



Masterarbeit

Einsatz von agilen Methoden in einem global agierenden Unternehmen unter Betrachtung deren Anwendung bei verschiedenen Zusammenarbeitsmodellen

Philipp Gruber, BSc

Mai, 2019

Betreuer: Dipl.-Ing. Dipl.-Ing. Michael Rachinger
Begutachter: Univ.-Prof. Dipl.-Ing. Dr.techn. Stefan Vorbach
Firmen
Betreuer: Dipl.-Ing. Dr.techn. Christoph Falk-Gierlinger

Technische Universität Graz

Institut für Unternehmensführung und Organisation



EIDESSTATTLICHE ERKLÄRUNG***AFFIDAVIT***

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master thesis.

Datum /Date

Unterschrift/Signature

Kurzfassung

Eine klassische Projektabwicklung kann bei sich ändernden Marktbedingungen oder dem Kundenwunsch nach mehr Flexibilität schnell an seine Grenzen stoßen und damit den Projekterfolg in Gefahr bringen. ALTEN sieht in der agilen Projektabwicklung das Potential, diesem Problem entgegenwirken zu können.

Das Ziel der vorliegenden Masterarbeit war es, mögliche Anpassungen bzw. Änderungen in Bezug auf die unterschiedlichen ALTEN-internen Zusammenarbeitsmodelle (IQP-Modelle) zu identifizieren, um ein agiles Projekt erfolgreich abwickeln zu können. Auch wurde untersucht, worin ALTEN die Vorteile von agiler Projektabwicklung sieht und welche Grundvoraussetzungen für eine erfolgreiche agile Abwicklung vorhanden sein müssen.

Zur Beantwortung der Forschungsfragen wurden eine ausführliche Literaturrecherche und mehrere ALTEN-interne Experteninterviews durchgeführt. Die bei den Interviews erhaltenen Daten wurden vollständig transkribiert und einer qualitativen Inhaltsanalyse nach Mayring (Mayring, 2002) unterzogen.

Die gewonnenen Ergebnisse zeigen, dass eine agile Projektabwicklung neben den wirtschaftlichen Vorteilen auch zu einer gesteigerten Kunden- und Mitarbeiterzufriedenheit führt. Als wichtigste Grundvoraussetzung für agile Abwicklung wurde das agile Mindset identifiziert. Sowohl der Kunde als auch das Projektteam auf ALTEN-Seite müssen hierzu die agile Arbeitsweise verstehen, akzeptieren und anwenden können. Um agile Ansätze erfolgreich in zukünftige Projekte implementieren zu können, ist es notwendig, Anpassungen im Angebot bzw. Vertragswerk vorzunehmen. In standardisierten Angebotsvorlagen sollten die agilen Grundsätze, die agilen Arbeitsweisen und in diesem Zusammenhang stehenden Rechte und Pflichten von ALTEN und Kunden verständlich niedergeschrieben werden.

Abstract

A classic project approach can quickly reach its limits for changing market conditions or customers desire for more flexibility and therefore threaten the project success. ALTEN sees the potential in an agile project approach to counteract this problem.

The aim of this master thesis was to identify possible adjustments or changes in relation to the different ALTEN internal collaboration models (IQP models) in order to successfully transact an agile project. Furthermore, it was examined, wherein ALTEN sees the advantages of an agile project approach and which basic requirements need to be present for a successful completion of an agile project.

To answer the different research questions a detailed literature research and several in-house expert interviews were conducted. The data obtained in the interviews were fully transcribed and analyzed with the qualitative content analysis of Mayring (Mayring, 2002).

The gained results show that an agile project approach does not only lead to economic benefits but also to increased customer and employee satisfaction. As the most important basic requirement the agile mindset was identified. Both the customer and the ALTEN project team have to understand, accept and deploy the agile operating principles. In order to successfully implement agile approaches in future projects, it is necessary to make adjustments to the offer or contract. Standard templates should clearly describe the agile principle, the agile way of working and in this context all related rights and obligations of ALTEN and its customers.

Inhaltsverzeichnis

Inhaltsverzeichnis	V
Abbildungsverzeichnis.....	VIII
Symbolverzeichnis und Abkürzungen	X
Tabellenverzeichnis	XI
1 Einleitung.....	1
1.1 Problemstellung	1
1.2 Zielsetzung und Forschungsfragen	2
1.3 Aufbau der Masterarbeit	3
2 Theoretische Grundlagen.....	4
2.1 Agiles Mindset.....	4
2.1.1 Agile Werte.....	5
2.1.2 Agile Prinzipien	5
2.1.3 Agile Techniken	6
2.1.4 Agile Methoden	15
2.2 Scrum – Grundlagen.....	16
2.2.1 Rollen.....	18
2.2.2 Artefakte	19
2.2.3 Ereignisse.....	20
2.3 Scrum – Erweiterungen	23
2.3.1 Projektanforderungen	23
2.3.2 Aufwandsabschätzung	23
2.3.3 Projektplanung.....	25
2.4 Scrum – Prozessmodell	27
3 Klassisch versus Agil	31
3.1 Unterschiede und Gemeinsamkeiten	31
3.1.1 Gemeinsame Basis.....	31
3.1.2 Zieldreieck	32
3.1.3 Rahmenbedingungen	33
3.1.4 Die Unterschiede im Überblick	34
3.2 Gründe für nicht erfolgreiche Projekte	35
3.2.1 Allgemeine Projekte	35

3.2.2	IT-Projekte.....	36
3.3	Einsatzgebiete.....	37
3.4	Vorteile eines agilen Ansatzes.....	40
4	Zusammenarbeitsmodelle	43
4.1	Abgrenzung	43
4.2	Werkverträge	44
4.2.1	IQP-3 Modell.....	45
4.2.2	IQP-4 Modell.....	46
4.2.3	IQP-5 Modell.....	47
5	Agile Vertragsgestaltung.....	48
5.1	Wichtige Aspekte für agile Vertragsgestaltung.....	48
5.1.1	Unvollständige Funktionsbeschreibung	49
5.1.2	Gemeinsames Verständnis.....	49
5.1.3	Kooperative Problemlösung	49
5.1.4	Rollen.....	50
5.1.5	Artefakte	51
5.1.6	Ereignisse.....	51
5.2	Klassischer Festpreis	51
5.3	Agile Abwandlungen.....	53
5.3.1	Klassischer Festpreis mit agilen Techniken	53
5.3.2	Festpreis mit garantiertem Minimalumfang	54
5.3.3	Money for Nothing, Change for Free	55
5.3.4	Der agile Festpreis	57
6	Forschungsfragen.....	61
7	Methodisches Vorgehen	63
7.1	Datenerhebung.....	63
7.1.1	Interviewleitfaden.....	63
7.1.2	Auswahl der Experten	64
7.1.3	Durchführung des Interviews	64
7.1.4	Transkription	64
7.2	Datenauswertung	65
7.2.1	Qualitative Inhaltsanalyse.....	65

7.2.2	Ablaufmodell	66
8	Ergebnisse.....	72
8.1	Hauptprobleme bei klassischer Projektabwicklung.....	73
8.2	Vorteile von agiler Projektabwicklung.....	76
8.3	Grundvoraussetzungen für agile Projektabwicklung.....	79
8.4	Anpassungen und Änderungen.....	82
8.5	Agile Projektabwicklung mit Scrum für ein Festpreisprojekt.....	86
9	Fazit.....	89
	Literaturverzeichnis	92
	Anhang.....	94
A.	Gesamter Leitfaden der Experteninterviews.....	94
B.	Interviewberichte	96

Abbildungsverzeichnis

Abbildung 1: Agiles Mindset	4
Abbildung 2: Agile Techniken	7
Abbildung 3: Anforderungen.....	8
Abbildung 4: Relative Schätzung	9
Abbildung 5: Task Board	10
Abbildung 6: WIP-Limit	11
Abbildung 7: Pareto Prinzip	13
Abbildung 8: Burn-Down-Chart.....	14
Abbildung 9: Velocity	15
Abbildung 10: Scrum Framework	16
Abbildung 11: Vereinfachter Scrum Prozess	17
Abbildung 12: Iterativ und Inkrementell	20
Abbildung 13: Ein Sprint als Timebox.....	21
Abbildung 14: Priorisiertes und geschätztes Product Backlog.....	25
Abbildung 15: Erweiterter Scrum Prozess	27
Abbildung 16: Grundlegender Projektmanagementprozess	31
Abbildung 17: Zieldreieck.....	32
Abbildung 18: Misserfolgskfaktoren für ein Projekt.....	35
Abbildung 19: Anteil erfolgreicher bzw. gescheiterter IT-Projekte.....	36
Abbildung 20: Stacey Landscape Diagram	38
Abbildung 21: Zusammenarbeitsmodelle von ALTEN.....	43
Abbildung 22: Werkvertragsmodelle	44
Abbildung 23: IQP-3 Modell.....	45
Abbildung 24: IQP-4 Modell.....	46
Abbildung 25: IQP-5 Modell.....	47
Abbildung 26: Schematischer Ablauf einer klassischen Beauftragung.....	52
Abbildung 27: Klassischer Festpreis mit agilen Techniken	53
Abbildung 28: Festpreis mit garantiertem Minimalumfang	55
Abbildung 29: Change for Free	56
Abbildung 30: Money for Nothing.....	56
Abbildung 31: Detaillierung der Produktvision beim agilen Festpreis	58

Abbildung 32: Ablaufmodell der qualitativen Inhaltsanalyse mit induktiver Kategorienbildung	66
Abbildung 33: Kategorisierung mit QCMap.....	69
Abbildung 34: Beispiel zu einem Kategoriensystem	70
Abbildung 35: Kategoriensystem - Forschungsfrage 1	73
Abbildung 36: Zitate der Experteninterviews zu Forschungsfrage 1	74
Abbildung 37: Kategoriensystem - Forschungsfrage 2	76
Abbildung 38: Zitate der Experteninterviews zu Forschungsfrage 2	78
Abbildung 39: Kategoriensystem - Forschungsfrage 3	79
Abbildung 40: Zitate der Experteninterviews zu Forschungsfrage 3	80
Abbildung 41: Kategoriensystem - Forschungsfrage 4	82
Abbildung 42: Zitate der Experteninterviews zu Forschungsfrage 4	83
Abbildung 43: Zitate der Experteninterviews zu Forschungsfrage 4	85

Symbolverzeichnis und Abkürzungen

<i>Zeichen</i>	<i>Einheit</i>	<i>Bedeutung</i>
----------------	----------------	------------------

Abkürzungen

IQP	[-]	Indicator Quality Performance
WIP	[-]	Work in Progress
PBIs	[-]	Product Backlog Items

Tabellenverzeichnis

Tabelle 1: Optimale Rahmenbedingungen für agiles Vorgehen	33
Tabelle 2: Wesentliche Unterschiede zwischen klassischen und agilen Projekten	34
Tabelle 3: Misserfolgskriterien nach Chaos-Report	37
Tabelle 4: Vorteile von agilen Ansätzen	40
Tabelle 5: Abgrenzung Arbeitnehmerüberlassung zu Werkvertrag	44
Tabelle 6: Vorteile agiler Techniken im Festpreis	54
Tabelle 7: Kategoriendefinition und Abstraktionsniveau.....	68

1 Einleitung

Die vorliegende Masterarbeit wurde in Zusammenarbeit mit dem Unternehmen ALTEN Austria Süd GmbH sowie dem Institut für Unternehmensführung und Organisation der Technischen Universität Graz erstellt. Dieses Kapitel beinhaltet die Einleitung der Masterarbeit. Es erfolgt eine Gliederung in Kapitel 1.1 Problemstellung, Kapitel 1.2 Zielsetzung und Forschungsfragen und Kapitel 1.3 Aufbau der Masterarbeit.

1.1 Problemstellung

Das Unternehmen ALTEN ist ein global agierender Ingenieurdienstleister, dessen Tätigkeitsbereich sich auf die Geschäftsfelder Automotiv, Aerospace, IT, Energy und Telecom verteilt. Die Themenfelder der angebotenen Dienstleistungen reichen von Software- und Hardwareentwicklung über Simulation und Berechnung, Produktions- und Prozessplanung bis hin zu Projekt- und Qualitätsmanagement. Eine gute Kooperation mit dem Kunden sowie eine effektive Projektplanung und Durchführung ist in dieser Branche zwingend erforderlich, um langfristig erfolgreich bestehen zu können. Dementsprechend wird der Abwicklung von Werkverträgen große Beachtung geschenkt. Ein Werkvertrag ist folgendermaßen definiert:

„Vertrag, durch den sich der Unternehmer zur Herstellung eines Werkes, der Besteller zur Zahlung einer Vergütung verpflichtet“ (Berwanger, 2018)

So einfach diese Definition auch klingt, so schwer ist es, diese in der Praxis erfolgreich umzusetzen. Änderungswünsche des Kunden nach Projektstart, unklare Projektanforderungen zu Beginn, falsche Aufwandsabschätzungen oder Unzufriedenheit des Kunden mit den Ergebnissen sind nur einige wenige Beispiele, die im Zuge eines Projektes immer wieder zu Problemen und Unstimmigkeiten zwischen Kunde und Dienstleister führen. Der überwiegende Anteil aller Projekte bei ALTEN bedient sich dem altbewährten, klassischen Projektmanagement. Die Daseinsberechtigung dieses Ansatzes steht außer Frage, ob dieser jedoch für jedes Projekt die geeignetste Wahl darstellt, darf hinterfragt werden. Ändern sich Marktbedingungen, fordert der Kunde mehr Flexibilität oder Planungsfreiheit, stößt das klassische Projektmanagement mit seinem starren Planungsgerüst sehr schnell an seine Grenzen und oben genannte

Probleme können den Projekterfolg in Gefahr bringen. Dies sind nur einige von vielen Gründen, alternative Ansätze in Betracht zu ziehen.

1.2 Zielsetzung und Forschungsfragen

ALTEN besitzt 5 unterschiedliche Zusammenarbeitsmodelle, auch IQP-Modelle genannt (IQP 1 bis 5), die die Zusammenarbeit in Bezug auf die Aufwandsabrechnung und Angebotsgestaltung mit dem Kunden regeln. Die Anwendung der unterschiedlichen IQP-Modelle hängt von den Präferenzen des Kunden und der Eigenverantwortung, die ALTEN im Rahmen des Projektes übernimmt, ab. So bezeichnet man das Modell IQP-5 auch als Festpreisprojekt, bei dem ALTEN die gesamte Entwicklungsverantwortung trägt. Bei den Modellen IQP-3 und IQP-4 erfolgt nur eine Beauftragung von einzelnen Phasen eines Gesamtprojektes. IQP 3 bis 5 fallen somit in den Bereich der Werkverträge, wohingegen sich die Modelle IQP-1 und IQP-2 auf Arbeitnehmerüberlassung beziehen und daher in der Masterarbeit nicht weiter behandelt werden.

Wie schon im vorherigen Kapitel (siehe 1.1) angeführt, benötigt es alternative Projektmanagement Ansätze, um in einem dynamischen Markt dem Kundenwunsch nach mehr Flexibilität und Planungsfreiheit gerecht zu werden. ALTEN sieht in der agilen Projektabwicklung, im speziellen in dem agilen Vorgehensmodell Scrum, Potential, oben genannte Probleme zu adressieren und zu kompensieren. Das volle Potential agiler Abwicklung kann aber nur dann ausgeschöpft werden, wenn es richtig angewendet wird und sich das Projekt sowie das Zusammenarbeitsmodell für das agile Vorgehen eignen. Im Zuge der Masterarbeit sollen somit folgende Forschungsfragen beantwortet werden:

- F1:** *Worin liegen für ALTEN die Hauptprobleme bei klassischer Projektabwicklung?*
- F2:** *Worin sieht ALTEN die Vorteile von agiler Projektabwicklung?*
- F3:** *Welche Grundvoraussetzungen müssen für eine agile Projektabwicklung gegeben sein?*
- F4:** *Welche Anpassungen und Änderungen müssen für ein agiles Projekt in Bezug auf die unterschiedlichen IQP-Modelle getroffen werden, damit dieses erfolgreich abgewickelt werden kann?*
- F5:** *Wie könnte eine agile Projektabwicklung mit Scrum für ein Festpreisprojekt (IQP-5) aussehen und welche Vertragsinhalte müssen dabei berücksichtigt werden?*

Die Beantwortung oben genannter Forschungsfragen erfolgt mittels Literaturrecherche und firmeninternen Experteninterviews. Als Basis hierfür dienen semi-strukturierte Interviews, die anschließend mittels „qualitativer Inhaltsanalyse nach Mayring“ ausgewertet werden.

Die Ergebnisse sollen ALTEN einerseits einen Grundstock sinnvoller Handlungsempfehlungen bereitstellen, wie agile Ansätze erfolgreich in bestehende Prozesse und zukünftige Projekte implementiert werden können und andererseits als Grundlage für mögliche, zukünftige Bachelor- und Masterarbeiten im Bereich der agilen Projektabwicklung dienen.

1.3 Aufbau der Masterarbeit

Im Kapitel 2 (Theoretische Grundlagen) werden die Bausteine des agilen Projektmanagements sowie Grundlagen zur Scrum Methodik näher erläutert. Diese Informationen werden in Kapitel 3 (Klassisch versus Agil) den klassischen Projektmanagement Ansätzen gegenübergestellt, Unterschiede sowie Gemeinsamkeiten beleuchtet und Vorteile der agilen Ansätze erarbeitet. Kapitel 4 (Zusammenarbeitsmodelle) gibt einen kurzen Einblick in die unterschiedlichen Zusammenarbeitsmodelle (IQP 3-5) und skizziert den Untersuchungsbereich der Masterarbeit. Einen wesentlichen Erfolgsfaktor bei Projekten stellt die Vertragsgestaltung zwischen Dienstleister und Kunde dar. Die Notwendigkeit, Anpassungen des Vertragskonstrukts bei agilem Vorgehen vorzunehmen und „Best Practices“ hierzu, werden in Kapitel 5 (Agile Vertragsgestaltung) diskutiert. Im darauffolgenden Kapitel (6 Forschungsfragen) werden nochmals die Forschungsfragen angeführt. Das methodische Vorgehen, der Interviewleitfaden, die Interviewpartner sowie die Datenauswertung werden in Kapitel 7 (Methodisches Vorgehen) näher beschrieben. Die aus der Literaturrecherche sowie den Experteninterviews gewonnenen Erkenntnisse werden im vorletzten Kapitel (8 Ergebnisse) intensiv diskutiert. Abschließend erfolgt eine kurze Zusammenfassung sowie ein Ausblick (9 Fazit).

2 Theoretische Grundlagen

In diesem Kapitel werden alle notwendigen Grundlagen angeführt, um agiles Projektmanagement sowie das Vorgehensmodell Scrum in seiner Gesamtheit verstehen zu können. Dafür werden zuerst die Bausteine des agilen Projektmanagements (siehe Kapitel 2.1) beschrieben. Darauf aufbauend werden die Grundlagen zur Scrum Methodik (siehe Kapitel 2.2) und sinnvolle Erweiterungen (siehe Kapitel 2.3) angeführt. Im abschließenden Kapitel (siehe Kapitel 2.4) wird das Scrum Prozessmodell im Detail erklärt.

2.1 Agiles Mindset

Das agile Mindset, auch „Bausteine des agilen Projektmanagements“ genannt, dient als Hyperonym und beinhaltet die agilen Werte, agilen Prinzipien, agilen Techniken und agilen Methoden. Wie in Abbildung 1 zu sehen, bauen die einzelnen Bausteine aufeinander auf. Die agilen Werte stellen hier das Fundament aller agilen Arbeitsweisen dar und werden durch 4 Gegensatzpaare beschrieben (siehe Kapitel 2.1.1). Auf diesen Werten basieren die agilen Prinzipien, die die Handlungsgrundsätze bilden. Zur praktischen Umsetzung dieser werden agile Techniken eingesetzt. Die dafür benötigte Struktur wird durch agile Methoden gegeben. (Preußig, 2015, S. 9)

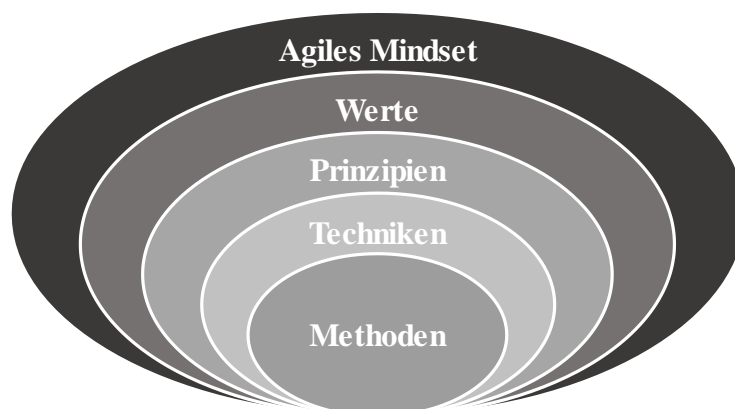


Abbildung 1: Agiles Mindset
Quelle: in Anlehnung an (Powers, 2017)

Für voll funktionsfähige, agile Methoden müssen also nicht nur agile Techniken beherrscht, sondern auch agile Werte und agile Prinzipien berücksichtigt werden. In den

folgenden Kapiteln werden die oben genannten Bausteine des agilen Projektmanagements näher beschrieben.

2.1.1 Agile Werte

Kent Beck, Jeff Sutherland, Ken Schwaber und weitere 14 Vordenker im Bereich der Softwareentwicklung verfassten 2001 in den USA das sogenannte „Agile Manifest“. Ihr Ziel war es, der sich langsam abzeichnenden agilen Bewegung einen Namen zu geben, um möglichst viele Gleichgesinnte zu erreichen. Dabei definierten sie 4 Leitsätze, auch agile Werte genannt, die basierend auf ihrem fundierten Expertenwissen zu nachhaltig besserer Softwareentwicklung führen sollten. Im Folgenden werden die, aus dem Englischen übersetzten, agilen Werte angeführt (Beck, et al., 2001):

- *„Individuen und Interaktionen wichtiger als Prozesse und Werkzeuge*
- *Funktionierende Software wichtiger als umfassende Dokumentation*
- *Kundenzusammenarbeit wichtiger als Vertragsverhandlungen*
- *Reaktion auf Veränderung wichtiger als das Befolgen eines Plans“*

Laut den Verfassern des Agilen Manifests werden die auf der rechten Seite angeführten Werte (Prozesse, Werkzeuge, Dokumentation, Vertragsverhandlungen und Pläne) zwar als wichtig erachtet, die linken Werte (Individuen, Funktionierende Software, Kundenzusammenarbeit und Reaktion auf Veränderungen) aber höher eingeschätzt. (Pieper & Roock, 2017, S. 2)

2.1.2 Agile Prinzipien

Neben den 4 agilen Werten wurden im „Agilen Manifest“ 12 agile Prinzipien niedergeschrieben, die die Wertaussagen genauer charakterisieren und die Auswirkung dieser auf die tägliche Arbeit näher definieren (Pieper & Roock, 2017, S. 3):

- *„Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen.*
- *Heiße Anforderungsänderungen, selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.*

- *Liefere funktionierende Software, regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.*
- *Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.*
- *Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.*
- *Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.*
- *Funktionierende Software ist das wichtigste Fortschrittsmaß.*
- *Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten ein gleichmäßiges Tempo auf unbegrenzte Zeit halten können.*
- *Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.*
- *Einfachheit, die Kunst, die Menge nicht getaner Arbeit zu maximieren, ist essenziell.*
- *Die besten Architekturen, Anforderungen und Entwürfe, entstehen durch selbstorganisierte Teams.*
- *In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an.“*

Zusammenfassend kann gesagt werden, dass für eine erfolgreiche agile Arbeitsweise laut Definition die Werte sowie die daraus resultierenden Prinzipien vollständig verstanden und noch wichtiger, auch angewendet werden müssen. Die agilen Prinzipien geben uns zur Anwendung aber keine Werkzeuge in die Hand, dies wird durch die sogenannten „agilen Techniken“ bewerkstelligt. Im nächsten Kapitel werden einige der wichtigsten Techniken kurz beschrieben.

