werden. Kurze Planungs- und Umsetzungshorizonte, z. B. durch Sprints bei Scrum, können dafür sorgen, schnell brauchbare Ergebnisse zu erzielen und das Projekt sinnvoll zu takten, sodass zum gewünschten Termin eine funktionierende Version der App vorliegt. Während im spezifikationsorientierten Rahmen umfassende PM-Kenntnisse und PM-Spezialisten notwendig sein könnten, kann ein agiles Projekt auch mit deutlich weniger davon auskommen, sofern die Selbstorganisation des Entwicklungsteams funktioniert. Auch die erkennbaren Unternehmensmerkmale sind vorteilhaft für eine agile Umsetzung.

## 5.4.2 Szenario 2: Migration

*Das deutsche Versicherungsunternehmen GesundPlus wurde 1995 gegründet und zählt heute zu den Großunternehmen. Das mittlerweile veraltete Kundenverwaltungssystem soll nun ersetzt werden und sämtliche Daten müssen in das neue System migriert werden. GesundPlus möchte ein IT-Unternehmen beauftragen, das den Systemwechsel im Rahmen eines Migrationsprojekts vollzieht. Dabei soll das Altsystem von Standardsoftware abgelöst werden. Die Anforderungen an das neue System sind dem Auftraggeber bereits bei Projektausschreibung relativ klar: Die bekannten Schwachstellen, die der veralteten Software zugrunde liegen, sollen mit Einführung des neuen Systems beseitigt werden. Weil GesundPlus mehrere Standorte hat, rechnet das Versicherungsunternehmen mit einem größeren Projektvorhaben, sodass das alte System vollständig nach einer Projektlaufzeit von 12-14 Monaten abgelöst werden sollte. Projektmanagement spielt im Hause GesundPlus schon seit einigen Jahren eine zunehmend wichtige Rolle, sodass es mittlerweile eine kleine Projektmanagement-Abteilung gibt.*

### 5.4.2.1 Bewertung der Projekteigenschaften

#### Komplexität:

- Eher normale Komplexität (Schwierigkeitsgrad eher gering bei hoher Laufzeit)
- Anforderungen sind bekannt und Änderungen eher nicht zu erwarten
- Projekt eher ein größeres Vorhaben

#### Vertragsregelung:

- Kosten und Leistung gut abschätzbar. Dies könnte für einen (Werk-)Vertrag mit Festpreis sprechen

#### Projektmanagement:

- Vorhandene PM-Erfahrung wäre ein Vorteil bei spezifikationsorientierter Umsetzung

### Unternehmensmerkmale:

- Großes, etabliertes Unternehmen, weshalb viele Unternehmensprozesse und Vorgehensweisen einzementiert sein können
- Eher höheres Maß an Betriebsblindheit zu erwarten
- Große Führungsetage, mehrere Stakeholder - daher u.U. kein uneingeschränkter Rückhalt für Projekte

#### 5.4.2.2 *Empfehlung auf Basis der Bewertung*

Die hier erkennbaren Eigenschaften legen den Schluss nahe, dass eine spezifikationsorientierte Umsetzung des Vorhabens sinnvoll ist. Mit eher geringer Komplexität und klaren Anforderungen kann die Planung von *Leistung* und *Kosten* im Vorfeld bereits realistisch durchgeführt werden, sodass auch im Hinblick auf die Perspektive *Zeit* eine sinnvolle Planung entsteht. Da auch umfassende PM-Kenntnisse vorhanden sind, sollte die fortschreitende Projektplanung und -überwachung kein Problem darstellen. Auch die erkennbaren Unternehmensmerkmale sprechen eher für einen spezifikationsorientierten Ansatz, weil große Unternehmen mit festgefahrenen Prozessen und Vorgehensweisen mit Spezifikationsorientierung eher vertraut und kompatibel sind als mit den eher modernen agilen Ansätzen. Da man bei einem derartig prozessorientierten Unternehmen auch in der Führungsetage eher Widerstand gegenüber Projekten - erst recht wenn sie agil sein sollen - erwarten kann, wäre hier eine klassische Herangehensweise wohl die aussichtsreichere Variante. Diese liegt einfach "näher" an der Kultur und an dem Wesen von GesundPlus.