2.1.3 Agile Techniken

Der Rahmen eines agilen Projektes wird durch die agilen Werte und Prinzipien definiert. Die agilen Techniken dagegen stellen konkrete Verfahren dar, um alle oben genannten Prinzipien in der Praxis anwenden zu können. Der Großteil der agilen Techniken (siehe Abbildung 2), beschäftigt sich mit dem Management von Anforderungen, Tasks,

Meetings und dem Fortschritt des Projektes. Im Folgenden werden die wichtigsten agilen Techniken kurz angeschnitten und erklärt. Deren konkrete Anwendung in einem agilen Projekt wird in Kapitel 2.2 (Scrum – Grundlagen) diskutiert. (Preußig, 2015, S. 78-79)



Abbildung 2: Agile Techniken

Quelle: In Anlehnung an (Preußig, 2015, S. 77-130)

User Stories

Bei der agilen Technik „User Stories“ werden die Anforderungen an das Produkt nur aus Sicht des Kunden bzw. Anwenders beschrieben. Der Kunde definiert also in den User Stories was er, bzw. der Anwender mit dem fertigen Produkt machen will und welchen Nutzen es haben soll. Dies sieht folgendermaßen aus:

„Als <Rolle> möchte ich <Funktion> um <Nutzen>“ (Plewa, 2017)

Ein Beispiel hierfür wäre: Als <Student der TU Graz> möchte ich <mich im TeachCenter anmelden>, um <mich für diverse Vorlesungen und Übungen bestmöglich vorzubereiten>. Diese Nutzerdarstellung führt einerseits bei dem Entwickler zu einem besseren Verständnis der Anforderungen, es kommt also zu weniger Missverständnissen, andererseits kann sich auch der Kunde einfacher bei Entscheidungen sowie Priorisierungen einbringen, da die Anforderungsbeschreibung mit der Sprache des Kunden und nicht des Dienstleisters formuliert wurde. (Preußig, 2015, S. 92-93)

Epics

User Stories sind meist sehr detailliert beschrieben, bei vielen Anforderungen kann man hier sehr schnell den Überblick verlieren. Mittels einem Epic lassen sich mehrere, miteinander verknüpfte User Stories zusammenfassen. Dies ermöglicht, das Projekt zu Beginn besser zu strukturieren und wie schon erwähnt, bei größeren Projekten den Überblick zu bewahren (siehe Abbildung 3). (Preußig, 2015, S. 94)

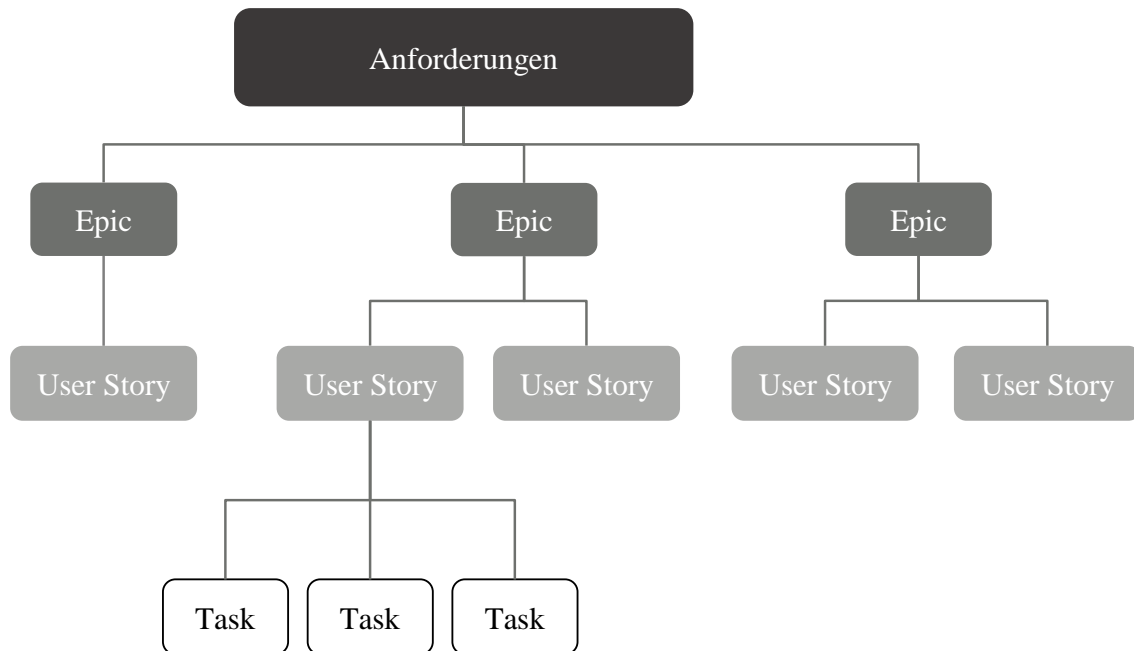


Abbildung 3: Anforderungen
Quelle: In Anlehnung an (Plewa, 2017)

Story Points

In klassischen Projekten wird hauptsächlich mit Personenstunden oder Personentagen gearbeitet. Die Erfahrung hat aber gezeigt, dass Menschen mit absoluten Schätzungen sehr häufig Schwierigkeiten haben. Laut Rubin (Rubin, 2014) können mit relativen Schätzungen hingegen, sehr viel bessere Ergebnisse erzielt werden. So fällt es zum Beispiel schwer, das Gewicht eines Steines zu schätzen, eine relative Größenabschätzung zu anderen Steinen hingegen ist problemlos möglich und liefert auch noch sehr gute Ergebnisse (siehe Abbildung 4). (Rubin, 2014, S. 163-164)

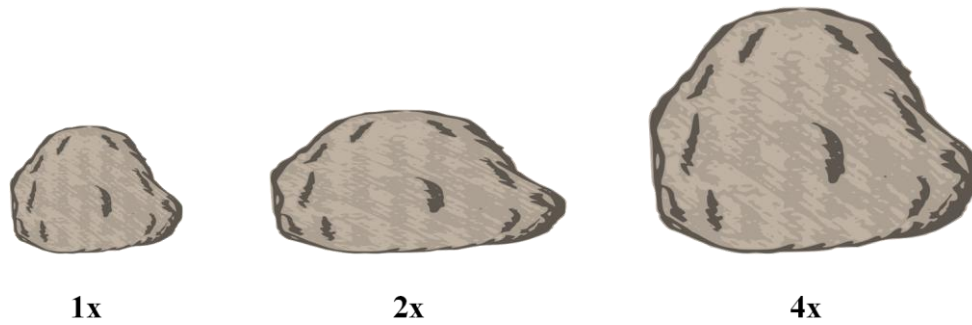


Abbildung 4: Relative Schätzung
Quelle: In Anlehnung an (Rubin, 2014, S. 164)

Neben der agilen Anforderungsbeschreibung (siehe User Stories, Seite 7) gibt es auch eine agile Technik für die relative Aufwandsschätzung (siehe Planning Poker, Seite 9). Das Schätzmaß wird als Story Point bezeichnet und ist ein abstraktes Maß. Story Point deshalb, weil bei dieser Technik User Stories geschätzt werden. Je nach Umfang bzw. Komplexität der User Stories werden weniger oder eben mehr Story Points vergeben. Ist eine User Story aus Sicht der Schätzenden zweimal so groß wie eine Andere, so werden dieser auch zweimal so viele Story Points zugewiesen. Damit aber eine Schätzung durchgeführt werden kann, muss zu Beginn eine Referenz User Story bestimmt werden. Diese sollte keinen zu großen, aber auch keinen zu kleinen Umfang haben. Hat man sich auf eine Referenz geeinigt, weist man dieser eine gewisse Anzahl an Story Points zu. Da im Anschluss relative Schätzungen zu dieser Referenz stattfinden, kann der Ausgangswert beliebig gewählt werden. (Rubin, 2014, S. 165-167)

Planning Poker

Planning Poker ist eine Technik für eine strukturierte agile Aufwandsschätzung von User Stories mittels Story Points. Zuerst muss ein Schätzteam zusammengestellt werden. Im Idealfall besteht dieses aus dem zukünftigen Entwicklungsteam. Ebenfalls wird eine Schätzskala benötigt. Dies wird durch den minimalen und maximalen Wert, die einer User Story zugewiesen werden können, beschränkt. In der Praxis hat sich hier eine leicht abgewandelte Fibonacci-Folge bewährt: $\frac{1}{2}$, 1, 2, 3, 5, 8, 13, 20, 40, 100. Der Hintergedanke besteht darin, dass mit zunehmender Größe der Stories auch die Schätzabweichung zunimmt und daher der Abstand zwischen den Zahlen immer größer wird. Beträgt der Umfang einer User Story 2 oder 3 Story Points, kann dies noch relativ einfach abgeschätzt werden. Ob eine User Story 20 oder 21 Story Points hat, liefert jedoch kein seriöses Ergebnis. Bevor die Schätzung starten kann, werden Schätzkarten, die sich

aus den oben genannten Zahlen zusammensetzen, an jeden „Spieler“ ausgegeben. Der Projektleiter stellt anschließend die erste User Story vor und bittet alle Anwesenden gleichzeitig eine erste Schätzung abzugeben. Die Personen mit der niedrigsten und höchsten Schätzung werden dann vom Projektleiter zur Begründung ihrer Entscheidung aufgefordert. Danach startet eine neue SchätZRunde mit derselben User Story. Nun sollten die Schätzungen näher zusammenliegen, im Idealfall sogar gleich sein. Dieser Vorgang wird solange wiederholt, bis sich die Schätzungen auf ein akzeptables Maß angenähert haben. Sind User Stories jedoch schlecht beschrieben und es kommen Fragen auf, so können diese in der SchätZRunde sogleich diskutiert und besser definiert werden. (Preußig, 2015, S. 98-103)

Task Board

Ein Task Board visualisiert alle anstehenden Aufgaben sowie deren Bearbeitungsstand (siehe Abbildung 5). Meist wird das Board in 4 Spalten unterteilt. Sobald ein Task (Teilaufgabe einer Story) bearbeitet wird, wandert dieser von „ToDo“ in die Spalte „In Progress“. Wird ein Task beendet, kann dieser in „Done“ abgelegt werden. Grundsätzlich soll so ein Board Transparenz schaffen und für alle Teammitglieder motivierend sein. Es zeigt aber auch ein Gesamtbild des Arbeitsfortschritts und dient somit als Planungswerkzeug. (Timinger, 2017, S. 175)

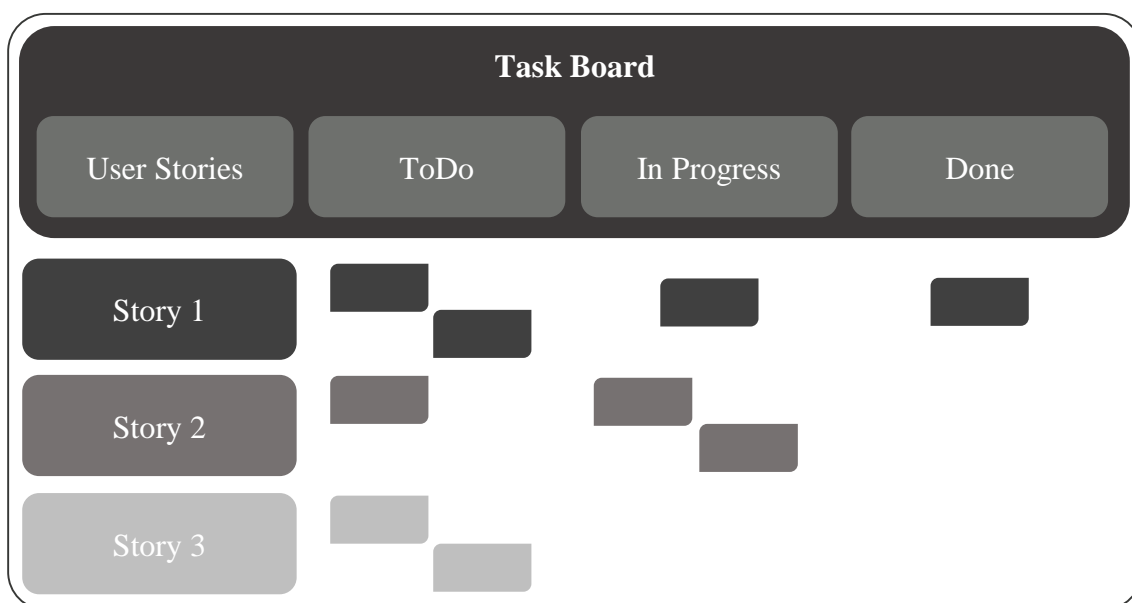


Abbildung 5: Task Board

Quelle: In Anlehnung an (Timinger, 2017, S. 175)

Definition of Done

Bevor ein Task nun aber von der Spalte „In Progress“ weiter nach „Done“ geschoben werden kann, müssen alle Kriterien für dessen Fertigstellung erfüllt sein. Diese Kriterien werden frühestens bei der Schätzung der Stories, jedoch spätestens wenn die Tasks mit auf das Task Board genommen werden, vom gesamten Team definiert und als „Definition of Done“ bezeichnet. Auch bei dieser agilen Technik soll wie schon bei der Aufwandsschätzung ein Konsens des gesamten Teams gefunden werden, um Reibungsverluste, aber auch Produktfehler zu vermeiden. (Preußig, 2015, S. 152-153)

Work-in-Progress-Limits

In einem Projekt ist es üblich, dass eine Person mehrere Tasks gleichzeitig bearbeitet. Da es zu immer wieder zu Stillständen verschiedener Aufgaben kommen kann, ist dies natürlich der richtige Ansatz, um den Arbeitsfluss nicht zu unterbrechen und die Produktivität zu steigern. Werden jedoch zu viele Aufgaben parallel abgearbeitet, kommt es zu einem Komplexitätsanstieg der Tasks, was wiederum zu einer abflachenden Produktivität führt. Dieser Zusammenhang kann mit dem sogenannten Work-in-Progress-Limit, kurz WIP-Limit, dargestellt werden (siehe Abbildung 6). Jedes Team, bzw. jede Person hat ein individuelles WIP-Limit. Dieses Limit muss somit für jedes Projekt und jede Person neu definiert werden. Ein Task Board, in welchem die Anzahl der Tasks in der Spalte „In Progress“ mit dem WIP-Limit begrenzt werden, liefert hierfür eine gute Basis. (Preußig, 2015, S. 87-91)

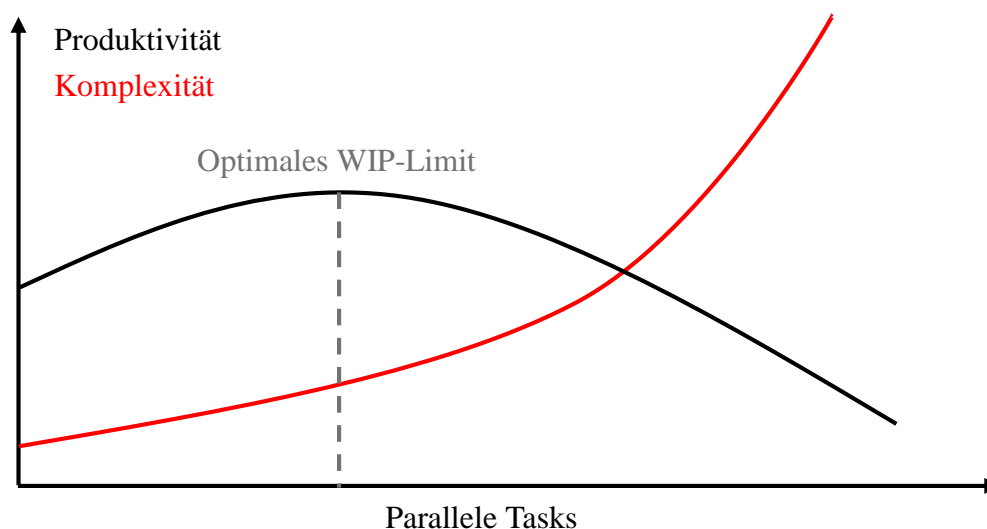


Abbildung 6: WIP-Limit

Quelle: In Anlehnung an (Preußig, 2015, S. 88)

Daily-Standup-Meetings

Eine ebenso wichtige agile Technik ist das „Daily-Standup-Meeting“. Dieses findet jeden Tag statt, es wird im Stehen abgehalten, um die Dynamik des Meetings zu erhöhen. Pro Meeting Teilnehmer sollte es ca. 2 Minuten, in Summe jedoch maximal 15 Minuten dauern. Das Meeting ist strikt durchgeplant und läuft folgendermaßen ab: Jeder Teilnehmer muss über seine geleistete Arbeit vom Vortag, die Aufgaben des Tages und mögliche Hindernisse berichten. Dieses Vorgehen soll dem Team dabei helfen, sich gegenseitig effektiv abzustimmen und zu informieren. (Preußig, 2015, S. 82-83)

Osmotische Kommunikation

Wie schon bei den agilen Werten (siehe 2.1.1) angeführt, besitzen Individuen und Interaktionen im agilen Projektmanagement einen besonderen Stellenwert. Die Kommunikation innerhalb des Teams ist deshalb ein sehr wichtiger, wenn nicht der wichtigste Baustein für erfolgreiche agile Projekte. Alistair Cockburn definierte 2004 hierzu den aus dem Englischen übersetzten Begriff „Osmotische Kommunikation“. Cockburn nahm an, dass Informationen innerhalb eines Teams nicht aktiv unter den einzelnen Teammitgliedern ausgetauscht werden müssen, sondern diese sich nur in näherer Umgebung oder im selben Raum befinden müssen, um relevante Gesprächsinhalte zu teilen oder zu erhalten. Es kommt also zu einem ständigen Informationsausgleich, ähnlich dem Teilchenaustausch bei der Osmose. Der Hintergedanke dieser agilen Technik liegt in der Erzeugung von Strukturen, die diese osmotische Kommunikation fördern. Dazu gehören gemeinsame Räumlichkeiten, Pausenbereiche, aber auch Chat-Tools, wenn die Unterbringung in einem gemeinsamen Büro nicht möglich ist. (Cockburn, 2004)

Timeboxing

Als Timeboxing bezeichnet man die strikte Einhaltung von Terminen und Plänen. Es ist eine der grundlegendsten und wichtigsten Techniken im agilen Projektmanagement und hat einen maßgeblichen Einfluss auf die Effizienz von Prozessen. Es wird sowohl bei kleinen Meetings, als auch bei größeren Zeitplänen stets angewendet. All jene Themen, die nicht in der dafür vorhergesehenen Time-Box besprochen oder bearbeitet werden können, müssen verschoben oder gestrichen werden. Bekräftigt wird diese agile Technik durch das allgemein bekannte Pareto Prinzip. Dieses, nach dem italienischen Mathematiker und Ökonom Vilfredo Pareto benannte Prinzip besagt, dass 80 % der

Ergebnisse mit 20 % des Gesamtaufwands erreicht werden können. Dieses Phänomen ist nicht allgemein gültig, kann aber im Projekt- und Zeitmanagement sehr gut angewendet werden. Abbildung 7 soll diesen Zusammenhang veranschaulichen. Die für einen Projektschritt oder Meeting eingeplante Zeit wird mit hoher Wahrscheinlichkeit effektiver genutzt als jene, die über die Time-Box hinausgehen würde. Im Allgemeinen soll die Anwendung dieser agilen Technik dazu führen, dass die in einem Projekt nur begrenzt vorhandenen Zeitressourcen möglichst effektiv genutzt werden, indem man sich bei Meetings auf das Wesentliche konzentriert, bei der Aufgabenabarbeitung Prioritäten setzt und stets dafür sorgt, dass geplante Termine auch eingehalten werden. (Preußig, 2015, S. 106-109)

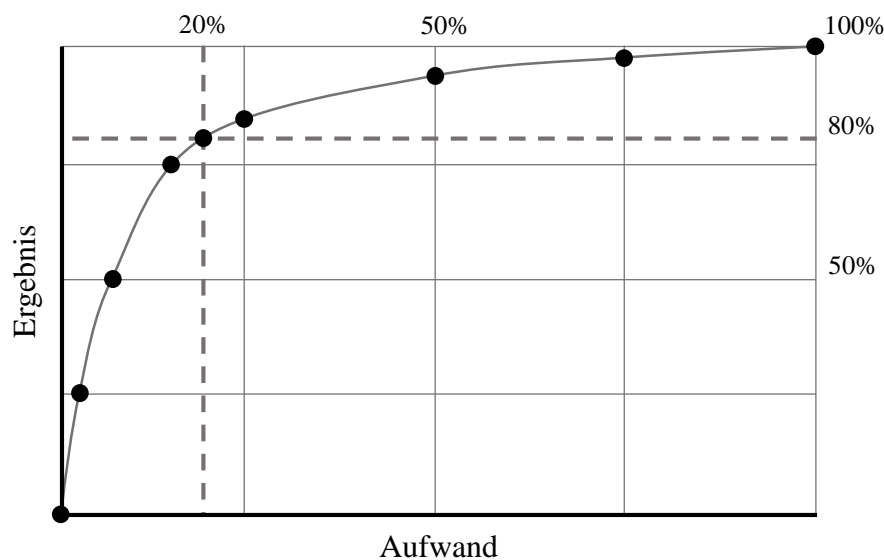


Abbildung 7: Pareto Prinzip

Quelle: In Anlehnung an (Fleig, 2016)

Burn-Down-Chart

Mittels einem Burn-Down-Chart lässt sich der Arbeitsfortschritt eines Projektes sehr einfach und übersichtlich visualisieren. Dabei wird auf der Abszisse der Projektverlauf, auf der Ordinate der Restaufwand (zum Beispiel Story Points) aufgetragen (siehe Abbildung 8). Mit fortschreitendem Projekt sollte demnach der Restaufwand abnehmen, bis er schließlich unter Annahme perfekter Planung am Ende des Projektes den Wert Null erreicht. In den wenigsten Fällen wird der Chart linear verlaufen, sondern um den idealen Burn Down schwanken. Im Falle, dass sich der reale vom idealen Burn Down mit zunehmenden Projektfortschritt negativ entfernt, wäre dies ein klares Zeichen, Gegenmaßnahmen einzuleiten. Dieser Chart hilft demnach den Fortschritt eines Projektes

besser zu überwachen, aber auch einen negativen Trend frühzeitig zu erkennen, um darauf reagieren zu können. (Preußig, 2015, S. 110)

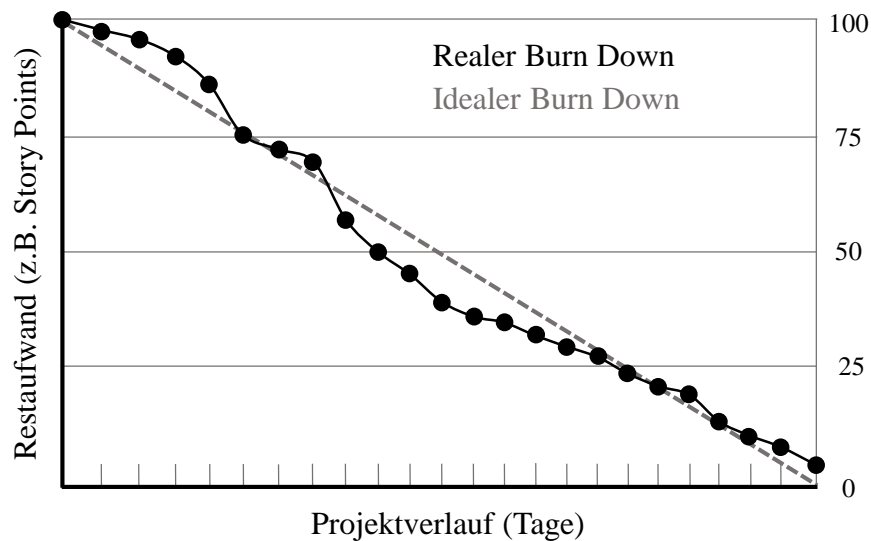


Abbildung 8: Burn-Down-Chart
Quelle: In Anlehnung an (Preußig, 2015, S. 110)

Velocity

Unter der Velocity versteht man die Entwicklungsgeschwindigkeit eines Teams. Die Einheit der Velocity kann grundsätzlich selbst festgelegt werden, in der Praxis werden meist Story Points dafür verwendet. Die in einem vorher definierten Zeitraum (bei Scrum wären das die Sprints) abgearbeiteten Story Points definieren somit die Velocity. Liegt ein Burn-Down-Chart vor, könnte man damit sehr einfach die Velocity berechnen und darstellen. Abbildung 9 zeigt diesen Zusammenhang. Die schwarze Punktlinie definiert hierbei den Restaufwand in Story Points. Je weiter das Projekt fortgeschritten ist, desto weniger Story Points müssen abgearbeitet werden. Im ersten Teilabschnitt des Projektes, hier Sprint 1 bezeichnet, hat das Team 8 Story Points umgesetzt. In Sprint 2 waren es dann schon knapp unter 20. Die Balken stellen hier also eine Momentaufnahme der Velocity dar. Betrachtet man diese über einen längeren Zeitraum, also über mehrere Sprints, kann eine durchschnittliche Entwicklungsgeschwindigkeit gefunden werden. Diese wird sehr häufig als Unterstützung für die Sprintplanung herangezogen, dient aber auch als Grundlage für die Planung neuer Projekte.

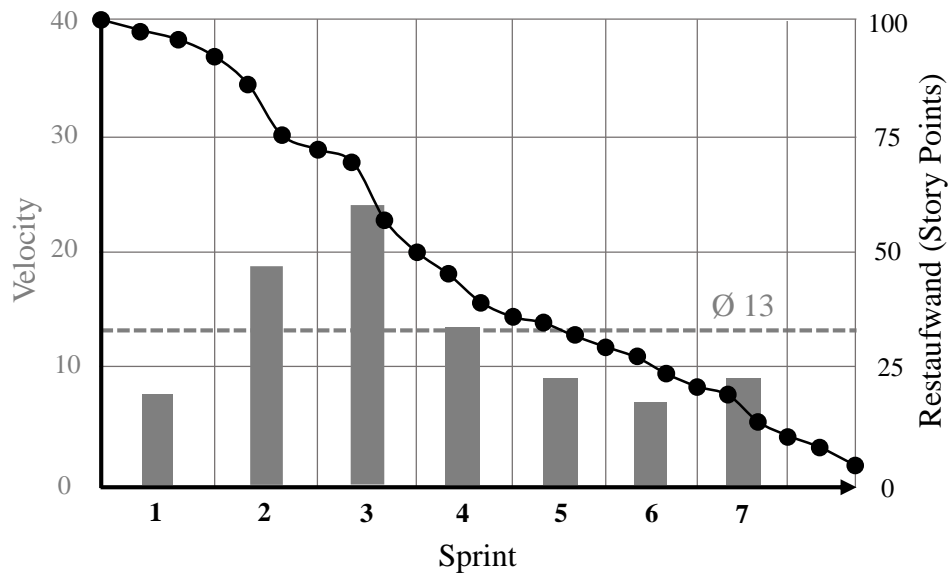


Abbildung 9: Velocity

Quelle: Eigene Darstellung

2.1.4 Agile Methoden

Wie in Abbildung 1 schon gezeigt, bilden die agilen Methoden den Kern des agilen Mindset. Sie werden von den agilen Techniken, Prinzipien und Werten umgeben. Das Ziel agiler Methoden besteht darin, Projekten eine Struktur zu geben und hierbei auf die oben genannten Techniken, Prinzipien und Werte zurückzugreifen. (Preußig, 2015, S. 132)

Es gibt heutzutage viele Methoden, die im Bereich des agilen Projektmanagements eingesetzt werden können, wobei der Großteil dieser Methoden aus der Softwareentwicklung stammt und hauptsächlich auch in diesem Gebiet angewendet wird. Außerdem wurde aber auch eine Methode namens Scrum entwickelt, die aufgrund der allgemeinen Beschreibung ebenfalls in anderen Bereichen seine Anwendung findet. Daher wird dieses Vorgehensmodell in der Praxis am häufigsten angewendet. Das folgende Kapitel zeigt neben den Grundlagen von Scrum, wie oben genannte agile Techniken in einem Scrum Prozess optimal integriert werden.

2.2 Scrum – Grundlagen

Die erste namentliche Nennung von Scrum erfolgt in einem Artikel von Hirotaka Takeuchi und Ikujiro Nonaka im Harvard Business Review aus dem Jahre 1989 mit dem Titel „*The New New Product Development Game*“ (Takeuchi & Nonaka, 1989). Diese untersuchten Unternehmen wie Honda, Canon und Fuji-Xerox und stellten fest, dass sich bei der Entwicklung neuer, komplexer Produkte hervorragende Ergebnisse mittels kleinen selbstorganisierten Teams sowie einer ganzheitlichen Produktentwicklung erzielen ließen. Den Begriff Scrum, aus dem Englischen übersetzt „Gedränge“ und aus dem Rugbysport kommend, verwendeten Takeuchi und Nonaka, um das Teamwork besser zu veranschaulichen. Inspiriert von diesen Ansätzen, entwickelte Ken Schwaber 1993 einen Scrum Prozess, um diesen bei der Software-Entwicklung einsetzen zu können. 1995 veröffentlichte er seinen ersten Artikel über Scrum auf der OOPSLA-Konferenz (Object-Oriented Programming, Systems, Languages, and Applications). (Rubin, 2014, S. 35-36)

Scrum wird heute oft als der Standard in der agilen Soft- und Hardwareentwicklung angesehen. Da es aber eine sehr allgemeine Beschreibung aufweist und keine konkreten agilen Techniken fordert, sondern nur den Projektrahmen vorgibt, wie zum Beispiel Projektrollen und Prozessablauf, handelt es sich bei Scrum eher um ein Rahmenwerk (Framework) für agiles Vorgehen, innerhalb dessen die Soft- oder Hardwareentwicklung stattfindet. Scrum definiert dazu verschiedene Rollen, Ereignisse und Artefakte (siehe Abbildung 10). (Opelt, et al., 2018, S. 3ff)

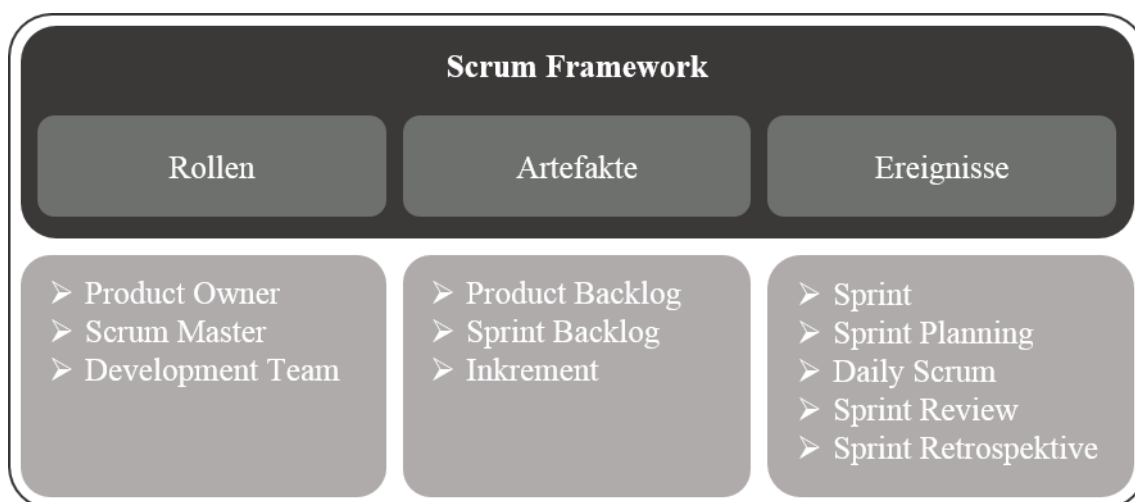


Abbildung 10: Scrum Framework

Quelle: In Anlehnung an (Opelt, et al., 2018, S. 3ff)

Das gesamte Framework beruht auf einem empirischen, iterativen und inkrementellen Ansatz. Das Produkt wird hier in wiederkehrenden (iterativen) Entwicklungszyklen, den sogenannten Sprints, entwickelt. Die nach den Sprints entstehenden Zwischenergebnisse bauen dabei aufeinander auf (inkrementell) und wachsen stetig zum fertigen Produkt. Dabei greift Scrum auf die Theorie der „empirischen Prozesssteuerung“ zurück. Diese beruht auf den 3 Säulen der Transparenz, Überprüfung (Inspect) und Anpassung (Adapt). Dabei müssen Fortschritte aber auch Hürden stets mit höchster Transparenz innerhalb aller Stakeholder kommuniziert und das Produkt und der Prozess kontinuierlich überprüft und angepasst werden. Um diese Zusammenhänge sowie die chronologische Abfolge des Scrum Prozesses verstehen zu können, gibt Abbildung 11 einen kurzen Überblick oben genannter Scrum Aktivitäten. Wie sich das Framework nun im Detail zusammensetzt und welche agilen Techniken für einen erfolgreichen Scrum Prozess nicht fehlen dürfen, wird in den folgenden Kapiteln erklärt. (Opelt, et al., 2018, S. 3ff)

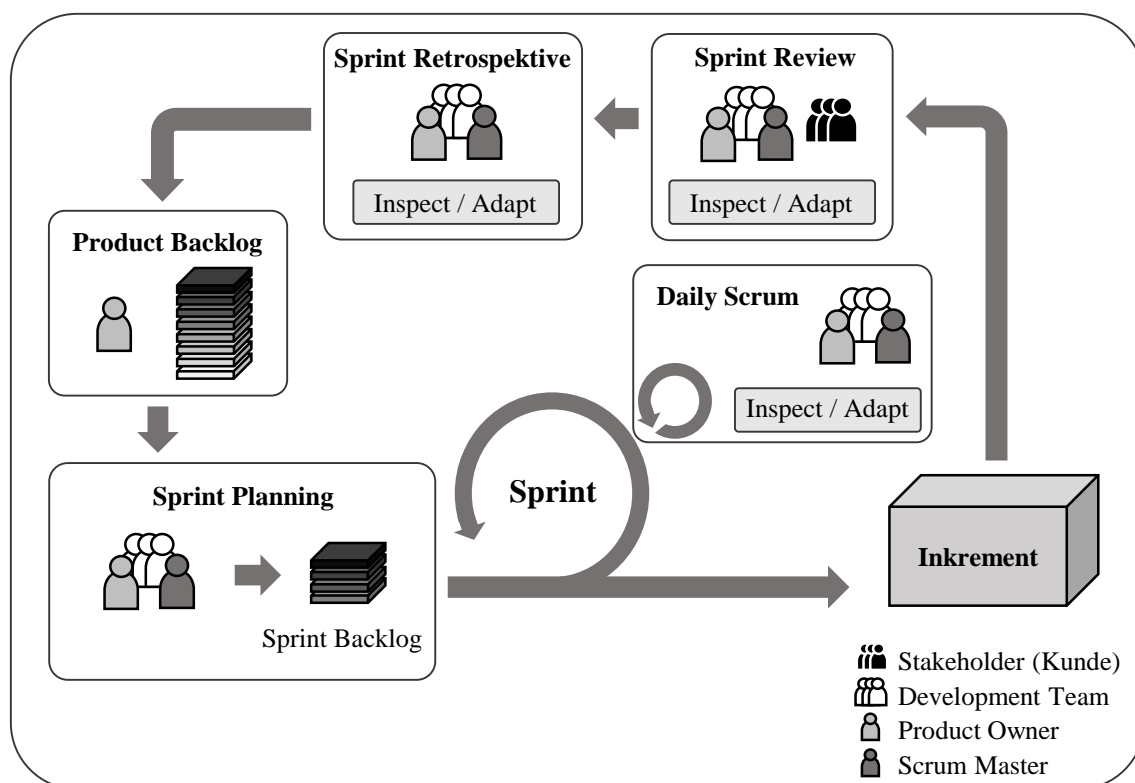


Abbildung 11: Vereinfachter Scrum Prozess
Quelle: In Anlehnung an (Rubin, 2014, S. 51)

2.2.1 Rollen

Innerhalb eines Scrum Teams erfolgt die Aufgabenverteilung auf drei Rollen. Diese sind der Product Owner, der Scrum Master und das Development Team. Eine Person darf maximal eine Rolle einnehmen, um Interessenskonflikte zu vermeiden. Die Rolle des Scrum Masters und des Product Owners weisen im Idealfall keinen Stellvertreter auf und sollten nicht doppelt besetzt sein. Die Frage nach der optimalen Größe eines Scrum Teams kann nicht genau beantwortet werden, es sollten jedoch nicht weniger als fünf, aber auch nicht mehr als elf Personen im Team sein. (Heimrath, 2018, S. 25ff)

Product Owner

Der Product Owner kann am ehesten mit einem Projektmanager im klassischen Projektmanagement verglichen werden. Einer seiner Hauptaufgaben ist die Pflege und Priorisierung der zu entwickelnden Features (User Stories). Dies bedarf einer engen Kommunikation mit dem Development Team und allen involvierten Stakeholdern. Er wird deshalb auch oft als Schnittstelle zwischen diesen beiden Parteien angesehen. Zudem trägt der Product Owner die Verantwortung für das Projektergebnis, in technischer, als auch ökonomischer Sicht. Er muss dafür sorgen, dass das Team einerseits termingerecht abgeliefert und dies andererseits auch mit den geplanten Ressourcen möglich ist. (Heimrath, 2018, S. 25ff)

Scrum Master

Ein entscheidendes Kriterium im Scrum Framework ist die Einhaltung der Scrum Prozesse. Dafür gibt es einen Scrum Master. Dieser überwacht alle mit Scrum in Verbindung stehenden Prozesse und ist für die Lösung intern wie auch extern verursachter Probleme verantwortlich. Auch hier bedarf es einer guten Kommunikation mit allen Teammitgliedern. Neben diesen Aufgaben dient er als Trainer für das Team und muss als Ansprechperson für Fragen in Bezug auf Scrum jederzeit verfügbar sein. (Heimrath, 2018, S. 25ff)

Development Team

Das Development Team besteht im Normalfall aus ca. 3-9 Personen und ist für die Entwicklung des Produktes verantwortlich. Das Team sollte möglichst cross-funktional aufgebaut sein, um alle für die Erreichung des Projektziels notwendigen Kompetenzen zu

vereinen. Dabei haben sich in der Praxis vor allem generalisierte Spezialisten bewährt, also Personen, die ein Spezialgebiet vorweisen können, aber dennoch ein großes Breitenwissen besitzen. Genau dieses Breitenwissen ermöglicht es ihnen, sich hervorragend in Teams einzuordnen, da sie andere Disziplinen für die Lösung von Aufgaben erkennen und auch akzeptieren. Ein besonders wichtiges Merkmal des Development Teams ist die Selbstorganisation und die daraus resultierende vollkommen autonome Aufgabenverteilung. Wie das vom Product Owner gesteckte Ziel erreicht wird, bestimmt demnach das Development Team selbst. Ein Einfluss von außen muss vom Scrum Master möglichst verhindert werden. (Heimrath, 2018, S. 33-35)

2.2.2 Artefakte

Neben den drei Rollen definiert Scrum drei Artefakte (siehe Abbildung 10). Das Product Backlog, Sprint Backlog und das Inkrement. Diese helfen dem Scrum Team die Übersicht zu behalten und dienen zugleich als Indikator für den Arbeitsfortschritt.

Product Backlog

Unter dem Product Backlog versteht man die Gesamtheit aller Anforderungen an das zu entwickelnde Produkt. Es handelt sich hier also um eine Art Liste von Product Backlog Items (PBIs), meist User Stories (siehe 2.1.3) genannt. Diese wird jedoch nicht wie bei einem traditionellen Lastenheft als vollständig angesehen, sondern lässt sich aufgrund verschiedenster Einflüsse (ändernde Marktbedingungen oder Kundenwünsche) dynamisch anpassen. Dabei ist es wichtig, dass die Items im Backlog nach Geschäftswert priorisiert werden, da wie bei allen agilen Methoden stets die wichtigsten und nützlichsten Aufgaben zuerst bearbeitet werden sollen. Es müssen zu Beginn jedoch nicht alle PBIs eine Priorisierung erhalten, sondern nur diejenigen, die zuerst abgearbeitet werden. Die gesamte Pflege (Priorisierung, Änderungen, usw.) des Backlogs obliegt, wie schon oben angeführt, dem Product Owner. (Rubin, 2014, S. 52-54)

Sprint Backlog

Da Scrum ein iteratives und inkrementelle Vorgehen aufweist, wird das Gesamtprodukt in kleinen Teilschritten, den sogenannten Sprints, entwickelt (siehe Abbildung 12). Aus allen Anforderungen (PBIs) im Product Backlog müssen also Items in das Sprint Backlog für den anstehenden Sprint übertragen werden. Die Anzahl der zu transferierenden

Einträge hängt von deren Aufwand sowie der Sprintlänge ab. Das Sprint Backlog ist also wie das Product Backlog eine Ansammlung von Anforderungen, jedoch nur von so vielen, wie auch tatsächlich in einem Sprint vom Development Team bewältigt werden können. (Rubin, 2014, S. 54-56)

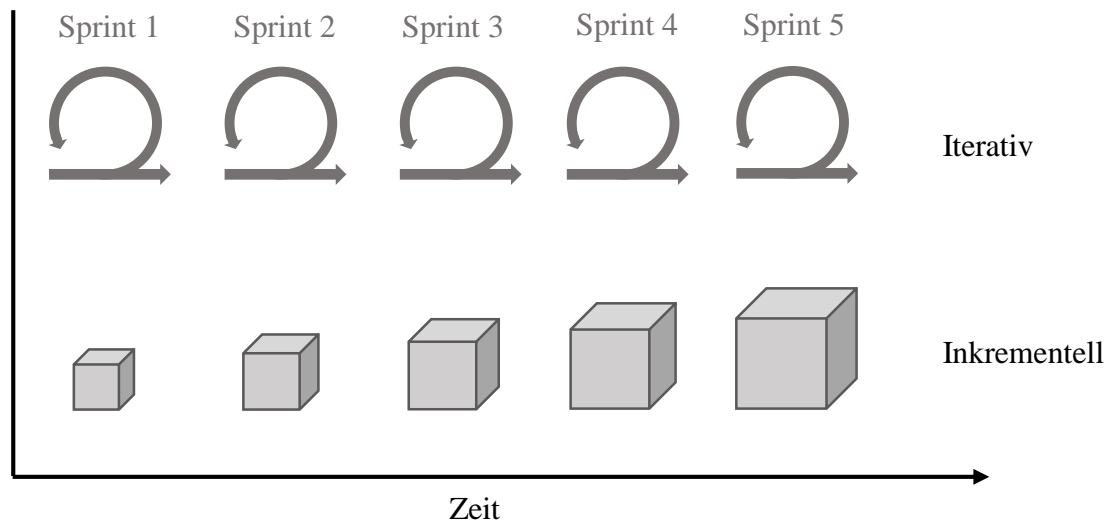


Abbildung 12: Iterativ und Inkrementell
Quelle: Eigene Darstellung

Inkrement

Das dritte Artefakt heißt Inkrement und beschreibt das Ergebnis eines Sprints, es entspricht also einem Teilprodukt. Bei jedem weiteren Sprint entsteht auch ein weiteres Inkrement, welches auf dem vorherigen Inkrement aufbaut (siehe Abbildung 12). Dabei ist darauf zu achten, dass jedes fertige Inkrement einen auslieferungsfähigen, beziehungsweise einsatzfähigen Zustand aufweisen muss. (Preußig, 2015, S. 137)

2.2.3 Ereignisse

Unter den Scrum Ereignissen versteht man die wiederkehrenden Iterationen, also die sogenannten Sprints und verschiedene dazu in Bezug stehende Scrum Meetings (siehe Abbildung 10).

Sprint

Wie schon oben angeführt, erfolgt die Entwicklung bei Scrum in Sprints, die im Normalfall zwischen 2 und 4 Wochen dauern. Die Anzahl der benötigten Sprints hängt vom Gesamtaufwand des Projektes ab. Zudem kann jeder Sprint als Timebox angesehen

werden und weist deshalb eine konstante Länge auf, die während des Projektes im Idealfall nicht geändert wird (siehe Abbildung 13). Eine der wichtigsten Einschränkungen betrifft Änderungen in Bezug auf Umfang und Personal. Die Aufnahme von weiteren Anforderungen oder personelle Umstrukturierungen sollten immer erst nach Abschluss eines Sprints durchgeführt werden. (Rubin, 2014, S. 54-56)

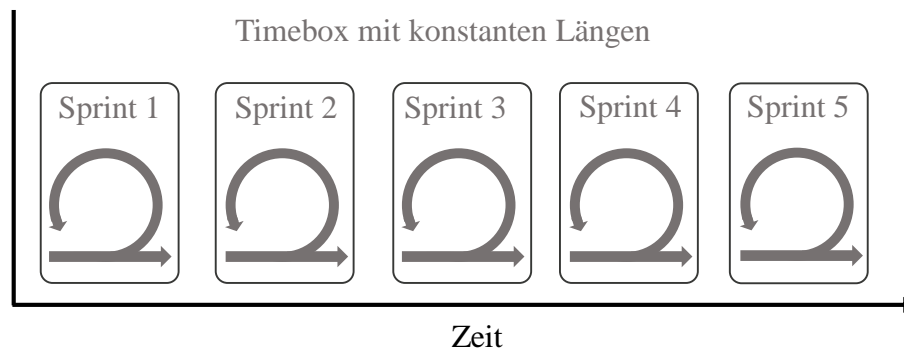


Abbildung 13: Ein Sprint als Timebox
Quelle: In Anlehnung an (Rubin, 2014, S. 54)

Sprint Planning

Bevor ein Sprint gestartet werden kann, müssen im Sprint Backlog alle für den Sprint vorgesehenen Anforderungen definiert werden. Dazu wird das Sprint Planning Meeting benötigt, bei dem sich der Product Owner, der Scrum Master und das Development Team treffen. Wie alle Meetings in Scrum, unterliegt auch das Sprint Planning Meeting dem strikten Timeboxing. Bei einer Sprintlänge von zwei Wochen sollten hierfür ca. vier Stunden eingeplant werden. Zu Beginn des Meetings wird gemeinsam ein Sprint-Ziel festgelegt. Basierend auf dieser Grundlage werden Items aus dem Product Backlog entnommen und das Sprint Backlog damit befüllt. Im Normalfall handelt es sich hier um die am höchsten priorisierten Einträge. Die Verantwortung, wie viele Items in dem anstehenden Sprint abgearbeitet werden können und demnach in das Backlog transferiert werden, obliegt einzig und alleine dem Development Team. (Rubin, 2014, S. 55-56)

Daily Scrum

Das Daily Scrum, auch als Daily-Standup-Meeting bezeichnet, ist ein tägliches, meist 15 Minuten langes Meeting, das im Stehen abgehalten wird und zum gegenseitigen Informationsaustausch des gesamten Scrum Teams dient. Da es schon im Kapitel 2.1.3 (Agile Techniken) angeführt wurde, wird es hier nicht weiter beschrieben.

Sprint Review

Eine gute Zusammenarbeit mit dem Kunden sowie ein frühes Feedback zum Teilprodukt sind wichtige Bestandteile der agilen Werte und Prinzipien und werden deshalb im Sprint Review berücksichtigt. Es ist die vorletzte Aktivität in einem Sprint und dient zur Überprüfung und Anpassung des entwickelten Teilproduktes. Dabei treffen sich alle Stakeholder sowie das gesamte Scrum Team. Bei einem zweiwöchentlichen Sprint sollten ca. zwei Stunden für das Review angesetzt werden. Das Development Team stellt dabei allen involvierten Parteien das entwickelte Inkrement vor. Diese haben nun die Möglichkeit mit ihrem Input die weitere Entwicklung und damit das zukünftige Produkt zu beeinflussen. Das Sprint Review ist somit die perfekte Plattform, um wertvolle Kritik, Anregungen und Änderungswünsche einzubringen. In einer konstruktiven Diskussion zwischen dem Scrum Team und den Stakeholdern kann nun die Richtung für den nächsten Sprint definiert werden. Der Wunsch nach neuen, oder die Anpassung bestehender Anforderungen kann sogleich vom Product Owner im Product Backlog vorgenommen werden. (Rubin, 2014, S. 60)

Sprint Retrospektive

Die letzte Aktivität in einem Sprint ist die Sprint Retrospektive. Diese findet meist nach dem Review und vor dem nächsten Sprint Planning statt und dient ebenfalls zur Überprüfung und Anpassung, jedoch nicht auf Produkt- sondern auf Prozessebene. Am Meeting beteiligen sich der Scrum Master, der Product Owner und das Development Team. Das Ziel ist eine stetige Verbesserung des Scrum Prozesses sowie der angewendeten Techniken. Es handelt sich hier also um eine Art Lessons Learned, wobei die notwendigen Aktionen für effizientere Prozesse und bessere Techniken gleich im nächsten Sprint umgesetzt werden. Ist die Sprint Retrospektive abgeschlossen, kann sogleich mit dem nächsten Sprint, also dem Sprint Planning gestartet werden. (Rubin, 2014, S. 61)

2.3 Scrum – Erweiterungen

Neben dem allgemeinen Prozess, den Rollen, Artefakten und Ereignissen gibt Scrum keine weiteren Aktivitäten vor. Die in Kapitel 2.1.3 angeführten Techniken können somit frei eingebunden werden. Dies ist auch notwendig, da ohne eine weitere Konkretisierung der Rollen, Artefakte und Ereignisse, das Scrum-Framework in seiner reinen Form nicht angewendet werden kann. In diesem Kapitel sollen deshalb all jene Aktivitäten beziehungsweise agilen Techniken angeführt werden, die sich in der Praxis bewährt haben und eine gute Erweiterung für den allgemeinen Scrum Prozess darstellen. Der Fokus wird hier auf die Projektanforderungen, Aufwandsabschätzung und Projektplanung gelegt.

2.3.1 Projektanforderungen

Im klassische Projektmanagement kommuniziert der Kunde seine gewünschten Anforderungen im Allgemeinen über ein Lastenheft. Diese sind je nach Produktkomplexität und Größe des Projektes entweder sehr umfangreich und detailliert beschrieben, oder eben sehr grobkörnig. In beiden Fällen sind sie aber fixiert und können nach Projektstart nicht mehr ohne Mehraufwand geändert werden. Bei agilen Projekten hingegen, muss eine gewisse Flexibilität der Anforderungen gewährleistet werden. Zudem werden die Inhalte des Lastenhefts oft vom Entwickler anders verstanden, als vom Kunden vorgesehen. In der Praxis haben sich hier die agilen Techniken der User Stories, Epics und Tasks bewährt (siehe Seite 7). Diese helfen einerseits durch ihre gute Strukturierung einen Überblick zu behalten und andererseits, ein besseres Verständnis dafür zu entwickeln, was der Kunde wirklich benötigt. Eine weitere sehr hilfreiche agile Technik ist die „Definition of Done“ (siehe Seite 11). Diese nennt alle Akzeptanzkriterien für eine Anforderung und hilft somit, Missverständnisse bei allen Beteiligten in Bezug auf die Fertigstellung zu reduzieren.