## 6 Fazit und Ausblick

---

Im Verlauf dieser Arbeit wurden die Unterschiede, welche jeweils die agile und die spezifikationsorientierte Projektwelt voneinander abgrenzen, herausgearbeitet. Diese bedingen unterschiedliche Herangehensweisen sowohl beim Projektcontrolling, als auch beim Projektmanagement.

Die Gegenüberstellung von Vorgehensmodellen aus beiden Projektwelten macht die jeweiligen systemischen Vor- und Nachteile deutlich. Allerdings findet ein Projekt immer in einem spezifischen Rahmen statt und muss sich daher stets mit Project-Constraints auseinandersetzen. Die verantwortlichen Entscheidungsgremien (einen Projektmanager gibt es zu diesem frühen Zeitpunkt möglicherweise noch nicht) tun gut daran, diese limitierenden Faktoren möglichst frühzeitig zu identifizieren und zu bewerten.

Auf diesen Überlegungen aufbauend konnte mit *Tabelle 23* und den darin aufgelisteten typischen Projekteigenschaften ein möglicher Ansatz vorgestellt werden, der zur Entscheidungsfindung beitragen könnte. Das Vorgehen dazu wurde anhand von zwei beispielhaften Szenarien aufgezeigt. Dabei könnte die Methode verbessert werden, indem weitere Merkmale ergänzt werden und somit mehr Informationen zur Verfügung stünden. Je mehr Project-Constraints und Projektmerkmale bekannt sind und in die Entscheidungsfindung einfließen, desto qualitativer kann die Entscheidung ausfallen.

„Agil“ ist zwar im Trend, jedoch konnte in dieser Arbeit gezeigt werden, dass nicht jedes Projektvorhaben davon profitieren würde. Auch ist im Hinblick auf die in der Praxis gelebten Ausprägungen, die sich u.a. in den vorgestellten Skalierungsframeworks zeigen, kritisch anzumerken, dass agile Projekte nicht unbedingt zu 100 Prozent dem entsprechen müssen, was das Agile Manifest beschreibt. Mit Sinn und Verstand geführte Projekte profitieren folglich von Vorgehensweisen und Methoden aus beiden Projektwelten.

## Quellenverzeichnis

---

- Arnold, Ross D.; Wade, Jon P. (2015): „A Definition of Systems Thinking: A Systems Approach“. URL: [https://ac.els-cdn.com/S1877050915002860/1-s2.0-S1877050915002860-main.pdf?\\_tid=91a73e50-e72c-46c0-bf1b-73b91f429077&acdnat=1530884462\\_af06d9f1ef42c4ab311839e97bac723a](https://ac.els-cdn.com/S1877050915002860/1-s2.0-S1877050915002860-main.pdf?_tid=91a73e50-e72c-46c0-bf1b-73b91f429077&acdnat=1530884462_af06d9f1ef42c4ab311839e97bac723a) [Stand: 31.07.2018].
- Bibliographisches Institut GmbH (2018): „Duden online, Stichwort agil“. URL: <https://www.duden.de/> [Stand: 31.07.2018].
- Broy, Manfred; Kuhrmann, Marco (2013): Projektorganisation und Management im Software Engineering. Heidelberg: Springer Vieweg (Xpert.press).
- Bundesverband deutscher Banken e. V. (Hrsg.) (2012): „Bargeldlos bezahlen“. URL: <https://bankenverband.de/media/publikationen/bargeldlos-beza.pdf> [Stand: 31.07.2018].
- Cohn, Mike (2016): „Advice on How to Split Reporting User Stories“. URL: <https://www.mountaingoatsoftware.com/blog/advice-on-how-to-split-reporting-user-stories> [Stand: 31.07.2018].
- cPrime (2018): „A Peek Inside Agile. Scrum and Kanban“. URL: <https://www.cprime.com/resource/cprime-online/peek-inside-agile/> [Stand: 31.07.2018].
- Der Beauftragte der Bundesregierung für Informationstechnik (Hrsg.) (2018a): „Das V-Modell XT“. URL: [https://www.cio.bund.de/Web/DE/Architekturen-und-Standards/V-Modell-XT/vmodell\\_xt\\_node.html](https://www.cio.bund.de/Web/DE/Architekturen-und-Standards/V-Modell-XT/vmodell_xt_node.html) [Stand: 31.07.2018].
- Der Beauftragte der Bundesregierung für Informationstechnik (Hrsg.) (2018b): „V-Modell XT Bund“. URL: [http://download.gsb.bund.de/BundesCIO/V-Modell\\_XT\\_Bund/V-Modell-XT-Bund-2.0-Gesamt.pdf](http://download.gsb.bund.de/BundesCIO/V-Modell_XT_Bund/V-Modell-XT-Bund-2.0-Gesamt.pdf) [Stand: 31.07.2018].
- Der Beauftragte der Bundesregierung für Informationstechnik (Hrsg.) (2018c): „Teil 5: V-Modell-Referenz Produkte“. URL: [http://download.gsb.bund.de/BIT/V-Modell\\_XT\\_Bund/V-Modell%20XT%20Bund%20HTML/f436f8cfc083ae.html#reff436f8cfc083ae](http://download.gsb.bund.de/BIT/V-Modell_XT_Bund/V-Modell%20XT%20Bund%20HTML/f436f8cfc083ae.html#reff436f8cfc083ae) [Stand: 31.07.2018].