2.3.2 Aufwandsabschätzung

Scrum besitzt eine Arte Liste, die alle vom Kunden geforderten Anforderungen enthält, das sogenannte Product Backlog. Über ein Sprint Planning Meeting wird zu Beginn des Sprints das Sprint Backlog erzeugt. Dabei handelt es sich um eine Auswahl von PBIs für den anstehenden Sprint. Damit diese Auswahl aber getroffen werden kann, muss der

Aufwand für die unterschiedlichen Anforderungen (auch User Stories genannt) zuerst geschätzt werden. Man benötigt also ein Schätzverfahren und ein Schätzmaß. Für diese Aktivität nennt Scrum aber keine unterstützende Technik. Sehr viele Unternehmen, die agil arbeiten, verwenden als Schätzmaß „Story Points“ (siehe Seite 8) und als Schätzverfahren „Planning Poker“ (siehe Seite 9). Damit lassen sich die Vorteile des relativen Schätzens und der konsensbasierten Expertenmeinungen zusammenführen. Bei sehr großen Teams oder umfangreichen Backlogs, stößt dieses Verfahren aber an seine Grenzen. Neben dem Planning Poker erfreut sich deshalb ein weiteres Schätzverfahren namens „Magic Estimation“ immer größerer Beliebtheit. Diese von Boris Gloger (Opelt, et al., 2018) entwickelte Methode ermöglicht einerseits das Schätzen von sehr großen Backlogs und liefert andererseits auch noch schnellere und genauere Schätzungen als Planning Poker. Aus diesen Gründen soll es im Folgenden kurz dargestellt werden. (Opelt, et al., 2018, S. 55-56)

Magic Estimation

Der Product Owner platziert auf einem großen Tisch, am Boden, oder einer Pinnwand Karten der angepassten Fibonacci-Folge (1, 2, 3, 5, 8, 13, 20, 40, 100) in ausreichend großem Abstand zueinander. Zusätzlich muss der Product Owner Karten (empfohlen wird DIN A4) mit den User Stories erstellen, die er anschließend gleichmäßig auf die Teammitglieder verteilt. Wichtig ist, dass die Karten den Wert der Priorisierung enthalten und möglichst groß und deutlich geschrieben sind, dass die Information auch bei größerer Entfernung einfach gelesen werden kann. Das Schätzverfahren kann nun starten. Jedes Mitglied schaut sich seine Stories an und platziert sie unter dem Wert der Fibonacci-Folge, der seiner Meinung nach den Aufwand widerspiegelt. Haben alle Spieler ihre User Stories abgelegt, startet die zweite Runde. In dieser hat jeder Spieler Zugriff auf alle Stories. Das bedeutet, wenn jemand mit der Positionierung einer Story nicht zufrieden ist, kann dieser sie einfach verschieben. Dazu muss der Spieler nur ein kurzes Statement abgeben, warum er diese verschieben will. In dieser Runde ist es überaus wichtig, dass der Product Owner alles sehr genau beobachtet. Wenn eine Karte oft springt, liegt eine Meinungsverschiedenheit vor, die eine weitere Klärung benötigt. Man kann diese Karte aus dem Spiel nehmen, oder auch einfach markieren. Das Spiel endet, sobald es zu keinen weiteren Umpositionierungen kommt. Während des gesamten Spiels müssen drei Prinzipien befolgt werden. Zwischen den Spielern ist keine Kommunikation erlaubt, jeder Spieler darf jede User Story so oft er will bewegen und jede User Story, die entweder zu

groß ist oder nicht verstanden wird, kommt auf den Wert 100. Das Ergebnis des konsensbasierten Verfahrens Magic Estimation, sind wie auch beim Planning Poker, geschätzte User Stories, jedoch in kürzer Zeit und mit höherer Genauigkeit. (Opelt, et al., 2018, S. 56)

2.3.3 Projektplanung

Neben den agilen Techniken zur besseren Definition von Anforderungen (siehe Kapitel 2.3.1) und deren Aufwandsabschätzung (siehe Kapitel 2.3.2) werden auch solche für die gesamte Projektplanung, also auf Release-Ebene (langfristig) und Sprint-Ebene (kurzfristig) benötigt. Scrum nennt hierfür keine unterstützenden Aktivitäten. Mit der Einführung von User Stories, Story Points und Schätzverfahren wie Planning Poker oder Magic Estimation wurde schon ein guter Grundstein gelegt. Fragen wie, in welchem Zeitrahmen kann man das Projekt abschließen, wie viele User Stories können pro Sprint abgearbeitet werden oder wie kann man den Projektfortschritt effektiv tracken, können damit aber nicht beantwortet werden. Eine der wichtigsten Techniken, um die oben genannten Fragen zu beantworten, ist die Team-Velocity (siehe Seite 14). Kennt man die durchschnittliche Entwicklungsgeschwindigkeit des Teams und liegt ein priorisiertes, geschätztes Product Backlog vor, so kann sehr einfach eine erste Abschätzung in Bezug auf die Projektdauer (Release-Ebene) abgegeben werden. (Rubin, 2014, S. 297ff)

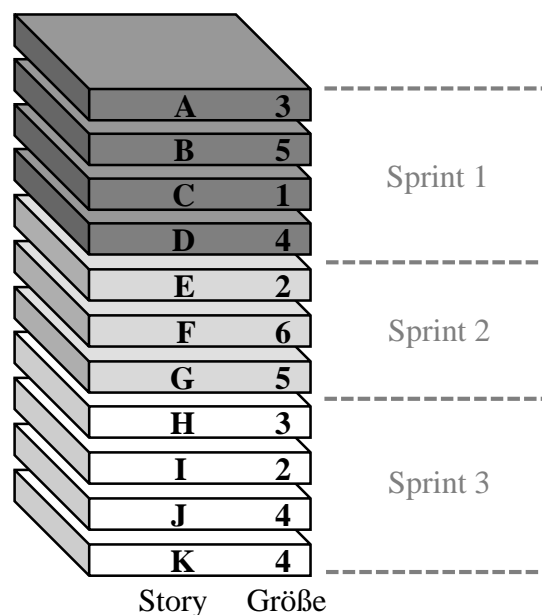


Abbildung 14: Priorisiertes und geschätztes Product Backlog
Quelle: In Anlehnung an (Rubin, 2014, S. 360)

Abbildung 14 zeigt ein priorisiertes (A-K) und geschätztes Product Backlog. Dieses hat 11 User Stories mit einem Gesamtaufwand von 39 Story Points. Liegt die durchschnittliche Velocity des Teams bei 13 Story Points pro Sprint, werden zumindest drei Sprints benötigt. Dauert ein Sprint zwei Wochen, kann der Gesamtaufwand mit sechs Wochen angegeben werden. Auch ist es sehr einfach eine grobe Zuordnung der einzelnen Elemente vorzunehmen. Jene Stories mit der höchsten Priorisierung werden Sprint 1 zugeordnet bis 13 Story Points erreicht werden. Genauso verfährt man auch mit Sprint 2 und Sprint 3. Liegen keine historischen Daten zur Team Velocity vor, muss diese vor Sprint 1 abgeschätzt und im Laufe des Projekts kontinuierlich angepasst werden. Um den Projektfortschritt zu überwachen und einen möglichen stagnierenden Arbeitsfortschritt frühzeitig zu erkennen, würde sich ein Burn-Down-Chart (siehe Seite 13) hervorragend eignen. Auch kann von diesem die Team-Velocity mit wenig Aufwand tagesaktuell abgelesen werden.

2.4 Scrum – Prozessmodell

Wie nun diese angeführten Erweiterung (siehe Kapitel 2.3) in einem detaillierten Scrum-Prozess Anwendung finden, wird im Folgenden erklärt und visualisiert. Dazu wird der Ausgangsprozess (Abbildung 11) um die in Kapitel 2.3 angeführten agilen Techniken erweitert (siehe Abbildung 15).

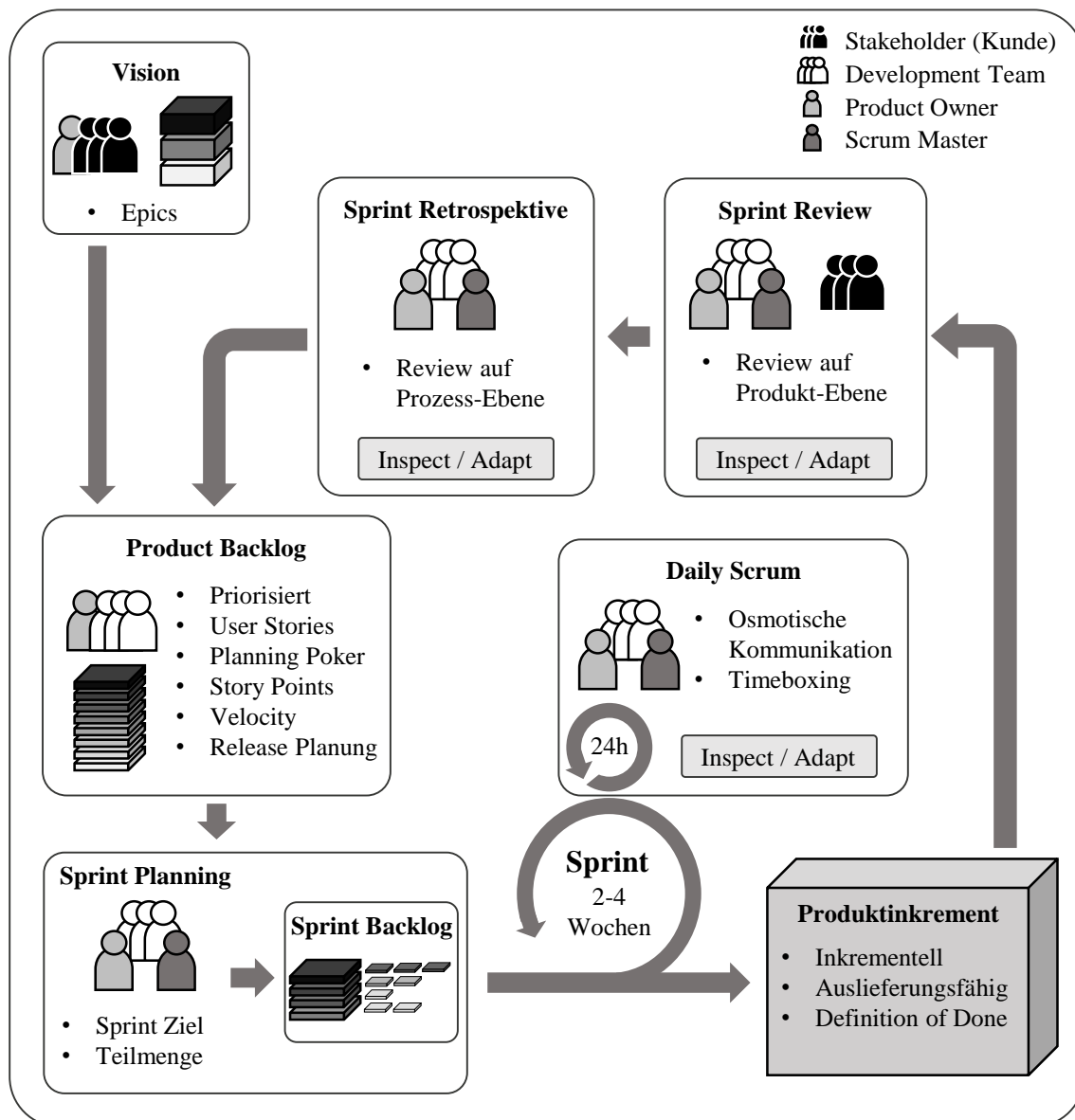


Abbildung 15: Erweiterter Scrum Prozess
Quelle: In Anlehnung an (Rubin, 2014, S. 51)

Vision

An erster Stelle steht die Vision des Kunden. In dieser wird das gewünschte Produkt kurz beschrieben und jene Bereiche hervorgehoben, die einen Mehrwert für alle Stakeholder

darstellen. Im Zuge der Visionsfindung müssen dann die Produktdetails in Form von Epics (Seite 8) dargestellt werden. Diese Aufgabe übernimmt der Product Owner, idealerweise in Kooperation mit dem Scrum Master und dem Development Team. Wird das Projekt mit einem Entwicklungsdienstleister durchgeführt, würde dieser den Scrum Master und das Development Team stellen, der Kunde im Idealfall den Product Owner. Je nach Wunsch des Kunden kann die Rollenverteilung aber auch eine abweichende Struktur aufweisen. Vorstellbar wäre auch ein Product Owner auf Seiten des Entwicklungsdienstleisters, der als Proxy des Kunden Product Owners fungiert. (Rubin, 2014, S. 329-348)

Product Backlog

Im nächsten Schritt muss das erste Product Backlog angelegt werden. Dieses basiert auf der Produktvision und beinhaltet alle Kundenanforderungen in Form von Epics und User Stories (Seite 8). Zu Beginn müssen die Epics Schritt für Schritt vom Product Owner und dem Development Team in User Stories übergeführt werden. Anschließend werden alle gefundenen Stories vom Development Team mittels Planning Poker (Seite 9) oder Magic Estimation (Seite 24) geschätzt. Idealerweise wird dazu ein relatives Größenmaß wie Story Points (Seite 8) verwendet. Anschließend führt der Product Owner eine Priorisierung aller Product Backlog Items nach Geschäftswert durch. Das Ergebnis ist eine nach Aufwand geschätzte und nach Geschäftswert priorisierte Liste an Kundenanforderungen, in Form von User Stories. Wenn vom Kunden gewünscht, kann an dieser Stelle auch eine erste Schätzung der Gesamtprojektdauer (Release Planung) abgegeben werden (siehe auch Seite 25). Das nun vorliegende Backlog dient als Startpunkt für den eigentlichen Scrum Prozess und wird nach jeder Iteration neu bewertet und gegebenenfalls angepasst. (Rubin, 2014, S. 137-155; Heimrath, 2018, S. 37-50)

Sprint Planning

Da ein Backlog im Normalfall mehrere Wochen oder gar Monate an Arbeit repräsentiert, wird im nächsten Schritt das sogenannte Sprint Planning durchgeführt. In diesem Meeting ist das gesamte Scrum Team vertreten. Der Product Owner und das Development Team einigen sich dabei auf ein Sprint Ziel. Auf Basis des Ziels, der Team Velocity (falls bekannt) und natürlich der Sprint Länge (2-4 Wochen), erstellen sie zusammen das Sprint Backlog. Dieses beinhaltet all jene User Stories, die in diesem Sprint umgesetzt werden sollen. Ob die hier enthaltenen Stories in einem nächsten Schritt weiter in kleinere Tasks

zerlegt werden, hängt von deren Ausgangsgröße und den individuellen Wünschen des Teams ab. Grundsätzlich empfiehlt sich dieses Vorgehen, da sich anschließend mittels einem Task Board (Seite 10) der Arbeitsfortschritt besser steuern und überwachen lässt. (Rubin, 2014, S. 377-388; Preußig, 2015, S. 148-150)

Sprint

Ist das Sprint Planning abgeschlossen und ein Sprint Backlog gefunden, kann mit der Sprintausführung gestartet werden. In dieser findet die eigentliche Produktentwicklung, sei es nun Software oder Hardware, statt. Das Team trifft sich zur gegenseitigen Abstimmung und Informationsweitergabe zu einem täglichen Standup-Meeting, dem Daily Scrum. Ein Task Board (physisch oder digital) kann hier als visuelle Unterstützung dienen. Die Zuordnung der Tasks auf die unterschiedlichen Entwickler erfolgt vollkommen autonom und darf unter keinen Umständen weder vom Scrum Master, noch vom Product Owner selbstständig geändert werden. Diese müssen jedoch darauf achten, dass die Entwickler nicht zu viele Tasks gleichzeitig abarbeiten und deren Produktivität dadurch abflacht (siehe WIP, Seite 11). Mit Fortschritt des Sprints wandern also immer mehr Tasks in die Spalte „Done“ des Task Boards. Hier ist es wichtig eine gute „Definition of Done“ (Seite 11) festzulegen, um spätere Diskussionen und Produktfehler zu vermeiden. Neben dem Task Board empfiehlt es sich auch einen Burn-Down-Chart (Seite 13) anzulegen. Dieses lässt sich einfach erstellen, kann mit dem Task Board verknüpft werden und hilft bei der Arbeitsfortschrittskontrolle. (Rubin, 2014, S. 389-403; Preußig, 2015, S. 147-148)

Produktinkrement

Am Ende des Sprints werden idealerweise alle User Stories des Sprint Backlogs abgearbeitet und damit ein auslieferungsfähiges Produktinkrement erstellt. Darunter soll aber nicht verstanden werden, dass das Produkt wirklich ausgeliefert wird, sondern die vorher zugesagten Items des Sprint Backlog auch wirklich umgesetzt wurden. (Rubin, 2014, S. 389-403)

Sprint Review

Im nächsten Meeting, dem Sprint Review, findet eine Überprüfung des geschaffenen Produktinkrements statt. Hier ist vor allem die Teilnahme der wichtigsten Stakeholder, allen voran des Kunden, essentiell. Denn dieser hat das Produkt in Auftrag gegeben und

nur er kann auch beurteilen, ob das Inkrement seinen Erwartungen entspricht und die Entwicklung in die richtige Richtung geht. Sehen der Kunde oder auch andere Stakeholder Änderungsbedarf, kann dies in dem Meeting sogleich diskutiert werden. Änderungswünsche werden dann wieder im Kollektiv bewertet und dem Product Backlog zugeordnet. (Rubin, 2014, S. 405-415; Preußig, 2015, S. 150-151)

Sprint Retrospektive

Das letzte Meeting in einem Sprint ist die Sprint Retrospektive. Hier erfolgt das Review nicht auf Produkt- sondern auf Prozess-Ebene. Das gesamte Scrum Team bewertet in diesem Meeting alle Prozesse und angewendete Techniken innerhalb eines Sprints, diskutiert mögliche Schwächen und definiert Verbesserungspotentiale. Das Ziel sind also konkrete Aktionen zur Prozessverbesserung, die sofort im nächsten Sprint umgesetzt werden können. Häufig ist jedoch die Sprint Retrospektive das am wenigsten geschätzte Ereignis innerhalb des Scrum Frameworks. Viele Teams sehen es als zeitraubend und nicht notwendig. Aber genau dieses Meeting bildet neben dem Produkt Review die Grundlage des kontinuierlichen Verbesserungsprozesses und sollte deshalb unter keinen Umständen ausgelassen werden. (Rubin, 2014, S. 417-436; Preußig, 2015, S. 151)

Neue Iteration

Mit Abschluss dieses Meetings kann direkt in den nächsten Sprint übergegangen werden. Dieser startet wieder mit dem Product Backlog. Die neuen Anforderungen werden vom Product Owner zusammen mit dem Development Team wieder in User Stories übergeführt und über eine anschließende Aufwandsabschätzung und Priorisierung dem bestehenden Product Backlog zugeführt. Auch beim nachfolgenden Sprint Planning Meeting müssen mögliche Veränderungen miteinbezogen werden. So hat eine verbesserte Team Velocity direkten Einfluss auf das Sprint Backlog, da mehr Items aus dem Product Backlog mit in den kommenden Sprint übernommen werden können. Wird dieser Umstand nicht berücksichtigt, kann es zu Leerzeiten und damit zu sinkender Produktivität kommen. Dies sind nur wenige Beispiele möglicher Veränderungen beziehungsweise Verbesserungen. Die Aufgabe des Product Owners und Scrum Masters ist es, den Überblick zu bewahren und für eine reibungslose Integration von Verbesserungen auf Produkt- und Prozessebene zu sorgen. (Rubin, 2014, S. 417-436)

3 Klassisch versus Agil

In diesem Kapitel werden die Unterschiede und Gemeinsamkeiten agiler und klassischer Projektmanagementansätze gegenübergestellt (siehe Kapitel 3.1). Weiteres wird untersucht, worin die Hauptgründe für nicht erfolgreiche Projekte liegen und wie agile Ansätze helfen können, diese Bilanz zu verbessern (siehe Kapitel 3.2). Danach werden beide Ansätze nochmals betrachtet und Vorteile des agilen Ansatzes abgeleitet (siehe Kapitel 3.4). Welche Projekte von einem agilen Vorgehen schlussendlich profitieren können und wo die Grenzen liegen, wird in Kapitel 3.3 diskutiert.

3.1 Unterschiede und Gemeinsamkeiten

Die agilen Methoden weisen große Unterschiede zum klassischen Vorgehen auf, es können aber auch Gemeinsamkeiten gefunden werden. Im Folgenden werden die wichtigsten Gemeinsamkeiten und Unterschiede beider Ansätze dargestellt.

3.1.1 Gemeinsame Basis

Die gemeinsame Basis beider Ansätze bildet der Projektmanagementprozess. Dieser wird, egal ob es sich um ein klassisches oder agiles Projekt handelt, von den meisten Experten gleich definiert und ist in Abbildung 16 dargestellt. (Timinger, 2017, S. 162)

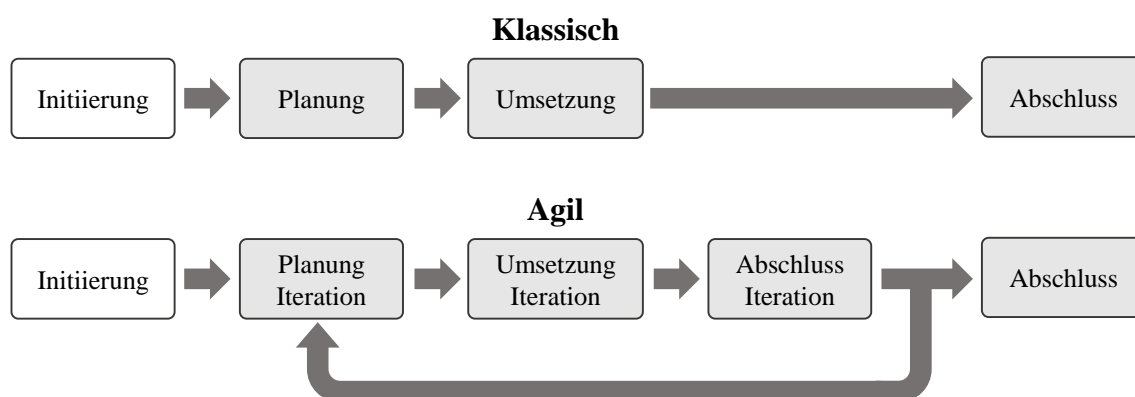


Abbildung 16: Grundlegender Projektmanagementprozess
Quelle: In Anlehnung an (Timinger, 2017, S. 162)

Jedes Projekt startet mit einer Initiierungsphase. In dieser werden die Projektziele und die Anforderungen des Kunden grob definiert. Die Planungsphase stellt im klassischen

Projektmanagement einen umfangreichen Prozess dar, da hier der gesamte Inhalt und Umfang des Projektes festgelegt wird, wohingegen bei agilem Vorgehen nur eine Planung der nächsten Iteration stattfindet. Im nächsten Prozessschritt erfolgt die Umsetzung, also die Abarbeitung vorher definierter Aufgaben. Auch hier werden bei agilem Vorgehen nur Aufgaben der aktuellen Iteration bearbeitet, beim klassischen Vorgehen jedoch alle anfangs festgelegten Aufgaben. Nach Abschluss der Iteration startet man mit der Planung und Umsetzung einer neuen Iteration. Dieser iterative, inkrementelle Vorgang wird solange wiederholt, bis das Projektziel erreicht und das Projekt damit abgeschlossen wird. Bei klassischem Vorgehen werden, im Gegensatz dazu, schon in der ersten Umsetzungsphase alle Aufgaben sequenziell abgearbeitet und das Projekt anschließend beendet. (Timinger, 2017, S. 162-163)

3.1.2 Zieldreieck

Das Zieldreieck (siehe Abbildung 17) des klassischen und agilen Projektmanagements besteht aus drei Komponenten, den Kosten (z.B. Ressourcen), der Zeit (z.B. Termine und Projektdauer) und dem Umfang (z.B. Leistung und Qualität). (Preußig, 2015, S. 35-36)

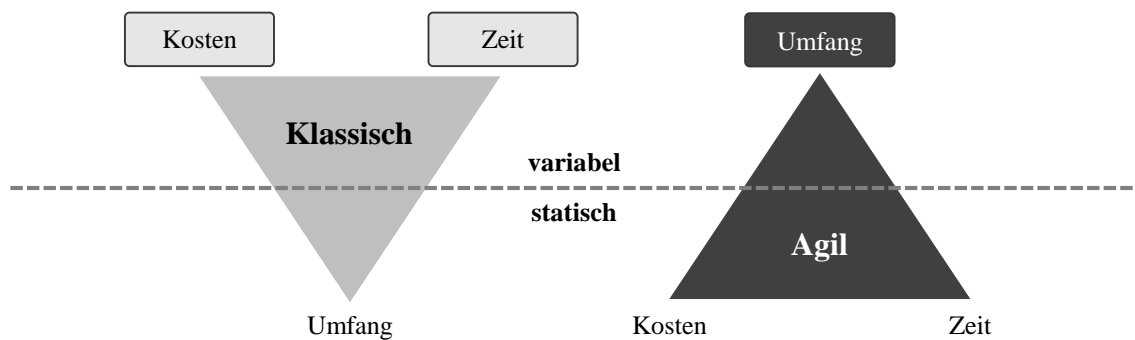


Abbildung 17: Zieldreieck
Quelle: In Anlehnung an (Proagile, 2017)

Um ein vorher definiertes Projektziel zu erreichen, stehen einem Projektmanager diese drei Komponenten als Stellschrauben zur Verfügung. Beim klassischen Projektmanagement geht man zu Beginn davon aus, dass ein Projekt und dessen Endzustand detailliert geplant werden kann. Man definiert die Komponente Umfang, wohingegen die Komponenten Zeit und Kosten auf dieser Basis geschätzt werden. Jegliche Änderungen und damit verbundene Abweichungen vom Plan werden als Störfaktor angesehen und sollten demnach möglichst vermieden werden. Kommt es

dennoch zu unvorhergesehenen negativen Einflüssen, werden in den meisten Fällen Termine des Projektes nach hinten verschoben oder die Kosten erhöht, um vorher definierte Leistungsumfänge zu erreichen. Die Komponenten „Zeit“ und „Kosten“ können hier als erste Stellschraube identifiziert werden (siehe Abbildung 17). Anders sieht dies beim agilen Projektmanagement aus. Hier wird die Komponente Zeit als besonders hohes Gut angesehen (siehe Timeboxing, Seite 12). Da Leistungsumfänge iterativ festgelegt und geplant werden, gelten diese auch als verhandelbar und stellen bei agilen Projekten die erste Stellschraube dar. (Preußig, 2015, S. 35-36)

3.1.3 Rahmenbedingungen

Neben dem Projekt an sich hat auch das Projektumfeld einen großen Einfluss auf die Entscheidung, ob ein Projekt agil oder eher klassisch gemanagt werden soll. In Tabelle 1 werden die optimalen Rahmenbedingungen für agiles Vorgehen dargestellt. (Preußig, 2015, S. 37-39)

Optimale Rahmenbedingungen für agiles Vorgehen	
Projekt	Projektumfeld
Relativ kleine Projektteams	Enge Zusammenarbeit mit Stakeholder
Projektteam an einem Standort	Projektanforderungen müssen zu Beginn noch nicht exakt definiert sein
Wenige einzelne Spezialisten (eher Generalisten)	Projektteam darf selbstorganisiert arbeiten
Lieferprodukt des Projektes ist in Teilprodukte zerlegbar	Projektteam arbeitet nur für dieses Projekt
Gesamtaufwand ist in kleine Aufgaben zerlegbar	

Tabelle 1: Optimale Rahmenbedingungen für agiles Vorgehen
Quelle: In Anlehnung an (Preußig, 2015, S. 37-38)

3.1.4 Die Unterschiede im Überblick

Tabelle 2 zeigt nochmals alle wesentlichen Unterschiede zwischen agilen und klassischen Projekten.

Wesentliche Unterschiede im Überblick	
Klassisch	Agil
Anforderungen zu Beginn bekannt	Anforderungen zu Beginn unscharf
Änderungen von Anforderungen während des Projektverlaufs schwierig	Änderungen an Anforderungen während des Projektverlaufs eingeplant
Hohe Kosten für spätere Änderungen	Mäßige Kosten für spätere Änderungen
Anforderungsbeschreibung aus technischer Sicht (Features)	Anforderungsbeschreibung aus Kundensicht (User Stories)
Sequenzieller Entwicklungsprozess	Iterativer Entwicklungsprozess
Starrer Projektmanagementprozess	Fortlaufende Prozessverbesserungen
Kunde sieht nur Endergebnis oder wenige Zwischenergebnisse	Kunde sieht viele Zwischenergebnisse und bewertet diese
Erste Stellschraube ist die Zeit	Erste Stellschraube ist der Aufwand
Klare Hierarchie	Selbstorganisierte Teams
Sehr große Teams möglich	Relativ kleine Teams nötig
Viele Spezialisten im Team	Viel gemeinsame Verantwortung
Team kann verteilt sitzen	Team sollte möglichst zusammensitzen
Aufgaben werden von oben zugeteilt	Aufgabenvergabe erfolgt selbständig
Viel Kommunikation über Dokumente und lange Meetings	Viele informelle Kommunikation und Standup-Meetings
Aufwandsabschätzung durch Projektleiter oder Experten	Aufwandsabschätzung gemeinsam im Team

Tabelle 2: Wesentliche Unterschiede zwischen klassischen und agilen Projekten

Quelle: In Anlehnung an (Preußig, 2015, S. 40-42)

3.2 Gründe für nicht erfolgreiche Projekte

In diesem Kapitel werden die Hauptgründe für nicht erfolgreich abgeschlossene Projekte angeführt. Dabei wird auf zwei Studien der „GPM und PA Consulting Group GmbH“ und der „Standish Group“ verwiesen. Die GPM und PA Consulting Group GmbH untersuchte dabei die Erfolgs- und Misserfolgskriterien von Projekten von 82 Unternehmen aus den unterschiedlichsten Branchen, wie Automotive, IT, Maschinenbau, Energie und anderen. Die Standish Group dagegen, legt mit dem jährlich erscheinenden Chaos-Report (Standish Group, 2015) den Hauptfokus auf IT-Projekte. Es erfolgt nachstehend eine Unterteilung in Allgemeine Projekte (siehe Kapitel 3.2.1) und IT-Projekte (siehe Kapitel 3.2.2), in denen jeweils die größten negativen Einflussfaktoren für ein Projekt angeführt werden.

3.2.1 Allgemeine Projekte

Laut der 2008 veröffentlichten Studie der GPM und PA Consulting Group GmbH scheitern mehr als 30 % aller Projekte. Die Hauptgründe hierfür sind in der untenstehenden Abbildung (Abbildung 18) dargestellt. (Engel & Quadejacob, 2009)

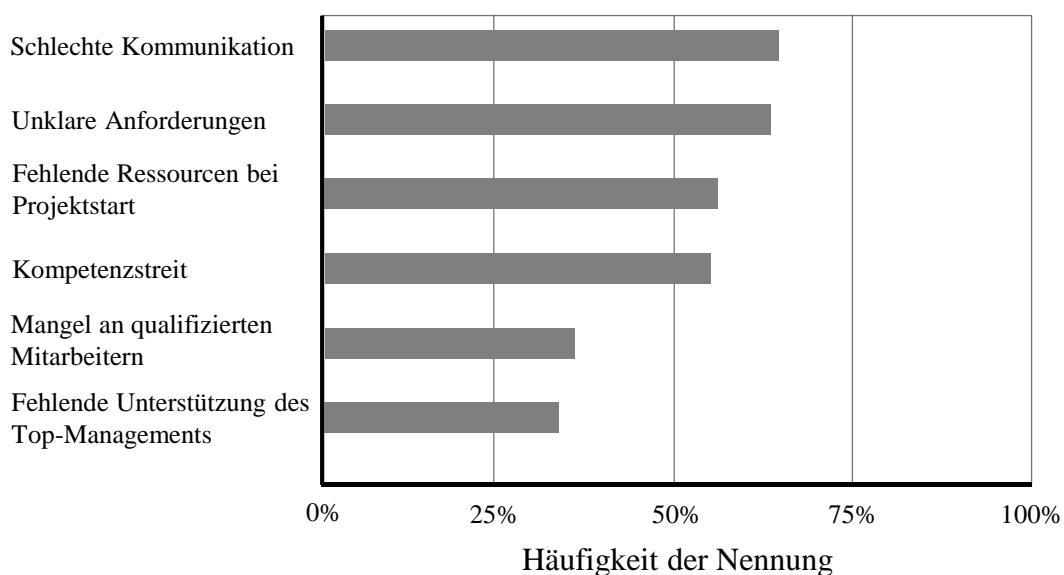


Abbildung 18: Misserfolgskriterien für ein Projekt
Quelle: In Anlehnung an (Engel & Quadejacob, 2009)

Spitzenreiter der Misserfolgskategorien bilden dabei „Schlechte Kommunikation“ und „Unklare Anforderungen“, gefolgt von „Fehlenden Ressourcen bei Projektstart“ und „Kompetenzstreits“.

3.2.2 IT-Projekte

Die Standish Group führt seit 1994 eine jährliche Studie, den sogenannten Chaos-Report, zu Erfolgs- und Misserfolgskategorien in IT-Projekten durch und hat seitdem mehr als 40.000 Einzelprojekte untersucht. In dieser Studie werden drei Erfolgskategorien angeführt (erfolgreiche Projekte, teilweise erfolgreiche Projekte und fehlgeschlagene Projekte). Ein Projekt wird als erfolgreich eingestuft, wenn der anfangs geplante Umfang umgesetzt und das Projekt innerhalb des Budgets und Zeitrahmens abgeschlossen wird. Kann eine dieser Komponenten nicht eingehalten werden, wird es den teilweise erfolgreichen Projekten zugeteilt. Wird es gar nicht abgeschlossen, kann es den fehlgeschlagenen Projekten zugeteilt werden. Abbildung 19 zeigt das Ergebnis des Chaos Reports im Zeitraum von 1994-2015. (Standish Group, 2015)

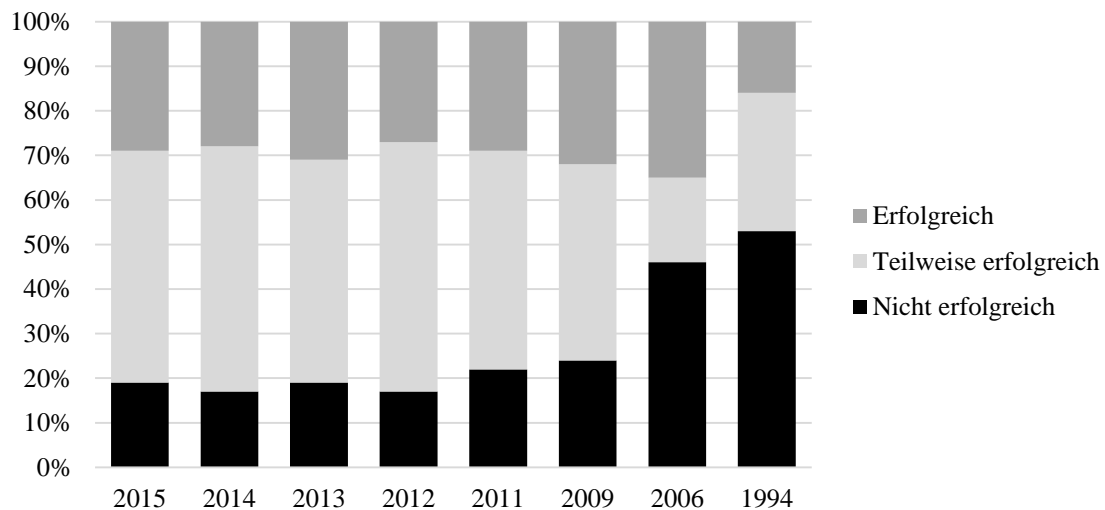


Abbildung 19: Anteil erfolgreicher bzw. gescheiterter IT-Projekte
Quelle: In Anlehnung an (Standish Group, 2015)

Der Anteil nicht erfolgreicher Projekte ist zwar in den letzten 20 Jahren stark zurückgegangen, die erfolgreichen Projekte konnten aber nur leicht zulegen. Betrachtet man nur die letzten 10 Jahre, gibt es hier kaum negative, aber auch kaum positive Bewegungen. Der Chaos-Report nennt folgende Misserfolgskategorien (Tabelle 3).

Misserfolgsk Faktoren
Fehlende Zusammenarbeit
Unvollständige und unklare Anforderungen sowie häufige Anforderungsänderungen
Fehlende Unterstützung durch das obere Management

Tabelle 3: Misserfolgsk Faktoren nach Chaos-Report
Quelle: In Anlehnung an (Standish Group, 2015)

Obwohl bei beiden Studien sehr unterschiedliche Branchen und Projekte betrachtet wurden, können viele Gemeinsamkeiten gefunden werden. So wird eine schlechte Kommunikation bzw. fehlende Zusammenarbeit als einer der Hauptfaktoren angeführt. Aber auch unklare bzw. unvollständige Anforderungen sowie fehlende Unterstützung des oberen Managements werden als negative Faktoren beschrieben.

3.3 Einsatzgebiete

Wie schon in Kapitel 2.2 (Scrum – Grundlagen) angeführt, ist das Scrum Framework sehr allgemein gehalten und kann deshalb in den unterschiedlichsten Branchen angewendet werden, wobei es derzeit hauptsächlich in der Softwareentwicklung Anwendung findet. In welchen Bereichen abgesehen von der Softwareentwicklung, Scrum eine Alternative zum klassischen Projektmanagement sein kann, wird im Folgenden diskutiert. Ralf D. Stacey (Stacey, 1996) bietet mit dem nach sich benannten Stacey Landscape Diagram (siehe Abbildung 20) eine gute Grundlage. Dieses visualisiert die Anwendungsbereiche von Scrum sowie klassischer Projektabwicklung und kann als Entscheidungshilfe herangezogen werden. In dem Diagramm wird der Lösungsansatz (skaliert von bekannt bis unbekannt) den Anforderungen (klar bis unklar) gegenübergestellt. Ein Produkt kann damit in einen der vier Bereiche (einfach, kompliziert, komplex und chaotisch) eingeordnet werden. (Pieper & Roock, 2017, S. 26)

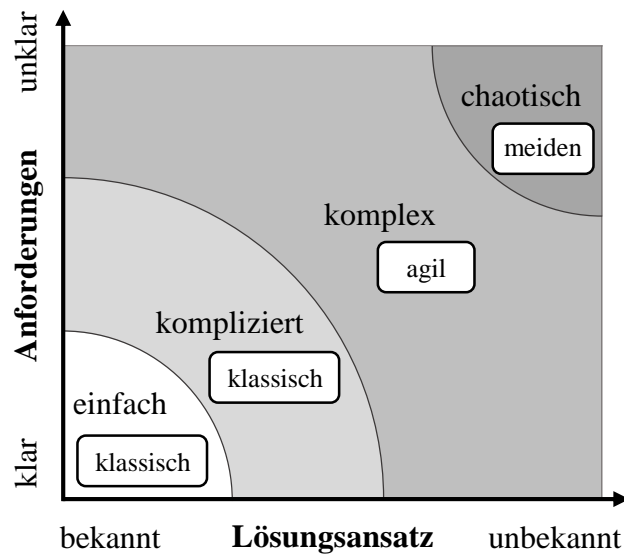


Abbildung 20: Stacey Landscape Diagram
 Quelle: In Anlehnung an (Stacey, 1996)

Einfach

Sind Anforderung klar definiert und ein Lösungsansatz bzw. eine Lösungstechnologie vorhanden und unbestritten, kann das Produkt oder die Aufgabe als einfach bezeichnet werden. Die Entwicklung des Produktes sollte in diesem Bereich mit bekannten „Best Practices“ erfolgen. Scrum kann grundsätzlich angewendet werden, stellt aber bei weitem nicht die effizienteste Methode dar. Ein vorher definierter Routineprozess mit klassischem Projektmanagement wäre hier besser geeignet. (Rubin, 2014, S. 41)

Kompliziert

Ist der Lösungsansatz teilweise bekannt und Anforderungen nicht vollständig klar definiert, gelangt man in den komplizierten Bereich. Auch hier empfiehlt sich die Entwicklung des Produktes mit „Best Practices“, wobei es viele Lösungsmöglichkeiten geben kann. Zur Auswahl der vielversprechendsten Lösung sind deshalb Expertenmeinungen notwendig. Man befindet sich hier im typischen Anwendungsbereich für die klassische Projektabwicklung. Aber auch in diesem Bereich ist eine agile Abwicklung mit Scrum möglich, wobei sich ein hybrider Ansatz, also eine Mischung von agilen und klassischen Elementen besser eignen würde, als der reine Scrum Prozess. (Rubin, 2014, S. 40-41)

Komplex

Werden Anforderungen immer unklarer und sind Lösungsansätze eher unbekannt, liegt man im komplexen Bereich. Das Produkt ist zu Beginn mit vielen unbekanntem Faktoren behaftet. Eine langfristige Planung ist nicht möglich und vorher definierte Routinelösungen führen selten zum Erfolg. Vielmehr bedarf es einer kontinuierlichen Überprüfung (Inspect) und Anpassung (Adapt) der Entwicklung. Das Scrum Vorgehensmodell eignet sich mit seiner iterativen, inkrementell und adaptiven Planung und Umsetzung hervorragend für solche Produkte bzw. Problemstellungen. (Rubin, 2014, S. 40)

Chaotisch

Sind Anforderungen und Lösungsansätze unbekannt, ist weder Scrum noch ein klassischer Ansatz die richtige Wahl. Solche Projekte sollte man möglichst meiden, da der Ausgang unberechenbar ist. Liegt dennoch ein chaotisches Projekt vor, muss man versuchen, dieses zumindest in den komplexen Bereich überzuführen. Praktische Maßnahmen wie Experimentieren, Lernen oder Hypothesenbildung können hier unterstützen. (Rubin, 2014, S. 41)

Grundsätzlich kann Scrum also für jedes Projekt, ob einfach, kompliziert oder komplex angewendet werden. Die Erfahrung hat aber gezeigt, dass es sich vor allem für komplexe Thematiken sehr gut eignet und in diesem Bereich dem klassischen Vorgehen überlegen ist. Bei einfachen und komplizierten Projekten muss abgewogen werden, ob nicht doch ein klassischer Ansatz die bessere Variante wäre.

3.4 Vorteile eines agilen Ansatzes

In diesem Kapitel werden einige der nennenswertesten Vorteile von agilem Vorgehen gegenüber klassischem Projektmanagement dargestellt und erklärt. Tabelle 4 gibt dazu einen kurzen Überblick.

Vorteile von agilem Vorgehen
Kürzere „Time to Market“
Höhere Qualität
Größere Effektivität
Mehr Innovation
Zufriedenere Mitarbeiter

Tabelle 4: Vorteile von agilen Ansätzen
Quelle: Eigene Darstellung

Kürzere „Time to Market“

Die „Time to Market“, also jene Zeit zwischen Projektstart und Markteinführung kann bei agiler Entwicklung reduziert werden und hat zwei Ursachen. Einerseits führt der kontinuierliche Verbesserungsprozess (Inspect/Adapt) zu einer besseren Team Velocity und damit zu einer effektiveren Abwicklung von Anforderungen. Andererseits wird durch die Priorisierung des Product Owners gewährleistet, dass nur die wichtigsten Anforderungen entwickelt werden und es somit zu einer geringeren Ressourcenverschwendung kommt. (Pieper & Roock, 2017, S. 18-20)

Höhere Qualität

Aus Sicht des Kunden ist Qualität ein Maß dafür, wie gut das entwickelte Produkt den bestehenden Anforderungen entspricht. Aufgrund der vielen Iterationsschleifen innerhalb eines Scrum Prozesses, erhält man vom Kunden regelmäßiges Feedback zum Produkt. Dies sorgt einerseits dafür, dass die Richtung der Entwicklung kontinuierlich den Wünschen des Kunden angepasst werden kann und andererseits, mögliche Fehler früher erkannt und eliminiert werden können. (Pieper & Roock, 2017, S. 21)

Größere Effektivität

Effektivität kann als Maß für die Zielerreichung angesehen werden. Eine einfache Definition dazu lautet „Die richtigen Dinge tun“ (Nicht zu verwechseln mit Effizienz „Die Dinge richtig tun“). Auch dieser Vorteil basiert auf der kontinuierlichen Priorisierung von Anforderungen, dem Timeboxing (siehe Pareto-Prinzip, Seite 12) und den stetigen Feedbackschleifen mit dem Kunden. (Preußig, 2015, S. 106-109)

Mehr Innovation

Laut Sawyer (Sawyer, 2007) entsteht Innovation vor allem aus Diversität und guter Zusammenarbeit. Scrum bietet deshalb mit selbstorganisierten, cross-funktionalen Teams, die auch meist in gemeinsamen Räumlichkeiten sitzen hervorragende Voraussetzungen dafür. (Pieper & Roock, 2017, S. 22)

Zufriedenere Mitarbeiter

Dan Pink (Pink, 2011) nennt 2 Arten der Motivation. Die extrinsische und intrinsische Motivation. Erstere wird durch äußere Reize, wie zum Beispiel Angst vor Bestrafung oder Bezahlung der Arbeit hervorgerufen und deshalb von Pink als schädlich für die Wissensarbeit angesehen. Im kompletten Gegensatz dazu steht die intrinsische Motivation, eine aus sich selbst entstehende Motivation. Hier werden Tätigkeiten durchgeführt, weil sie Spaß machen, interessant sind oder man diese als sinnvoll und herausfordernd ansieht. Die intrinsische Motivation kann nur dann entstehen, wenn drei Voraussetzungen erfüllt sind. (Pieper & Roock, 2017, S. 22-23)

- *Purpose: Ich muss den Zweck meiner Arbeit verstehen und sinnvoll finden.*
- *Mastery: Ich muss an der Aufgabe wachsen können, darf aber nicht maßlos überfordert sein.*
- *Autonomy: Ich kann das Wie der Aufgabenerledigung weitestgehend selbst bestimmen.* (Pieper & Roock, 2017, S. 22)

Mit Scrum werden diese drei Voraussetzungen sehr gut erfüllt. Ein Scrum Team ist selbstorganisiert, die Aufgabenverteilung erfolgt demnach autonom (*Autonomy*), Anforderungen werden zusammen entwickelt und nach Kundennutzen priorisiert (*Purpose*) und der kontinuierliche Verbesserungsprozess auf Produkt- und Prozessebene

führt zu stetiger Verbesserung des Teams und der Individuen (*Mastery*). (Pieper & Roock, 2017, S. 22-23)

Oben genannte Vorteile ergeben sich aber nur, wenn alle Beteiligten, also die Stakeholder und das Scrum Team selbst, die agilen Werte, Prinzipien, Techniken und Methoden verstehen und auch leben. Inwieweit eine Abweichung möglich ist, um dennoch Vorteile der agilen Arbeitsweise genießen zu können, kann nicht pauschal beantwortet werden. Hier gilt der Grundsatz „Mehr bringt mehr“.

4 Zusammenarbeitsmodelle

Das Dienstleistungsangebot von ALTEN umfasst einerseits die Abwicklung von Kundenprojekten mittels Werkverträgen und andererseits, die Bereitstellung von Mitarbeitern für Kundenprojekte, bei denen diese ausschließlich nach Kundenvorgaben arbeiten und ALTEN keinerlei Projektverantwortung übernimmt (Arbeitnehmerüberlassung). Die ALTEN Gruppe nennt dazu fünf verschiedene Zusammenarbeitsmodelle, auch IQP-Modelle genannt (siehe Abbildung 21). Diese regeln die Zusammenarbeit mit dem Kunden in Bezug auf die Angebotsgestaltung und die Aufwands- bzw. Leistungsabrechnung. Die Modelle 1 und 2 beziehen sich auf die Arbeitnehmerüberlassung, wohingegen die Modelle 3 bis 5 in den Bereich der Werkverträge fallen.