- Dombrowski, Uwe (Hrsg.) (2015): Lean Development. Aktueller Stand und zukünftige Entwicklungen. Berlin: Springer Vieweg.
- „Domendos“ Consulting It- und Projektberatungs GmbH (2018): „Projektreporting (Projektberichtswesen) – FAQs“. URL: <http://dieprojektmanager.com/faqs-projektreporting/> [Stand: 31.07.2018].
- Engstler, Martin u. a., (Hrsg.) (2016): „Herausforderungen bei der Einführung agiler Vorgehensmodelle für Finanzdienstleister - eine Fallstudie“. URL: <https://dl.gi.de/bitstream/handle/20.500.12116/591/79.pdf> [Stand: 31.07.2018].
- Fiedler, Rudolf (2016): Controlling von Projekten. Mit konkreten Beispielen aus der Unternehmenspraxis – Alle Aspekte der Projektplanung, Projektsteuerung und Projektkontrolle. 7., aktualisierte und überarbeitete Auflage. Wiesbaden: Springer Vieweg.
- Fiedler, Rudolf (2018): „Projektcontrolling“. URL: <https://www.controllingportal.de/upload/iblock/5f1/3dd6eb16d805d6dc881670f6ec1f7186.pdf> [Stand: 22.08.2018].
- Gadatsch, Andreas (2016): IT-Controlling für Einsteiger. Praxiserprobte Methoden und Werkzeuge (essentials). Wiesbaden: Springer Vieweg.
- Goll, Joachim; Hommel, Daniel (2015): Mit Scrum zum gewünschten System. Wiesbaden: Springer Vieweg.
- Hanser, Eckhart (2010): Agile Prozesse: Von XP über Scrum bis MAP (eXamen.press). Berlin: Springer.
- Häusling, André (2016): „Die sechs Dimensionen der agilen Organisation“. URL: [https://www.haufe.de/personal/hr-management/agilitaet/dimensionen-der-agilitaet-agil-werden-in-sechs-schritten\\_80\\_378526.html](https://www.haufe.de/personal/hr-management/agilitaet/dimensionen-der-agilitaet-agil-werden-in-sechs-schritten_80_378526.html) [Stand: 31.07.2018].
- Höhn, Reinhard; Höppner, Stephan (2008): Das V-Modell XT. Anwendungen, Werkzeuge, Standards (eXamen.press). Berlin: Springer.
- InLoox GmbH (2018a): „Meilenstein“. URL: <http://projektmanagement-definitionen.de/glossar/meilenstein/> [Stand: 31.07.2018].



- InLoox GmbH (2018b): „Virtuelles Team“. URL: <http://projektmanagement-definitionen.de/glossar/virtuelles-team/> [Stand: 31.07.2018].
- InLoox GmbH (2018c): „Projektumfeld“.  
URL: <https://www.inloox.de/projektmanagement-glossar/projektumfeld/>  
[Stand: 12.08.2018].
- InLoox GmbH (2018d): „In 4 Schritten zum Worst Practice“.  
URL: <https://www.inloox.de/unternehmen/blog/artikel/in-4-schritten-zum-worst-practice/> [Stand: 16.08.2018].
- itemis AG (Hrsg.) (2018): „Fortschrittskontrolle im Sprint“.  
URL: <https://blogs.itemis.com/de/fortschrittskontrolle-im-sprint>  
[Stand: 31.07.2018].
- Jung, Steffen (2014): „Macht unser Projekt überhaupt Sinn? Die Durchführbarkeitsanalyse am Beispiel eines Softwareprojekts“.  
URL: <https://www.braintool.com/blog/durchfuehrbarkeitsanalyse-softwareprojekt/> [Stand: 31.07.2018].
- Kargl, Herbert; Kütz, Martin (2007): IV-Controlling. 5., vollständig überarbeitete und erweiterte Auflage. München: Oldenbourg Verlag.
- Kerzner, Harold. (2009): Project Management. A Systems Approach to Planning, Scheduling and Controlling. 10. Aufl. Hoboken: John Wiley & Sons.
- Kiesling, Stephan (2011): „Integration agiler Entwicklungszyklen in das V-Modell auf Basis von Informationsflüssen“. URL: <http://www.se.uni-hannover.de/pub/File/pdfpapers/Kiesling2011.pdf> [Stand: 31.07.2018].
- Lent, Bogdan (2013): IT-Management als kybernetisches System. Wiesbaden: Springer Fachmedien.
- Leopold, Klaus; Kaltenecker, Siegfried (2013): Kanban in der IT. Eine Kultur der kontinuierlichen Verbesserung schaffen. 2., überarbeitete Auflage. München: Hanser.

- Lindner, Dominic (2017): „WIE KANN MAN EIGENTLICH SCRUM SKALIEREN?“. URL: <https://agile-unternehmen.de/scrum-skalieren/> [Stand: 31.07.2018].
- Maximini, Dominik (2018): Scrum – Einführung in der Unternehmenspraxis. Von starren Strukturen zu agilen Kulturen. 2., aktualisierte und erweiterte Auflage. Berlin: Springer-Verlag.
- McManus, John (2004): Managing Stakeholders in Software Development Projects. Oxford: Elsevier Butterworth-Heinemann.
- Montag, Torsten (2012a): „Projektmanagement: Was ist ein Lastenheft?“. URL: <https://www.gruenderlexikon.de/magazin/projektmanagement-was-ist-ein-lastenheft-2128> [Stand: 31.07.2018].
- Montag, Torsten (2012b): „Projektmanagement: Was ist ein Pflichtenheft?“. URL: <https://www.gruenderlexikon.de/magazin/projektmanagement-was-ist-ein-pflichtenheft-2129> [Stand: 31.07.2018].
- Muth, Bertil; Valentini, Uwe (2017): „Agiles Manifest für Manager“. URL: [https://www.sigs-datacom.de/uploads/tx\\_dmjournals/Muth\\_Valentini\\_OTs\\_Agility\\_16.pdf](https://www.sigs-datacom.de/uploads/tx_dmjournals/Muth_Valentini_OTs_Agility_16.pdf) [Stand: 31.07.2018].
- Orientation in Objects GmbH (Hrsg.) (2018): „SAFe Voraussetzungen“. URL: <https://www.braintime.de/methoden/ueberblick-scaled-agile-framework-beratung/safe-voraussetzungen/> [Stand: 31.07.2018].
- Pavlik, Franz (2018): „Scheitern von IT-Projekten“. URL: <http://dieprojektmanager.com/scheitern-von-it-projekten/> [Stand: 31.07.2018].
- Poppendieck, Tom (2003): „The Agile Customer’s Toolkit“. URL: <http://www.torak.com/files/The%20Agile%20Customer%E2%80%99s%20Toolkit.pdf> [Stand: 31.07.2018].
- Preißler, Peter R. (2014): Controlling. 14., überarb. und erw. Aufl. München: Oldenbourg Verlag.