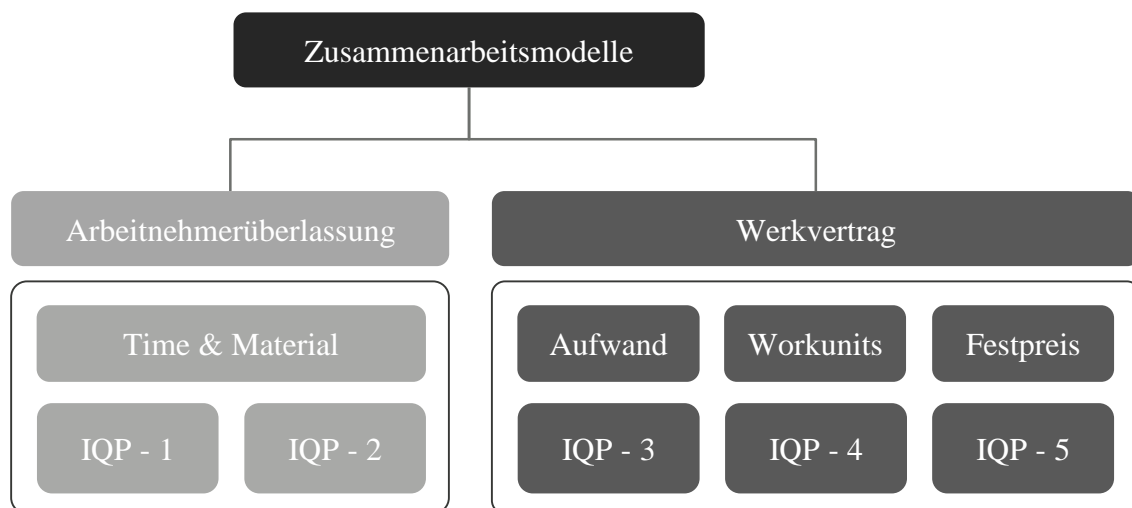


Abbildung 21: Zusammenarbeitsmodelle von ALTEN
Quelle: Eigene Darstellung

Wie sich Werkverträge zur Arbeitnehmerüberlassung abgrenzen lassen, wird in Kapitel 4.1 diskutiert. Da sich die Masterarbeit ausschließlich auf die unterschiedlichen Werkverträge bezieht, werden diese in Kapitel 4.2 näher betrachtet.

4.1 Abgrenzung

Ein Werkvertrag ist wie auch die Arbeitnehmerüberlassung eine Übereinkunft zweier Parteien. Der Vertragsgegenstand eines Werkvertrages ist die Erbringung einer vorher definierten Leistung, die vom Kunden abgenommen werden muss. Die Vergütung erfolgt

hier nach Erfolg. Bei der Arbeitnehmerüberlassung bildet die Bereitstellung geeigneter Mitarbeiter den Vertragsgegenstand. Nachfolgende Tabelle zeigt noch weitere Unterschiede zwischen einem Werkvertrag und einer Arbeitnehmerüberlassung (siehe Tabelle 5).

Werkvertrag	Arbeitnehmerüberlassung
Erbringung definierter Leistung „Werk“	Bereitstellung geeigneter Mitarbeiter
Vergütung nach Erfolg (Abnahme)	Vergütung nach Stundensatz
Abnahme des Werkes notwendig	Keine Abnahme erforderlich
Weisung nur durch Alten	Weisung durch den Kunden
Freie Auswahl der Mitarbeiter	Integration des Mitarbeiters in Organisation des Kunden
Mängelhaftung (Schadenersatz, Rücktritt, Minderung, etc.)	Haftung nur für fehlerhafte Auswahl des Personals

Tabelle 5: Abgrenzung Arbeitnehmerüberlassung zu Werkvertrag
Quelle: In Anlehnung an ALTEN-interne Dokumente

4.2 Werkverträge

Da die IQP-Modelle 3 bis 5 den zentralen Betrachtungspunkt der Masterarbeit darstellen, erfolgt eine nähere Beschreibung in diesem Kapitel. Zur Optimierung der Zusammenarbeit mit dem Kunden hat ALTEN für das Werkvertragsmodell drei Zusammenarbeitsmodelle entwickelt, die dann nach Kundenwunsch eingesetzt werden (siehe Abbildung 22).

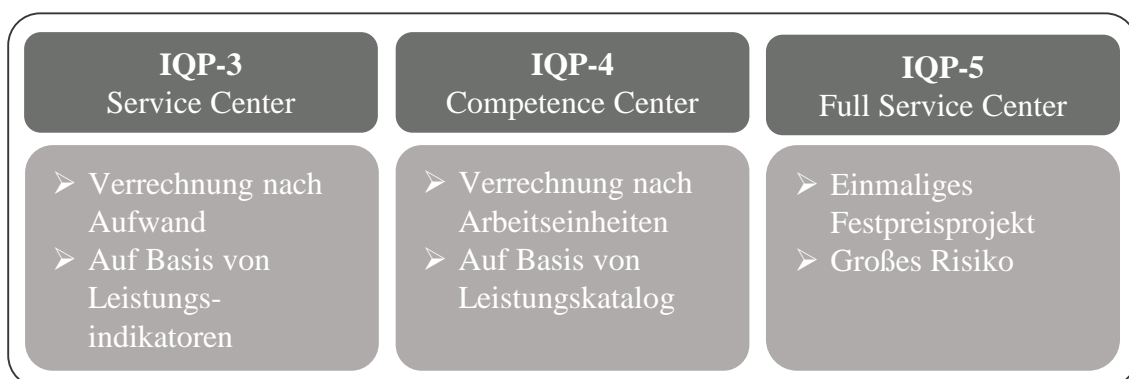


Abbildung 22: Werkvertragsmodelle
Quelle: In Anlehnung an ALTEN-interne Dokumente

Dabei wird unterschieden, wieviel Eigenverantwortung ALTEN im Rahmen des Projektes übernehmen soll. So überträgt der Kunde bei einem IQP-5 Modell, auch Festpreisprojekt genannt, die gesamte Entwicklungsverantwortung auf ALTEN. Bei den Modellen IQP-3 und IQP-4 hingegen, wird die Verantwortung auf einzelne Phasen eines Gesamtprojektes reduziert. Wie sich diese nun genau voneinander unterscheiden, wird in den folgenden Kapiteln behandelt.

4.2.1 IQP-3 Modell

Da zu Beginn eines Projektes der Umfang oft schwer abgeschätzt werden kann, wünscht der Kunde häufig nur eine Beauftragung von Teilumfängen, anstatt eines Komplettpaketes. Eine Möglichkeit dazu bietet das IQP-3 Modell, auch Service Center Modell genannt. Bei diesem erfolgt die Abwicklung auf Basis von Leistungsindikatoren, für die es ein vom Kunden definiertes Gesamtbudget gibt. Mögliche Leistungsindikatoren wären zum Beispiel Story Points oder vorher festgelegte Kriterien wie:

- Entwicklung einfacher Software
- Entwicklung mittlerer Software
- Entwicklung schwerer Software

Die Verrechnung dieser Leistungen erfolgt aufwandsbezogen im zeitlichen Fortschritt des Projektes. Dazu muss jeder Leistung ein Aufwand hinterlegt werden, welcher anhand der dafür benötigten Personenstunden definiert wird. Der geleistete Anteil kann dann vom Gesamtbudget in Abzug gebracht werden (siehe Abbildung 23).

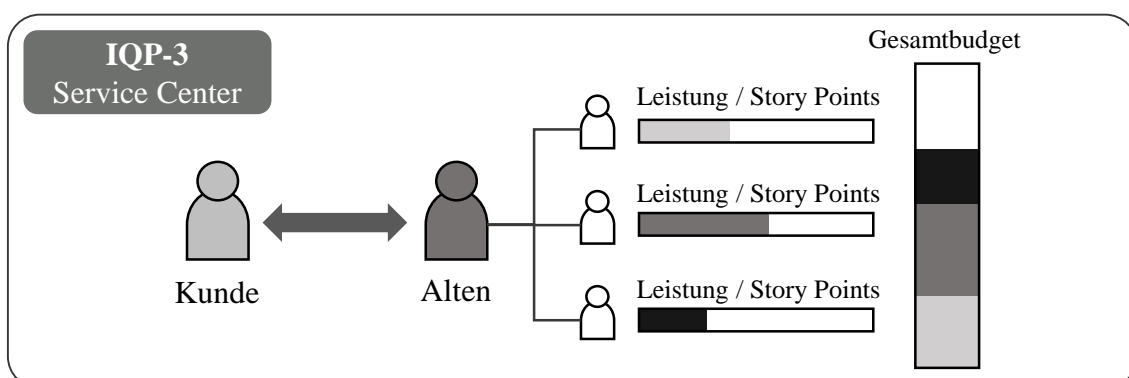


Abbildung 23: IQP-3 Modell

Quelle: In Anlehnung an ALTEN-interne Dokumente

Das Modell 3 basiert also auf der Berechnung von Stunden, wobei ein Ergebnis bzw. eine Leistung vom Kunden abgenommen wird. Angewendet wird dieses Modell, wenn Produktdetails zu Beginn noch nicht absehbar sind, der Kunde kurzfristige Kapazitätserweiterungen benötigt, seine Infrastruktur begrenzt oder ausgelastet ist. Dies ermöglicht hohe Flexibilität für neue Entwicklungsumfänge und einen gezielten Know-How Transfer von außen.

4.2.2 IQP-4 Modell

Das zweite mögliche Modell für die Beauftragung von Teilumfängen eines Gesamtprojektes ist das IQP-4 Modell, auch Competence Center Modell genannt. Hier erfolgt die Abrechnung nicht wie beim IQP-3 Modell nach Aufwand im zeitlichen Kontext, sondern auf Basis von vorher definierten Arbeitspaketen (AP). Abbildung 24 zeigt diesen Zusammenhang.

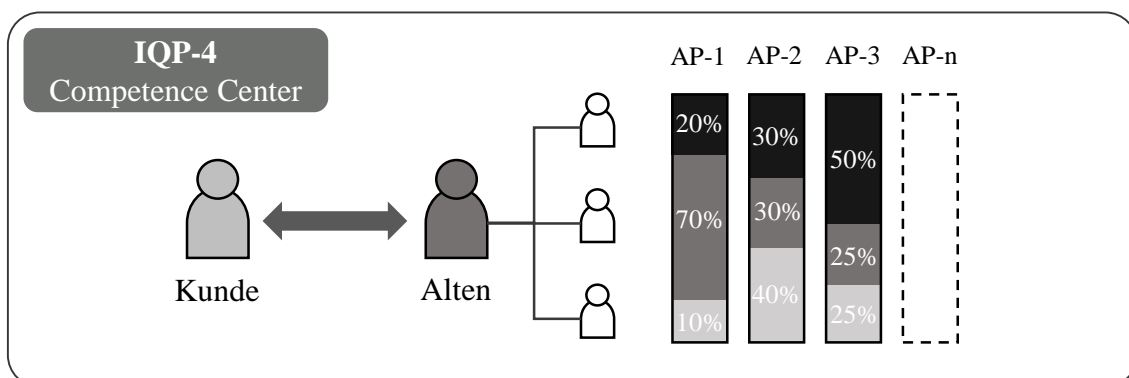


Abbildung 24: IQP-4 Modell

Quelle: In Anlehnung an ALTEN-interne Dokumente

Man definiert hier auf Basis des Lastenhefts des Kunden einen Leistungskatalog, bestehend aus unterschiedlichen Arbeitspaketen zu jeweils festen Preisen, die flexibel vom Kunden abgerufen werden können. Die ALTEN Mitarbeiter können je nach ihrer Qualifikation in den unterschiedlichen Arbeitspaketen ihre Leistung erbringen. Dies ergibt für ALTEN große Vorteile in Bezug auf eine flexible personelle Gestaltung. Angewendet wird das IQP-4 Modell vor allem für wiederkehrende Tätigkeiten, deren Umfang gut abgeschätzt werden kann.

4.2.3 IQP-5 Modell

Beim IQP-5 Modell erfolgt vom Kunden eine Beauftragung eines Komplettpaketes, bei dem ALTEN die gesamte Entwicklungsverantwortung übernimmt. Hier werden die Rahmenbedingungen wie Anforderungen, Zeit und Preis zu Beginn klar definiert (siehe Abbildung 25). Dies ermöglicht dem Kunden eine Auslagerung einer vollumfänglichen Produktentwicklung bei festem Budget mit minimalem Risiko. Für ALTEN stellt dieses Modell jedoch die risikoreichste Variante dar, da von Beginn an alle Anforderungen und Produktspezifikationen bekannt sein und deren Aufwände richtig abgeschätzt werden müssen. Fordert der Kunde Anpassungen, so ist dies grundsätzlich möglich, kann aber nur über einen aufwändigen und kostenintensiven „Change Request Prozess“ realisiert werden. Ein Festpreisprojekt findet bei einmaligen, nicht stetig wiederkehrenden Umfängen Anwendung, wobei Produkt- und Qualitätsmerkmale von Beginn an klar definierbar sein müssen.

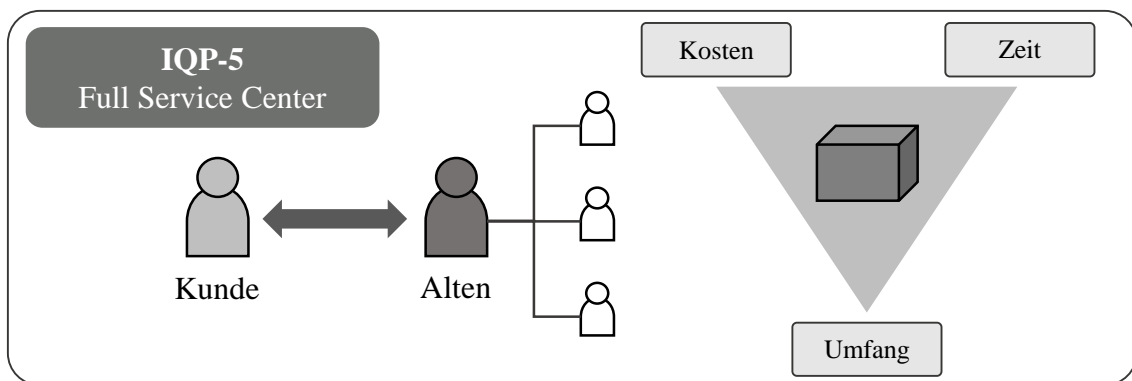


Abbildung 25: IQP-5 Modell

Quelle: In Anlehnung an ALTEN-interne Dokumente

5 Agile Vertragsgestaltung

Um ein Kundenprojekt erfolgreich abwickeln zu können, benötigt es neben einem guten Team vor allem eine gute Kooperation und Kommunikation innerhalb des Teams, aber auch mit dem Kunden. Die Art und Weise, über welche Kanäle die Kommunikation mit dem Kunden erfolgt, welche Mitwirkungspflichten dieser trägt, welche Projektrollen es gibt und vieles mehr, wird in einem Vertrag geregelt. Im Idealfall orientiert sich ein Vertragswerk an der Art der Entwicklung. So soll ein Vertrag den Rahmen für eine agile Projektabwicklung ermöglichen und fördern. Dazu müssen neben dem agilen Wertesystem (siehe Kapitel 2.1) auch das agile Vorgehensmodell im Vertragskonstrukt eingebunden werden. Welche Punkte im Falle agiler Abwicklung mit Scrum geregelt werden sollen wird in Kapitel 5.1 diskutiert. Im darauffolgenden Kapitel 5.2 werden die Grundzüge sowie Probleme von herkömmlichen Festpreisverträgen angeführt. Welche agilen Alternativen es dazu gibt, behandelt Kapitel 5.3. (Pieper & Roock, 2017, S. 25ff)

5.1 Wichtige Aspekte für agile Vertragsgestaltung

Agile Entwicklung mit Scrum basiert auf den agilen Wertevorstellungen, dem sogenannten agilen Mindset. Dieses muss in einem agilen Vertrag unbedingt seinen Niederschlag finden, wobei vor allem folgende drei agile Grundsätze berücksichtigt werden sollen. (Pieper & Roock, 2017, S. 30)

- Unvollständige Funktionsbeschreibung: *Wir wissen zu Projektbeginn noch nicht vollständig, wie das Produkt aussehen muss, um den erhofften Nutzen zu bringen.*
- Gemeinsames Verständnis: *Wir erzeugen ein gemeinsames Verständnis über agile Entwicklung*
- Kooperative Problemlösung: *Wir reagieren auf Probleme mit größter Nähe und nicht mit größter Distanz.* (Pieper & Roock, 2017, S. 30-31)

Zudem sollte auch das agile Vorgehensmodell Scrum mit seinen Rollen, Artefakten und Ereignissen in den Vertrag mit aufgenommen werden. Im Folgenden wird neben den drei Grundsätzen der Einfluss von Scrum auf das Vertragskonstrukt näher beschrieben. (Pieper & Roock, 2017, S. 30)

5.1.1 Unvollständige Funktionsbeschreibung

Wie schon in Kapitel 3.3 (Einsatzgebiete, Seite 39) angeführt lassen sich bei komplexen Produkten zu Beginn deren Anforderungen noch nicht genau abschätzen. Bei der agilen Entwicklung nach Scrum ermöglicht ein iteratives Vorgehen mit stetiger Überprüfung und Anpassung, Produkthanforderungen kontinuierlich nachzuschärfen, um somit den maximalen Kundennutzen zu generieren. Ein agiler Vertrag muss demnach so gestaltet werden, dass von vorher definierten Funktionen abgewichen werden kann und Anforderungen, die zu Beginn noch nicht bekannt waren, ohne Mehraufwand und Mehrkosten in die Entwicklung integriert werden können. (Pieper & Roock, 2017, S. 31)

5.1.2 Gemeinsames Verständnis

Das Vorgehensmodell Scrum basiert auf den im agilen Manifest niedergeschriebenen agilen Werten und agilen Prinzipien (siehe Kapitel 2.1). Für eine erfolgreiche Projektabwicklung müssen neben der Beherrschung des Scrum Prozesses auch die agilen Wertvorstellungen und Handlungsgrundsätze gelebt und verstanden werden. Da bei agilem Vorgehen auch der Kunde eng in die Entwicklung eingebunden werden muss, bedarf es einem gemeinsamen Verständnis, wie agile Entwicklung funktioniert und auf welchen Grundsätzen sie beruht. Es empfiehlt sich somit, das agile Manifest, oder zumindest die wichtigsten Werte und Prinzipien in einem agilen Vertrag anzuführen. (Pieper & Roock, 2017, S. 32)

5.1.3 Kooperative Problemlösung

Auch bei agilen Projekten wird es früher oder später zu Problemen und Missverständnissen kommen. Ein kooperativer Problemlösungsprozess sollte demnach in jedem agilen, aber auch klassischem Vertrag vereinbart werden. In diesem sollen idealerweise Termine (z.B. wöchentlich oder nach jedem Sprint Review), Teilnehmer und ein Eskalationsmechanismus festgelegt werden. Ein ebenfalls vorab im Vertrag definiertes Riskshare-Modell kann diesen Lösungsfindungsprozess noch forcieren. (Pieper & Roock, 2017, S. 32-35)

5.1.4 Rollen

Im Vertrag muss zwingend definiert werden, wer welche Scrum Rollen einnimmt und welche Rahmenbedingungen dabei einzuhalten sind. In den folgenden Unterteilungen werden die wichtigsten Aspekte, die zu berücksichtigen sind, kurz angeführt. (Pieper & Roock, 2017, S. 35-38)

Product Owner

Dieser sollte jedenfalls vom Kunden gestellt werden, da er sonst seiner Hauptaufgabe, der Pflege des Product Backlogs in den meisten Fällen nicht gerecht werden kann. Wird dennoch ein Product Owner auf Dienstleister Seite benötigt, müssen seine Aufgaben und Pflichten von Beginn an klar definiert werden. Fehlt dem Product Owner auf Kundenseite das Know-How für gewisse Scrum-Praktiken, wie Detaillierung von User Stories oder Epics, so muss das Development Team diese Lücke schließen. Auch solche Szenarien sollten im Vertrag abgesteckt werden. (Pieper & Roock, 2017, S. 36-38)

Scrum Master

Da der Scrum Master einen sehr engen Kontakt mit dem Development Team pflegt, sollte dieser vom Dienstleister bereitgestellt werden. Bei größeren Projekten kann es durchaus sinnvoll sein einen weiteren Scrum Master auf Kundenseite einzusetzen. Auch hier müssen dann deren Aufgaben und Pflichten im Vertrag niedergeschrieben werden. (Pieper & Roock, 2017, S. 36)

Development Team

Das Development Team wird vom Dienstleister gestellt, kann für spezielle Teilaufgaben aber auch Mitarbeiter des Kunden einbeziehen. In solchen Fällen sollte der Arbeitsbereich des Kunden und Dienstleisters genau definiert werden. Die Hauptaufgabe des Development Teams ist die Entwicklung von technisch einwandfreien Produkten. Um spätere Missverständnisse darüber, ob ein Produkt einwandfrei ist oder nicht zu vermeiden, empfiehlt es sich eine „Definition of Done“ in den Vertrag einzubinden. Auch sollen Beschränkungen für personelle Änderungen vertraglich geregelt werden, da diese einen erheblichen Einfluss auf die Teamdynamik und damit auf die Entwicklungsgeschwindigkeit haben können. Hier empfiehlt sich ein Zusatz, dass

Änderungen in der Teamstruktur nur in gegenseitiger Übereinstimmung vorgenommen werden dürfen. (Pieper & Roock, 2017, S. 35-36)

5.1.5 Artefakte

Das Product Backlog stellt neben dem Sprint Backlog das zentrale Element eines Scrum Prozesses dar. Der Grundaufbau des Backlogs, wie es zu führen und wer für die Pflege verantwortlich ist, sollte im Vertrag gemeinsam festgelegt werden. Zudem muss definiert werden, wann und wie die Abnahme der Produktinkremente erfolgen soll. (Pieper & Roock, 2017, S. 38-39)

5.1.6 Ereignisse

Um ein gemeinsames Verständnis des Scrum Vorgehensmodells und der chronologischen Abfolge aller Ereignisse zu schaffen, empfiehlt sich eine visuelle Darstellung des Prozesses im Vertrag. Zudem sollte man für jedes Meeting die Teilnehmer, den Zeitpunkt und die Dauer festlegen. Auch sollte eine Klausel aufgenommen werden, dass jegliche Änderungen an der Meeting Struktur (z.B. Abschaffung eines Meetings) nur im gegenseitigen Einvernehmen durchgeführt werden dürfen. (Pieper & Roock, 2017, S. 38)

Je nach Vertrauensverhältnis zum Kunden, kann ein Vertrag mehr oder weniger umfangreich ausgestaltet werden. Die oben angeführten Punkte sollten aber in jedem Fall mit aufgenommen werden, da sie von Anfang an für mehr Klarheit auf beiden Seiten sorgen und späteren Missverständnissen vorbeugen. (Pieper & Roock, 2017, S. 40)

5.2 Klassischer Festpreis

In diesem Kapitel werden die Grundzüge eines herkömmlichen Festpreisvertrages (auch klassischer Festpreis genannt), also einem Werkvertrag mit festem Preis und die damit verbundenen Probleme dargestellt. Ein Festpreisvertrag ist vor allem auf Auftraggeberseite eine beliebte Beauftragungsart, da sie aus seiner Sicht zu einer Risikominimierung führt und sich bei Ausschreibungen gut für die Auswahl des günstigsten Anbieters eignet. Die Beauftragung erfolgt dabei nach folgendem schematischen Ablauf (siehe Abbildung 26). (Opelt, et al., 2018, S. 38)



Abbildung 26: Schematischer Ablauf einer klassischen Beauftragung

Quelle: In Anlehnung an (Pieper & Roock, 2017, S. 28)

Auf Basis seiner Bedürfnisse erstellt der Auftraggeber für den Auftragnehmer ein Lastenheft mit allen Anforderungen an das zu entwickelnde Produkt. Der Auftragnehmer formuliert im Gegenzug ein Pflichtenheft, in dem er beschreibt, wie die angeführten Punkte des Lastenheftes umgesetzt werden können. Nimmt der Auftraggeber das Pflichtenheft an, startet der Auftragnehmer mit der Umsetzung. Dabei erfolgt die Gesamtlieferung des Vertragsgegenstandes in einem oder manchmal in zwei bis drei Releases. Das Ergebnis wird nach jedem Release vom Auftraggeber abgenommen, wenn die dafür vorgesehenen Anforderungen aus dem Lastenheft umgesetzt wurden. Theoretisch macht dieses Vorgehen Sinn, praktisch ist es aber mit sehr vielen Nachteilen behaftet. So kann es bei einfachen und auch komplizierten Projekten noch gut funktionieren, bei komplexen Problemstellungen aber an seine Grenzen stoßen. Grundsätzlich stimmt die Aussage, dass ein Festpreisvertrag die Kosten für ein vorher definiertes Produkt deckelt. Sehr oft ist aber das vorher definierte Produkt nicht jenes, das der Kunde später haben will. Stellt sich im Laufe des Projektes heraus, dass Anforderungen vom Kunden falsch definiert oder vergessen wurden, so muss dieser das entweder akzeptieren oder teure Change Requests anfordern, um das Produkt zu erhalten, das er benötigt. Kommt es hingegen beim Dienstleister aufgrund komplexer Problemstellung oder Missverständnisse der Anforderungen zu falschen Aufwandsschätzungen, kann dieser in den meisten Fällen seine Mehrarbeit nicht an den Kunden verrechnen. Beide Fälle bergen großes Streitpotential, führen oft zu langen Diskussionen und bieten nicht die besten Voraussetzungen für eine langfristige erfolgreiche Zusammenarbeit. Soll die Projektabwicklung agil erfolgen, ist ein klassischer Festpreis demnach nicht die beste Option. (Pieper & Roock, 2017, S. 28-50)

5.3 Agile Abwandlungen

In diesem Kapitel werden verschiedene Abwandlungen des herkömmlichen Festpreisvertrages diskutiert, die für agile Projekte angewendet werden können.