- Project Management Institute, Inc. (2012): PMI Lexicon of Project Management Terms. Newtown Square: PMI.
- Project Management Institute, Inc. (2018): „What is Project Management?“. URL: <https://www.pmi.org/about/learn-about-pmi/what-is-project-management> [Stand: 31.07.2018].
- Prudix, Dietmar (2016): Erfolgreiches Projektmanagement. Vom sicheren Umgang mit Menschen in Projekten. Wiesbaden: Springer Gabler.
- Rau, Karl-Heinz (2016): Agile objektorientierte Softwareentwicklung. Schritt für Schritt vom Geschäftsprozess zum Java-Programm. 1. Aufl. Wiesbaden: Springer Fachmedien.
- Scaled Agile, Inc. (Hrsg.) (2018a): „SAFe“. URL: <https://www.scaledagileframework.com> [Stand: 31.07.2018].
- Scaled Agile, Inc. (Hrsg.) (2018b): „What Is SAFe?“. URL: <https://www.scaledagileframework.com/what-is-safe/> [Stand: 31.07.2018].
- Schimpf, Sven (2017): „Agil oder nicht agil, das ist hier die Frage!“. URL: <https://blog.iao.fraunhofer.de/agil-oder-nicht-agil-das-ist-hier-die-frage/> [Stand: 31.07.2018].
- Schwaber, Ken; Scrum.org (Hrsg) (2018): „Nexus Guide. Der gültige Leitfaden zur Skalierung von Scrum mit dem Nexus: Die Spielregeln“. URL: <https://scrumorg-website-prod.s3.amazonaws.com/drupal/2018-01/2018-Nexus-Guide-German.pdf?nexus-file=https%3A%2F%2Fscrumorg-website-prod.s3.amazonaws.com%2Fdrupal%2F2018-01%2F2018-Nexus-Guide-German.pdf> [Stand: 31.07.2018].
- Schwaber, Ken; Sutherland, Jeff (2016): „Der Scrum Guide. Der gültige Leitfaden für Scrum: Die Spielregeln“. URL: <https://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-German.pdf> [Stand: 31.07.2018].

- Scrum.org (Hrsg) (2018): „A Framework to Help Organizations Scale Scrum“.  
URL: <https://www.scrum.org/resources/scaling-scrum>  
[Stand: 22.08.2018].
- Seidl, Jörg (2011): Multiprojektmanagement. Übergreifende Steuerung von Mehrprojektsituationen durch Projektportfolio- und Programmmanagement (Xpert.press). Heidelberg: Springer.
- Springer Gabler Verlag (Hrsg.) (2018a): „Gabler Wirtschaftslexikon, Stichwort: Geschäftsprozess“.  
URL: <https://wirtschaftslexikon.gabler.de/definition/geschaeftsprozess-35399/version-258881> [Stand: 31.07.2018].
- Springer Gabler Verlag (Hrsg.) (2018b): „Gabler Wirtschaftslexikon, Stichwort: Strategisches Management“.  
URL: <https://wirtschaftslexikon.gabler.de/definition/strategisches-management-46326/version-269608> [Stand: 31.07.2018].
- Springer Gabler Verlag (Hrsg.) (2018c): „Gabler Wirtschaftslexikon, Stichwort: Projektmanagement“.  
URL: <https://wirtschaftslexikon.gabler.de/definition/projektmanagement-pm-46130/version-269416> [Stand: 31.07.2018].
- Springer Gabler Verlag (Hrsg.) (2018d): „Gabler Wirtschaftslexikon, Stichwort: Berichtswesen“.  
URL: <https://wirtschaftslexikon.gabler.de/definition/berichtswesen-31540/version-255096> [Stand: 31.07.2018].
- Springer Gabler Verlag (Hrsg.) (2018e): „Gabler Wirtschaftslexikon, Stichwort: Risikomanagement“.  
URL: <https://wirtschaftslexikon.gabler.de/definition/risikomanagement-42454/version-265802> [Stand: 20.08.2018].
- Springer Gabler Verlag (Hrsg.) (2018f): „Gabler Wirtschaftslexikon, Stichwort: Wirtschaftlichkeit“.  
URL: <https://wirtschaftslexikon.gabler.de/definition/wirtschaftlichkeit-47252/version-270518> [Stand: 20.08.2018].