5.3.1 Klassischer Festpreis mit agilen Techniken

Wie schon in Kapitel 5.2 (Klassischer Festpreis) angeführt, ist der klassische Festpreisvertrag nicht die beste Variante für agile Projektentwicklung. Hier werden alle Steuerungsgrößen, also Kosten, Zeit und Umfang (siehe Zieldreieck, Seite 32) von Beginn an fixiert. Sollen Umfänge geändert werden, ist das bei dieser Beauftragungsform mit erheblichem Mehraufwand und Mehrkosten verbunden. Fordert der Kunde dennoch einen klassischen Festpreisvertrag, können durch Einbindung agiler Techniken so manche Vorteile generiert werden (siehe Tabelle 6). Der schematische Aufbau der Projektentwicklung bleibt erhalten, weist jedoch geringe Änderungen auf (siehe Abbildung 27). (Pieper & Roock, 2017, S. 101-102)

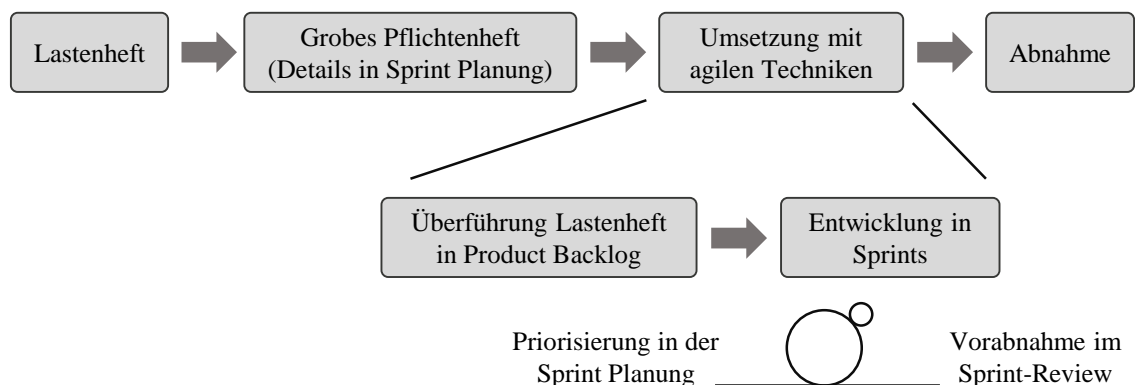


Abbildung 27: Klassischer Festpreis mit agilen Techniken

Quelle: In Anlehnung an (Pieper & Roock, 2017, S. 101)

Der Kunde liefert, wie auch beim herkömmlichen Festpreis ein Lastenheft, auf dessen Basis der Dienstleister ein Pflichtenheft erstellt. Dieses soll soweit wie möglich nur grob definiert sein, da die genauere Beschreibung der Umsetzung in der Sprint Planung und im Sprint selbst erfolgen soll. In der Umsetzungsphase wird dann das Lastenheft in ein Product Backlog übergeführt. Steht der Kunde für die Sprint Planung zur Verfügung empfiehlt sich eine gemeinsame Priorisierung der Product Backlog Elemente. Die Entwicklung wird anschließend in Sprints durchgeführt. Mittels einem Sprint-Review findet zwischen Kunde und Dienstleister abschließend eine Vorabnahme des erstellten

Produktinkrements statt. Um im Falle von notwendigen Anforderungsänderungen nicht auf schwerfällige Change Request zurückgreifen zu müssen, empfiehlt sich bei dieser Vorgehensweise zu Projektbeginn ein Mehraufwandsbudget zu definieren. Dieses kann individuell festgelegt werden und liegt in der Praxis oft bei ca. 20 % des Gesamtvolumens. Ebenfalls vorstellbar ist eine Kombination mit dem Modell „Money for Nothing, Change for Free“ (Näheres dazu im Kapitel 5.3.3). Tabelle 6 zeigt Vorteile die sich durch den Einsatz agiler Techniken ergeben. (Pieper & Roock, 2017, S. 101-106)

Vorteile agiler Techniken im Festpreis
Häufiges Feedback gibt Sicherheit das Richtige zu entwickeln
Agile Vorgehensweise ermöglicht frühe Validierung der Aufwandsabschätzung
Häufiger persönlicher Kontakt baut gegenseitiges Vertrauen auf
Mehraufwandsbudget ermöglicht schnellere Reaktion auf Änderungen

Tabelle 6: Vorteile agiler Techniken im Festpreis
Quelle: In Anlehnung an (Pieper & Roock, 2017, S. 105)

Es muss jedoch berücksichtigt werden, dass trotz der Einbindung agiler Techniken und der damit verbundenen Vorteile, dieses Vorgehen nicht die optimale Beauftragungsform für agile Projekte darstellt. (Pieper & Roock, 2017, S. 101-106)

5.3.2 Festpreis mit garantiertem Minimalumfang

Beim Konzept des Festpreises mit garantiertem Minimalumfang ist die Stellschraube nicht wie bei klassischen Festpreisen die Zeit oder die Kosten, sondern der Funktionsumfang. Dazu werden alle Anforderungen vom Kunden definiert, priorisiert und auf dieser Basis in drei Kategorien eingeteilt, den Must-have Features, Should-have Features und Could-have Features (siehe Abbildung 28). Diese Einteilung muss unbedingt vom Kunden selbst oder in Zusammenarbeit mit dem Dienstleister vorgenommen werden, da nur der Kunde den Geschäftswert der einzelnen Features bewerten kann. Die Must-have Features sollten dabei genauer definiert werden (zum Beispiel auf Basis von User Stories), bei den Should-have und Could-have Features reicht eine grobkörnige Darstellung (zum Beispiel auf Basis von Epics). Der Dienstleister führt anschließend eine Schätzung aller Features durch und garantiert bei festem Preis dem Kunden einen Minimalumfang auf Basis der Must-have Features. Kann der

Minimalumfang nicht mit dem vorher definierten festen Preis entwickelt werden, gehen die Mehrkosten zu Lasten des Dienstleisters. Können hingegen die Must-have Features mit weniger Kosten geliefert werden als geplant, werden Should-have und Could-have Features so lange entwickelt, bis der Festpreis erreicht wird. (Pieper & Roock, 2017, S. 108)

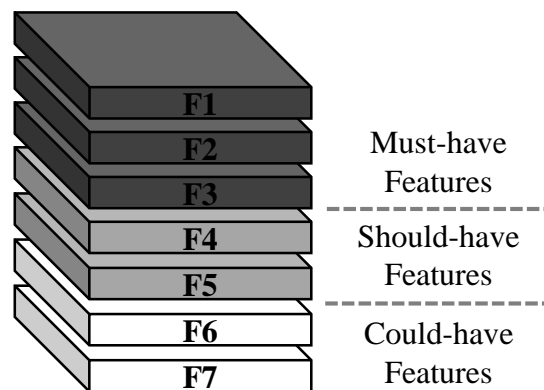


Abbildung 28: Festpreis mit garantiertem Minimalumfang
Quelle: In Anlehnung an (Pieper & Roock, 2017, S. 108)

Damit reduziert sich das Risiko auf Kunden Seite erheblich, da er in jedem Fall zu festem Preis ein nutzbares Produkt erhält. Aber auch der Dienstleister kann bei diesem Vorgehen Vorteile generieren. Zum einen reduziert sich das Schätzrisiko, zum anderen erhöht sich die Kundenzufriedenheit, wenn es dem Dienstleister gelingt, mehr als die Must-have Features zu entwickeln. Werden Anpassungen oder Änderungen der zu Beginn definierten Features vom Kunden gefordert, könnte dies mittels dem Prinzip „Change for Free“ (Näheres dazu im nachfolgenden Kapitel 5.3.3) umgesetzt werden. (Pieper & Roock, 2017, S. 108)

5.3.3 Money for Nothing, Change for Free

Das Prinzip „Money for Nothing, Change for Free“ wurde 2008 von Jeff Sutherland (Sutherland, 2008) entwickelt und hat sich seitdem in der Praxis bewährt. Wie auch beim klassischen Festpreis definiert der Kunde den gewünschten Lieferumfang in Form eines Lastenhefts und der Auftragnehmer erstellt ein Pflichtenheft und bietet das Gewerk zu einem festen Preis an. Anschließend müssen die Anforderungen in ein Product Backlog übergeführt und priorisiert werden. Der Kunde stellt dafür einen Product Owner, der diesen Vorgang verantwortet. Es empfiehlt sich, den Dienstleister, also das

Entwicklungsteam in diese Aufgabe zu integrieren. Die anschließende Entwicklung erfolgt iterativ mittels Sprints und die dabei entstehenden Produktinkremente werden dem Kunden in Sprint-Reviews vorgestellt. Fordert der Kunde neue Anforderungen, so ist dies ohne Mehraufwand möglich, wenn eine zumindest gleichwertige Anforderung aus dem Product Backlog entfernt wird (siehe Abbildung 29). (Pieper & Roock, 2017, S. 111-114)

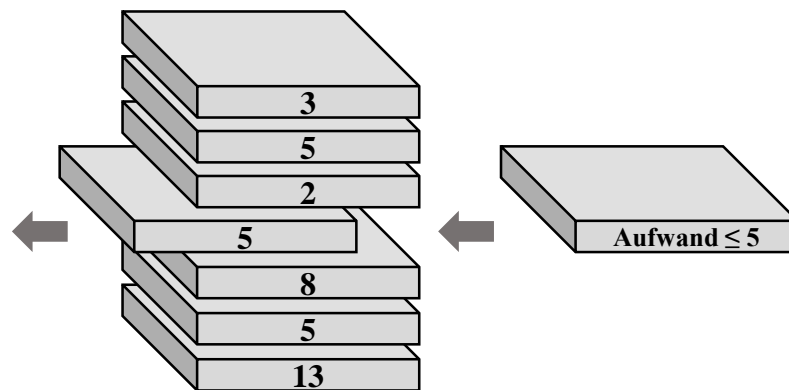


Abbildung 29: Change for Free
Quelle: In Anlehnung an (Pieper & Roock, 2017, S. 111)

Neben dem „Change for Free“ Prinzip, behandelt das Vertragsmodell auch ein vorzeitiges Beenden des Projektes. Durch die Priorisierung der Features, wird in vielen Fällen schon vor dem eigentlichen Ende des Projektes ein Produkt erstellt, dessen Weiterentwicklung sich wirtschaftlich gesehen nicht lohnt. Die kumulierte Wertentwicklung eines agilen Projekts folgt dabei einer S-Kurve und wird in Abbildung 30 dargestellt. (Pieper & Roock, 2017, S. 111-114)

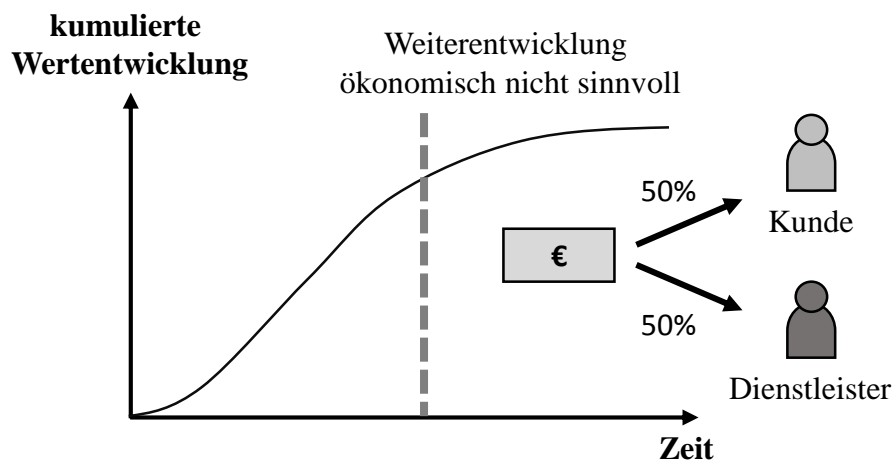


Abbildung 30: Money for Nothing
Quelle: In Anlehnung an (Pieper & Roock, 2017, S. 113)

Bei konstanter Entwicklungsgeschwindigkeit des Teams wird aufgrund der Priorisierung nach Geschäftswert, der kumulierte Wert zu Beginn einen starken Anstieg verzeichnen. Mit fortschreitendem Projekt nimmt dieser ab, da nur noch Features mit geringem Wert in das Produkt einfließen. Eine Beendigung des Projektes soll erfolgen, wenn ein nützliches Produkt vorliegt und die Weiterentwicklung ökonomisch nicht mehr sinnvoll ist. Anhand der wiederkehrenden Sprint-Reviews kann der Produkt Owner den optimalen Ausstiegspunkt leichter finden und Gebrauch von der „Money for Nothing“ Klausel machen. Dabei wird das Projekt mit dem Dienstleister beendet und der nicht verbrauchte Betrag anhand eines vorher definierten Schlüssels auf beide Parteien aufgeteilt. Pieper und Rock (Pieper & Roock, 2017) empfehlen auch bei diesem Modell ein Mehraufwandsbudget für zusätzliche Anforderungen zu Beginn des Projektes zu definieren. Zudem stellten Sie fest, dass der Projekterfolg mit der Häufigkeit des Anforderungsaustausches in einer Wechselbeziehung steht. Je öfter die „Change for Free“ Klausel angewendet wurde, desto häufiger wurden die gesteckten Ziele erreicht und nützliche Produkte entwickelt. (Pieper & Roock, 2017, S. 111-114)

5.3.4 Der agile Festpreis

In diesem Kapitel wird das letzte und auch das ausgefeilteste agile Vertragswerk, der sogenannte agile Festpreis erklärt. Dieser zeichnet sich dadurch aus, dass zum Beginn des Vertragsabschlusses der Vertragsgegenstand noch nicht im Detail bekannt ist. Es muss aber ein guter Überblick vorhanden sein, was der Auftraggeber als Resultat des Projektes erhalten möchte. Einerseits muss die Beschreibung für einen internen Business Case des Kunden ausreichen, andererseits muss auch der Dienstleister ein grobes Bild der Gesamtanforderungen haben. Um den Überblick über den doch sehr umfangreichen Prozess des agilen Festpreises zu wahren, erfolgt eine Unterteilung in vier Schritten.

Schritt 1- Vertragsgegenstand festlegen

In einem ersten Schritt wird der Vertragsumfang gemäß der Produktvision anhand von Epics definiert. Dieses Vorgehen soll das Produkt vollständig, aber vorerst nur sehr grob beschreiben. Abbildung 31 zeigt wie bei der Detaillierung der Vision vorzugehen ist. (Opelt, et al., 2018, S. 49-53)

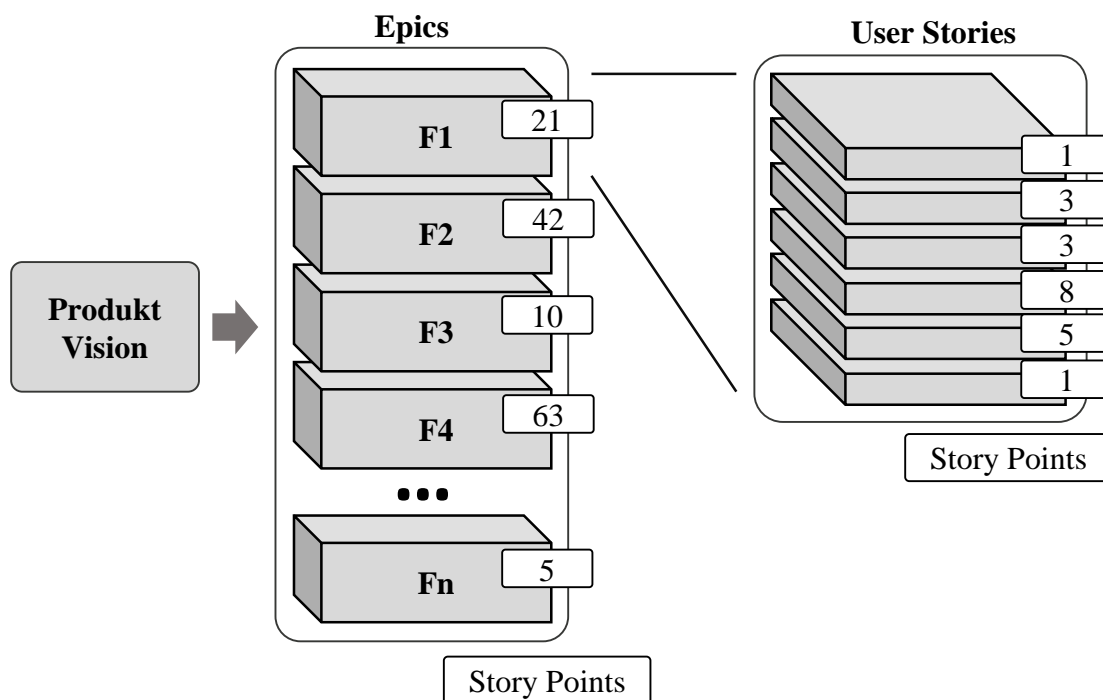


Abbildung 31: Detaillierung der Produktvision beim agilen Festpreis
 Quelle: In Anlehnung an (Opelt, et al., 2018, S. 53)

Schritt 2 – Detailspezifikation

Anschließend wird ein repräsentatives Epic (je nach Projektgröße auch zwei bis drei) bis auf die Ebene der User Stories verfeinert (siehe Abbildung 31). Sowohl die Definition der Epics, als auch der User Stories sollte der Kunde (Product Owner) in Kooperation mit dem Dienstleister (Development Team) durchführen, um schon von Beginn an ein gemeinsames Verständnis für das Produkt zu entwickeln. Wurden alle Epics und User Stories definiert, müssen zunächst vom Development Team die Aufwände der User Stories mittels Story Points abgeschätzt werden. Hierbei soll man auf eine konsensbasierte relative Aufwandsschätzung wie „Magic Estimation“ oder „Planning Poker“ zurückgreifen. Im oben angeführten Beispiel (siehe Abbildung 31) konnten für Epic-F1 sechs User Stories mit einem Gesamtaufwand von 21 Story Points gefunden werden. (Opelt, et al., 2018, S. 49-57)

Schritt 3 – Gesamtaufwand

Anhand dieses Referenzwertes können dann in einem nächsten Schritt die Aufwände für die restlichen Epics abgeschätzt werden. Dies erfolgt innerhalb eines gemeinsamen

Workshops, zwischen Kunde und Dienstleister. Dabei ergeben sich für beide Seiten folgende Vorteile (Opelt, et al., 2018, S. 57):

- *„Der Kunde versteht, an welchen Stellen der Dienstleister die Komplexität sieht.*
- *Beide Parteien verstehen, wie die jeweils andere die geschriebene Anforderung interpretiert.*
- *Zudem können im Zuge des Workshops die Beschreibungen der Epics ergänzt und erweitert werden, um Unklarheiten schon von Beginn an zu eliminieren.“* (Opelt, et al., 2018, S. 57)

Als Resultat der Schätzung erhält man für das Projekt eine Gesamtanzahl an Story Points. Opelt, Gloger, Pfarl & Mittermayer (Opelt, et al., 2018) empfehlen im Zuge der Aufwandsabschätzung zu jedem Epic und jeder User Story auch den Geschäftswert und das Umsetzungsrisiko mit zu schätzen. Diese machten häufig die Erfahrung, dass sich ohne diese Transparenz zwischen Kunde und Dienstleister die Priorisierung und Projektsteuerung in eine falsche Richtung entwickelt hat. Da der Kunde bei einem Festpreisprojekt an den Gesamtkosten interessiert ist, erfolgt noch eine letzte Schätzung. Dabei wird ein Umrechnungsfaktor von Story Points auf Teamkosten von einem Team bestehend aus erfahrenen Projektleitern und Entwicklern (nicht dem Development Team) ermittelt. Hier muss angemerkt werden, dass dieser Umrechnungsfaktor einen initialen Wert darstellt, welcher sich im Laufe des Projektes bei veränderter Team Velocity schnell ändern kann. Bevor der Festpreis final definiert wird, muss noch ein Sicherheitsaufschlag auf Basis der Komplexität und benötigten Qualität des Liefergegenstandes und der Expertise des Dienstleisters festgelegt werden. Nach Klärung all dieser Punkte liegt ein indikativer Festpreis (nicht vertraglich bindend) vor. Im Unterschied zum herkömmlichen Festpreis wird dieser von Dienstleister und Kunde gemeinsam ermittelt und verstanden. (Opelt, et al., 2018, S. 57-59)

Schritt 4 – Definition von Riskshare, Checkpoint-Phase und Ausstiegspunkte

In diesem Schritt wird ein Riskshare-Modell festgelegt. Dabei wird ein Schlüssel definiert, wie Mehrkosten im Falle von Überschreitung des Festpreisrahmens aufgeteilt werden. Dies soll einerseits die Kooperation beider Parteien unterstützen, da bei Problemen beide zur Kasse gebeten werden und andererseits, lange Diskussionen im vornhinein ausschließen. Fordert der Kunde Anforderungsänderungen so eignet sich

dafür das „Change for Free“ Modell (siehe Kapitel 5.3.3). Der Kunde hat hier die Möglichkeit, immer vor Definition des nächsten Sprint Backlogs Anforderungen auszutauschen, wenn diese denselben Aufwand aufweisen. (Opelt, et al., 2018, S. 59-61)

Neben einem Riskshare-Modell muss auch eine Checkpoint-Phase definiert werden. In dieser Phase vereinbaren beide Parteien, bis zu einem vorher festgelegten Zeitpunkt (meist 2-5 Sprints) zusammenzuarbeiten. Am Ende der Checkpoint-Phase angekommen, können sich Kunde und Dienstleister über eine Fortsetzung des Projektes abstimmen. Im Falle des Projektabbruchs muss festgelegt werden, wie das bis dahin erstellte Produkt vergütet wird. Entscheidet man sich aber dafür, das Projekt fortzusetzen werden alle getroffenen Annahmen und Schätzungen mittels den gewonnenen Erfahrungen angepasst. In den meisten Fällen wird auch der Maximalpreis nochmals nachgeschärft. Zuletzt sollten noch zum Schutz beider Parteien weitere Ausstiegspunkte neben der Checkpoint-Phase festgelegt werden. (Opelt, et al., 2018, S. 59-64)

6 Forschungsfragen

In diesem Kapitel werden die Forschungsfragen (vgl. Kapitel 1.2 Zielsetzung und Forschungsfragen) nochmals angeführt und näher beschrieben. ALTEN agiert in einem dynamischen Markt mit sich häufig ändernden Anforderungen und hat die Problematik der Projektabwicklung mit einem starren, klassischen Projektmanagement erkannt. Im agilen Projektmanagement sieht ALTEN das Potential, diese Probleme zu adressieren. Auf Basis der initialen Fragestellung, wie sich agile Ansätze erfolgreich in bestehende Prozesse und Projekte implementieren lassen, wurden fünf aufeinander aufbauende Forschungsfragen abgeleitet:

- F1:** *Worin liegen für ALTEN die Hauptprobleme bei klassischer Projektabwicklung?*
- F2:** *Worin sieht ALTEN die Vorteile von agiler Projektabwicklung?*
- F3:** *Welche Grundvoraussetzungen müssen für eine agile Projektabwicklung gegeben sein?*
- F4:** *Welche Anpassungen und Änderungen müssen für ein agiles Projekt in Bezug auf die unterschiedlichen IQP-Modelle getroffen werden, damit dieses erfolgreich abgewickelt werden kann?*
- F5:** *Wie könnte eine agile Projektabwicklung mit Scrum für ein Festpreisprojekt (IQP-5) aussehen und welche Vertragsinhalte müssen dabei berücksichtigt werden?*

Die erste Forschungsfrage beschäftigt sich mit den Hauptproblemen bei klassischer Projektabwicklung, also all jenen Faktoren, die einen negativen Einfluss auf den Projekterfolg haben.

Forschungsfrage zwei behandelt die Vorteile von agiler Projektabwicklung und beschäftigt sich mit den positiven Einflussfaktoren für den Projekterfolg.

Bei der dritten Forschungsfrage werden Grundvoraussetzungen für die unterschiedlichen Zusammenarbeitsmodelle (IQP-3 bis IQP-5) gesucht, die eine agile Projektabwicklung erst möglich machen und damit eine bessere Alternative zur klassischen Projektabwicklung darstellen.

Welche ALTEN-internen Anpassungen bzw. Änderungen anschließend vorgenommen werden müssen, um bei den unterschiedlichen IQP-Modellen ein agiles Projekt erfolgreich abwickeln zu können, untersucht Forschungsfrage vier.

Eine agile Projektabwicklung bei Festpreisprojekten zeigt sich als große Herausforderung (vgl. Kapitel 5.2 Klassischer Festpreis). Forschungsfrage fünf widmet sich daher der Exploration eines allgemeinen Prozesses und Best Practices in Bezug auf die Vertragsgestaltung (siehe Kapitel 5 Agile Vertragsgestaltung), um auch solche Projekte erfolgreich zu managen.