- Springer Gabler Verlag (Hrsg.) (2018g): „Gabler Wirtschaftslexikon, Stichwort: Nutzwertanalyse“.  
URL: <https://wirtschaftslexikon.gabler.de/definition/nutzwertanalyse-42926/version-266266> [Stand: 31.07.2018].
- Springer Gabler Verlag (Hrsg.) (2018h): „Gabler Wirtschaftslexikon, Stichwort: Controlling“.  
URL: <https://wirtschaftslexikon.gabler.de/definition/controlling-30235/version-253823> [Stand: 31.07.2018].
- Springer Gabler Verlag (Hrsg.) (2018i): „Gabler Wirtschaftslexikon, Stichwort: Balanced Scorecard“.  
URL: <https://wirtschaftslexikon.gabler.de/definition/balanced-scorecard-28000/version-251640> [Stand: 31.07.2018].
- Springer Gabler Verlag (Hrsg.) (2018j): „Gabler Wirtschaftslexikon, Stichwort: SWOT-Analyse“.  
URL: <https://wirtschaftslexikon.gabler.de/definition/swot-analyse-52664/version-275782> [Stand: 31.07.2018].
- Springer Gabler Verlag (Hrsg.) (2018k): „Gabler Wirtschaftslexikon, Stichwort: Change Management“.  
URL: <https://wirtschaftslexikon.gabler.de/definition/change-management-28354/version-251986> [Stand: 31.07.2018].
- Steimle, Toni (2007): Softwareentwicklung im Offshoring. Erfolgsfaktoren für die Praxis (Xpert.press). Heidelberg: Springer.
- the Agile Manifesto authors (Hrsg.) (2018a): „Manifest für Agile Softwareentwicklung“.  
URL: <http://agilemanifesto.org/iso/de/manifesto.html> [Stand: 31.07.2018].
- the Agile Manifesto authors (Hrsg.) (2018b): „Prinzipien hinter dem Agilen Manifest.“  
URL: <http://agilemanifesto.org/iso/de/principles.html> [Stand: 31.07.2018].
- The LeSS Company B.V. (2018a): „LeSS Framework“.  
URL: <https://less.works/less/framework/index.html> [Stand: 31.07.2018].
- The LeSS Company B.V. (2018b): „LeSS Huge“.  
URL: <https://less.works/less/less-huge/index.html> [Stand: 31.07.2018].

The LeSS Company B.V. (2018c): „Overall Retrospective“.

URL: <https://less.works/less/framework/overall-retrospective.html>

[Stand: 22.08.2018].

The LeSS Company B.V. (2018d): „Requirement Areas“.

URL: <https://less.works/less/less-huge/requirement-areas.html>

[Stand: 31.07.2018].

Wirtschaftslexikon24 (2018): „Total Quality Management (TQM)“.

URL: <http://www.wirtschaftslexikon24.com/d/total-quality-management-tqm/total-quality-management-tqm.htm> [Stand: 31.07.2018].

yeebase media GmbH (Hrsg.) (2013): „Agiles Projektmanagement: Scrum, Kanban und Scrumbuts im Einsatz“. URL:

<https://t3n.de/magazin/praxisbericht-scrum-kanban-scrumbuts-agiles-232822/> [Stand: 31.07.2018].

yeebase media GmbH (Hrsg.) (2016): „Was? Pflichtenheft und Lastenheft sind nicht dasselbe?“. URL: <https://t3n.de/news/pflichtenheft-lastenheft-unterschied-523402/>

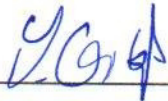
[Stand: 31.07.2018].

Young, Trevor L. (2007): The Handbook of Project Management. A practical guide to effective policies, techniques and processes. 2. Aufl. London: Kogan Page Limited.

## Selbstständigkeitserklärung

---

Diese Arbeit wurde von mir selbständig verfasst und in gleicher oder ähnlicher Fassung noch nicht in einem anderen Studiengang als Prüfungsleistung vorgelegt. Ich habe keine anderen als die angegebenen Hilfsmittel und Quellen, einschließlich der angegebenen oder beschriebenen Software, verwendet.



---

Lisa Grotepaß

Bernburg, den 26.09.2018