Die Grundlage zur Beantwortung oben angeführter Forschungsfragen wird aus einem Mix von Literaturrecherche und firmeninternen Experteninterviews gebildet. Wie die Datenerhebung und nachfolgende Datenauswertung durchgeführt wurde, wird im nächsten Kapitel im Detail beschrieben.

7 Methodisches Vorgehen

In diesem Kapitel wird die methodische Vorgehensweise der Datenerhebung (Interviews) und der anschließenden Datenauswertung (qualitative Inhaltsanalyse nach Mayring) beschrieben.

7.1 Datenerhebung

Die Datenerhebung erfolgte mittels leitfadengestützten Interviews in vier Schritten: Erstellung des Interviewleitfadens, Auswahl geeigneter Experten, Interviewdurchführung und Transkription. Diese vier Schritte werden im Folgenden kurz beschrieben.

7.1.1 Interviewleitfaden

Als Datenerhebungsinstrument wurde wie schon oben angeführt, ein leitfadengestütztes Interview verwendet. Dieses zählt zu den qualitativen Methoden der Sozialforschung und kam in seiner einfachsten Ausführung, dem semistrukturierten Experteninterview zur Anwendung. Bei dieser Form des Experteninterviews werden für jedes Interview vorgefertigte Fragestellungen verwendet, deren Beantwortung offen ist, also keinen Einschränkungen unterliegen. Zudem können mehrere Themenbereiche behandelt werden und eine Improvisation bzw. Abweichung von den Fragen ist ebenfalls möglich. Da es sich bei den zugrunde liegenden Forschungsfragen um einen Mix aus firmeninternen und allgemeinen Fragestellungen im Bereich des klassischen und agilen Projektmanagements handelt, kann diese Interviewform als zweckmäßig eingestuft werden. (Helfferich, 2005)

Auf Basis der vorher durchgeführten Literaturrecherche und der zu beantwortenden Forschungsfragen wurde das Interview in 5 Abschnitte geteilt:

- A *Hauptprobleme bei klassischer Projektabwicklung*
- B *Grundvoraussetzungen für agile Projektabwicklung*
- C *Eignung der unterschiedlichen Zusammenarbeitsmodelle*
- D *Notwendige Anpassungen um agilem Vorgehen zu genügen*
- E *Allgemeiner Scrum basierter Prozess*

Zu jeder dieser übergeordneten Kategorien wurden anschließend geeignete Fragestellungen erarbeitet, um die gewünschten Auskünfte zu den Forschungsfragen zu erhalten (gesamter Interviewleitfaden mit allen Unterfragen, siehe Anhang A). Großer Wert wurde dabei auf eine offene und vor allem verständliche Fragestellung gelegt.

7.1.2 Auswahl der Experten

Um eine ausreichend große Datenmenge zu generieren, aber auch innerhalb des vorgegebenen Rahmens einer Masterarbeit zu bleiben, wurden in Summe sechs Experteninterviews durchgeführt. Bei den Befragten handelt es sich um leitende Angestellte unterschiedlichen Alters und unterschiedlicher Standorte. Dabei wurde jedenfalls darauf geachtet, dass jeder befragte Experte mehrere Jahre in dem für die Masterarbeit relevanten Forschungsgebiet an ausreichender Erfahrung verfügt.

7.1.3 Durchführung des Interviews

Der Großteil der durchgeführten Interviews erfolgte persönlich, für den Rest wurde auf telefonische Interviews zurückgegriffen. Zu Beginn wurde den Befragten das Thema, die Problemstellung und das Ziel der Masterarbeit mitgeteilt. Dann erfolgte ein kurzer Überblick zu den fünf Interviewabschnitten (siehe Interviewleitfaden, Seite 63). Zur Wahrung der Anonymität erfolgte anschließend eine Information, dass jegliche Gesprächsinhalte des Interviews ohne Angaben von Namen und Unternehmungen ausgewertet werden. Abschließend wurde noch die Zustimmung zur digitalen Aufzeichnung des Interviews mittels Diktiergerät eingeholt. Die durchschnittliche Interviewlänge lag bei etwa 60 Minuten. Das erste Interview diente als „Pretest“ für den Interviewleitfaden. Dabei erfolgte am Ende des Interviews eine Befragung des Experten zu Aufbau, Verständlichkeit und Dauer des Interviews. Dieser nannte wenige Kritikpunkte, somit konnte der Interviewleitfaden in seiner Gesamtheit mit nur kleinen Adaptionen für die restlichen Interviews verwendet werden.

7.1.4 Transkription

Im letzten Schritt der Datenerhebung erfolgte die sogenannte Transkription. Darunter versteht man die Verschriftung digitaler Gesprächsprotokolle nach festgelegten Regeln

(Deppermann, 2013, S. 39). In der vorliegenden Masterarbeit erfolgte eine vollständige und wörtliche Transkription aller Interviews, lediglich Dialektäußerungen wurden durch geeignete hochdeutsche Wörter ersetzt. Dabei wurde stets darauf geachtet, keine Veränderungen am Kontext des Satzes vorzunehmen. Das hier entstandene Transkript konnte anschließend als Grundlage zur nachfolgenden Datenauswertung verwendet werden. Der Interviewer wurde dabei mit „I:“, die Befragten mit „B1“ bis „B6“ bezeichnet, wobei sich die Nummerierung durch die chronologische Reihenfolge der Interviews ergab.

7.2 Datenauswertung

Die Auswertung des Datenmaterials, also der Transkripte der Experteninterviews, erfolgte mittels qualitativer Inhaltsanalyse nach Mayring. Welche konkrete Technik zur Anwendung kam und wie dabei vorgegangen wurde, zeigen die folgenden Kapiteln.

7.2.1 Qualitative Inhaltsanalyse

Unter einer qualitativen Inhaltsanalyse versteht man eine Methode zur systematischen Textanalyse, um Textmaterial in Form von Transkripten auf eine Fragestellung hin regelgeleitet und nachvollziehbar zu interpretieren und auszuwerten (Mayring, 2002). Dabei werden jene für die Forschungsfrage relevanten Textpassagen schrittweise und streng theoriegeleitet in ein Kategoriensystem eingeordnet (Baur & Blasius, 2014, S. 544). Das Kategoriensystem ermöglicht dabei eine systematische Reduktion des Datenmaterials auf die Kernaussagen und stellt somit das Hauptelement der qualitativen Inhaltsanalyse dar. Als Ergebnis erhält man verschiedene Kategorien, gefüllt mit den relevanten Inhalten aller Interviews. Mayring nennt zwei verschiedene Ansätze, wie die Kategorienbildung erfolgen kann: die induktive Kategorienentwicklung und die deduktive Kategorienanwendung (Baur & Blasius, 2014, S. 547-548). Bei der induktiven Kategorienentwicklung werden anhand des vorliegenden Textmaterials Kategorien schrittweise entwickelt. Dabei werden für die Forschungsfrage relevante Textstellen zu Kategorien abstrahiert, oder bereits entwickelten Kategorien zugeordnet. Bei der deduktiven Kategorienanwendung hingegen, werden zur Einordnung relevanter Textpassagen vorher festgelegte Kategorien verwendet, die zum Beispiel aus der Literatur gewonnen wurden. Der große Unterschied liegt darin, dass bei induktivem

Vorgehen Kategorien erst gebildet und bei deduktiven Vorgehen Kategorien schon vordefiniert werden müssen. Da es sich bei den vorliegenden Forschungsfragen hauptsächlich um ALTEN-interne Fragestellungen handelt, war eine Kategoriendefinition vorab nicht möglich. Es wurde deshalb die qualitative Inhaltsanalyse mit induktiver Kategorienentwicklung verwendet. Im nachfolgenden Kapitel wird das Vorgehen der qualitativen Inhaltsanalyse mit induktiver Kategorienentwicklung anhand eines speziell dafür entwickelten Ablaufmodells näher beschrieben. (Baur & Blasius, 2014, S. 543ff)

7.2.2 Ablaufmodell

Abbildung 32 zeigt das Ablaufmodell für induktive Kategorienbildung.

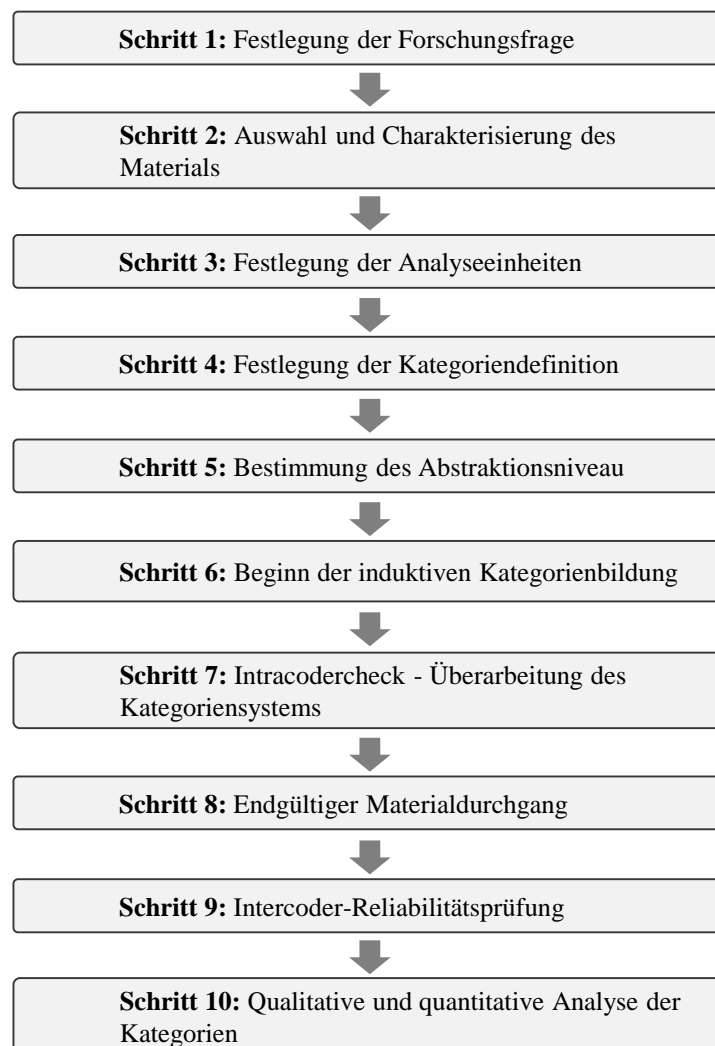


Abbildung 32: Ablaufmodell der qualitativen Inhaltsanalyse mit induktiver Kategorienbildung
Quelle: In Anlehnung an (Baur & Blasius, 2014, S. 550)

Der Grundgedanke des Ablaufmodells liegt darin, eine nachvollziehbare, intersubjektiv überprüfbare und streng regelgeleitete Analyse zu erhalten (Mayring, 2002). Bevor man mit der Analyse jedoch starten kann, müssen zuerst in Bezug auf die jeweilige Fragestellung, Rahmenbedingungen für die Methode festgelegt werden. Wie dies für die vorliegende Masterarbeit erfolgte, wird anhand der in Abbildung 32 angeführten 10 Schritte im Folgenden erklärt.

Schritt 1

Das Ablaufmodell nennt in einem ersten Schritt die Festlegung aller relevanten Forschungsfragen. Diese wurden schon zu Beginn der Masterarbeit definiert und werden zum besseren Überblick nachfolgend nochmals angeführt:

- F1:** *Worin liegen für ALTEN die Hauptprobleme bei klassischer Projektabwicklung?*
- F2:** *Worin sieht ALTEN die Vorteile von agiler Projektabwicklung?*
- F3:** *Welche Grundvoraussetzungen müssen für eine agile Projektabwicklung gegeben sein?*
- F4:** *Welche Anpassungen und Änderungen müssen für ein agiles Projekt in Bezug auf die unterschiedlichen IQP-Modelle getroffen werden, damit dieses erfolgreich abgewickelt werden kann?*

Forschungsfrage fünf wird in oben genannter Auflistung nicht erwähnt, da diese vorrangig mittels Literaturrecherche beantwortet werden soll und daher für die qualitative Inhaltsanalyse wenig Relevanz aufweist.

Schritt 2

Anhand der Forschungsfragen wurden firmeninterne, leifadengestützte Interviews als geeignetes Untersuchungsmaterial ausgewählt. Dabei wurden sechs semistrukturierte Experteninterviews durchgeführt und vollständig transkribiert.

Schritt 3

Zur Präzisierung der Inhaltsanalyse erfolgte anschließend die Festlegung der Analyseeinheiten. Diese werden im Folgenden genannt und kurz erklärt:

- Als *Kodiereinheit* (kleinster zu kategorisierender Textbestandteil) wurde die sinntragende Phrase festgelegt.

- Die *Kontexteinheit* (größter Textbestandteil einer Kategorie) wurde mit dem einzelnen Interview definiert.
- Als *Auswertungseinheit* (welche Texte nacheinander ausgewertet werden) wurde das gesamte Textmaterial, also alle Interviews festgelegt. (Mayring, 2010, S. 59)

Schritt 4 - 5

Bevor die eigentliche Arbeit am Textmaterial beginnen kann, sind als weitere vorbereitende Schritte die Kategoriendefinition und die Bestimmung des Abstraktionsniveaus notwendig.

- Die *Kategoriendefinition*, auch Selektionskriterium genannt, beschreibt, welche Textstellen im Material kategorisiert werden dürfen.
- Das *Abstraktionsniveau* legt fest, auf welcher Verallgemeinerungsebene die Kategorien sprachlich formuliert werden. (Baur & Blasius, 2014, S. 547)

Diese beiden Schritte wurden für jede Forschungsfrage (F1 – F4) separat durchgeführt und werden in nachfolgender Tabelle dargestellt (siehe Tabelle 7):

Fragestellung (F)	Kategoriendefinition (K) / Abstraktionsniveau (A)
F1: Worin liegen für ALTEN die Hauptprobleme bei klassischer Projektabwicklung?	K: Alle Hinweise auf negative Einflussfaktoren auf den Projekterfolg bei klassisch gemanagten Projekten
	A: Konkret beschreibbare negative Einflussfaktoren
F2: Worin sieht ALTEN die Vorteile von agiler Projektabwicklung?	K: Alle Hinweise auf positive Einflussfaktoren auf den Projekterfolg bei agiler Projektabwicklung
	A: Konkret beschreibbare positive Einflussfaktoren
F3: Welche Grundvoraussetzungen müssen für eine agile Projektabwicklung gegeben sein?	K: Alle Hinweise auf mögliche Voraussetzungen die es für eine agile Projektabwicklung benötigt
	A: Konkret beschreibbare Voraussetzungen für agile Projektabwicklung
F4: Welche Anpassungen bzw. Änderungen müssen für ein agiles Projekt in Bezug auf die unterschiedlichen IQP-Modelle getroffen werden, damit dieses erfolgreich abgewickelt werden kann?	K: Alle Hinweise auf notwendige oder sinnvolle Anpassungen bzw. Änderungen, damit ein agiles Projekt erfolgreich abgewickelt werden kann
	A: Konkrete Vorschläge oder Ideen zu Verbesserungen, Anpassungen und Änderungen

Tabelle 7: Kategoriendefinition und Abstraktionsniveau

Quelle: Eigene Darstellung

Schritt 6 - 8

Zur Unterstützung der induktiven Kategorienbildung wurde das webbasierte Programm „QCAmap“ verwendet. Dazu wurden alle sechs Interviews als Textdateien in das Programm hochgeladen und die Forschungsfragen, die Analyseeinheiten, die Kategoriendefinition und das Abstraktionsniveau, wie in den Schritten 1-5 gezeigt, definiert. Anschließend erfolgte eine schrittweise induktive Kategorienbildung für die ausgewählten Forschungsfragen in Bezug auf Definition und Abstraktionsniveau.

I: Was sind deiner Meinung nach negative Einflussfaktoren für den Projekterfolg für klassisch gemanagte Projekte?

B1: Negativer Einfluss ist, dass man sehr oft unflexibel ist. Man hat ein relativ starres Planungsgerüst, wo man halt einen Anfang und ein Ende hat und wo man dann, die Dinge oft versucht, parallel zu machen und am Schluss manches Mal, Themen nicht zusammenpassen. Also, da jeder so seinen eigenen Plan verfolgt, aber am Schluss dann feststellt, dass Dinge nicht zusammenpassen oder zeitlich komplett aus dem Rahmen kommen, weil man halt oft sehr lange Phasen hat, in so einer klassischen Projektplanung...

K1

K2

Abbildung 33: Kategorisierung mit QCAmap

Quelle: Eigene Darstellung

Abbildung 33 zeigt die Kategorienbildung mittels des Programms QCAmap anhand eines Textausschnittes der Interviews. Relevante Textpassagen werden markiert und auf Basis des Inhaltes, entweder einer neu entwickelten Kategorie (hier K1: Unflexibel aufgrund starren Planungsgerüsts) oder einer schon vorhandenen Kategorie zugewiesen. Diese werden mit einer fortlaufenden Nummer (K1, K2, ...) abgekürzt. Der Vorgang wurde für jede Forschungsfrage und dem gesamten Textmaterial durchgeführt. Nach der Hälfte des Materialdurchgangs erfolgte der sogenannte „Intracodercheck“ (Schritt 7), welcher die Stabilität der Analyse gewährleisten soll. Dabei wurden Ausschnitte des Textmaterials erneut kodiert und den schon vorher erstellten Kategorien gegenübergestellt. Gab es Abweichungen wurden die Kategorien angepasst, wenn notwendig zusammengefasst oder auch gelöscht. Hierbei wurde großes Augenmerk daraufgelegt, dass alle Kategorien demselben Abstraktionsniveau entsprechen und klar voneinander abgegrenzt werden können. Nach dieser Zwischenüberprüfung erfolgte der endgültige Materialdurchgang (Schritt 8). (Baur & Blasius, 2014, S. 546)

Schritt 9 - 10

Zur Überprüfung der Zuverlässigkeit und Reproduzierbarkeit der Inhaltsanalyse erfolgte in einem nächsten Schritt die „Intercoder-Reliabilitätsprüfung“, welche vom Betreuer der Masterarbeit durchgeführt wurde. Hierbei wurden alle Kategorien auf eine verständliche Formulierung, sinnvolle Kategorienentwicklung und nachvollziehbare Kategorienzuordnung zu den ausgewählten Textpassagen überprüft. Auf Grundlage der Intercoder-Reliabilitätsprüfung erfolgte eine abschließende Überarbeitung aller entwickelten Kategorien und Kategorienzuordnungen. Basierend auf den gewonnenen Kategorien wurde in einem weiteren Schritt für jede Forschungsfrage ein Kategoriensystem in Anlehnung an die *Gioia-Methode* (Gioia & Corley, 2004) gebildet. Bei diesem Vorgehen erfolgte eine schrittweise Zusammenfassung sinnähnlicher Kategorien zu Über- und Hauptkategorien, um die Ergebnisse besser interpretieren zu können. Im Gegensatz zu der streng regelgeleiteten Entwicklung von Kategorien gibt es zur Bildung von Über- und Hauptkategorien kein Regelwerk und somit mehr Raum für Interpretationen. (Baur & Blasius, 2014, S. 546-547)

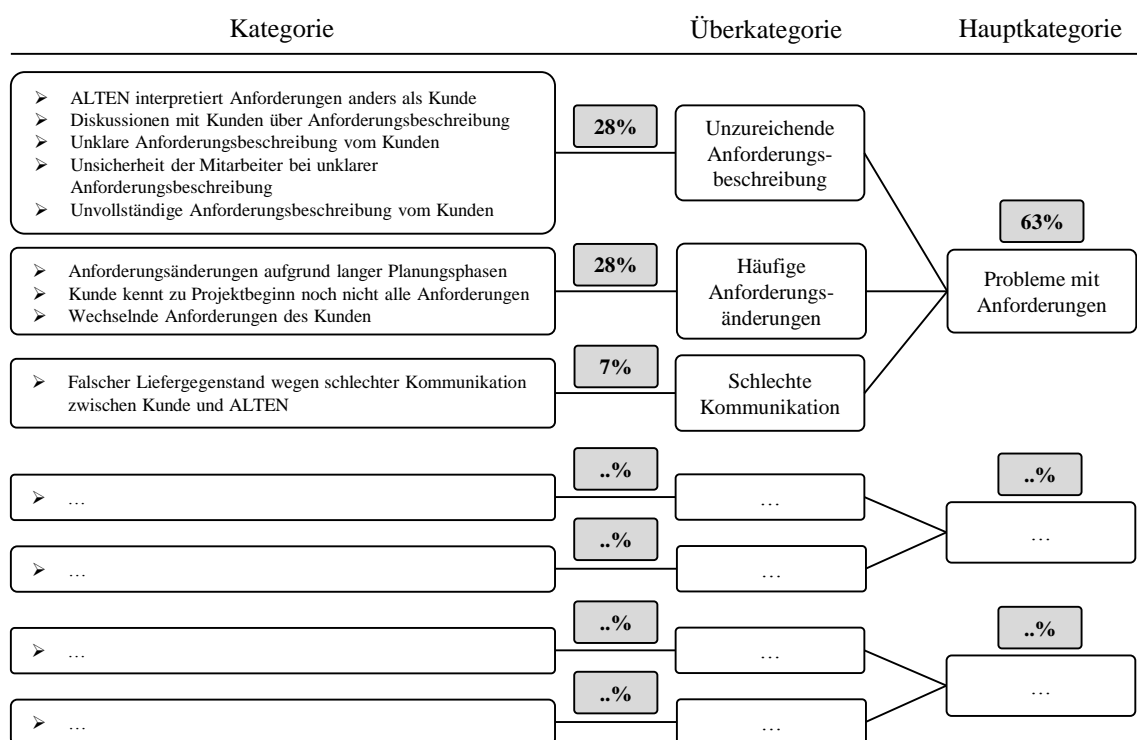


Abbildung 34: Beispiel zu einem Kategoriensystem
Quelle: Eigene Darstellung

Zudem erfolgte eine quantitative Auswertung des gesamten Kategoriensystems. Dabei wurden die Nennungen der einzelnen Kategorien gezählt und auf Basis dieser Daten relative Häufigkeiten für Über- und Hauptkategorien berechnet. Die quantitative Auswertung soll dabei helfen, die Relevanz der gefundenen Kategorien für die betrachtete Forschungsfrage besser einschätzen zu können.

Abbildung 34 zeigt einen Ausschnitt des entwickelten Kategoriensystems für Forschungsfrage eins und soll im Folgenden kurz erklärt werden. Die linke Spalte zeigt die aus dem Textmaterial entwickelten Kategorien. Sinnähnliche Kategorien können dabei in Form von Überkategorien (Spalte zwei) dargestellt werden. Diese Überkategorien können in einem letzten Schritt zu Hauptkategorien (Spalte drei) aggregiert werden. Die in grau hinterlegten Prozentzahlen geben dabei die relative Häufigkeit der Nennung von Über- und Hauptkategorien wieder. Der ersten Überkategorie (Unzureichende Anforderungsbeschreibung) können beispielsweise 28 % aller entwickelten Kategorien zugeordnet werden.

8 Ergebnisse

Die Darstellung der aus der qualitativen Inhaltsanalyse und Literaturrecherche erhaltenen Ergebnisse erfolgt in gleicher Reihenfolge wie die in Kapitel 6 angeführten Forschungsfragen.

Zunächst werden die gefundenen Hauptprobleme für ALTEN bei klassischer Projektabwicklung (siehe Kapitel 8.1) behandelt. Kapitel 8.2 zeigt die Vorteile agiler Projektabwicklung aus der Perspektive der befragten Experten. Welche Grundvoraussetzungen für agile Projektabwicklung vorhanden sein müssen, wird in Kapitel 8.3 dargestellt. Ideen aus Sicht der Experten für mögliche Anpassungen beziehungsweise Änderungen für zukünftige agile Projekte, behandelt Kapitel 8.4. Im abschließenden Kapitel 8.5 werden Best Practices in Bezug auf die Vertragsgestaltung und Projektdurchführung für Festpreisprojekte dargestellt. Wie schon in Kapitel 7.2.2 (Ablaufmodell, Seite 67) angeführt, wurden die Ergebnisse zu den Forschungsfragen eins bis vier hauptsächlich aus den Expertenbefragungen und nachfolgender Inhaltsanalyse abgeleitet. Die erarbeiteten Kategoriensysteme dienen hierbei als Interpretationsgrundlage. Die Ergebnisse zu Forschungsfrage fünf hingegen, basieren zum größten Teil auf Literaturrecherche.

8.1 Hauptprobleme bei klassischer Projektabwicklung

Dieses Kapitel beschreibt die Hauptprobleme bei klassischer Projektabwicklung aus der Perspektive von ALTEN und zielt auf die Beantwortung der ersten Forschungsfrage (siehe Kapitel 6) ab. Mittels der qualitativen Inhaltsanalyse konnten in Summe 20 Kategorien, 8 Überkategorien und 3 Hauptkategorien identifiziert werden. Abbildung 35 zeigt das Kategoriensystem dazu.

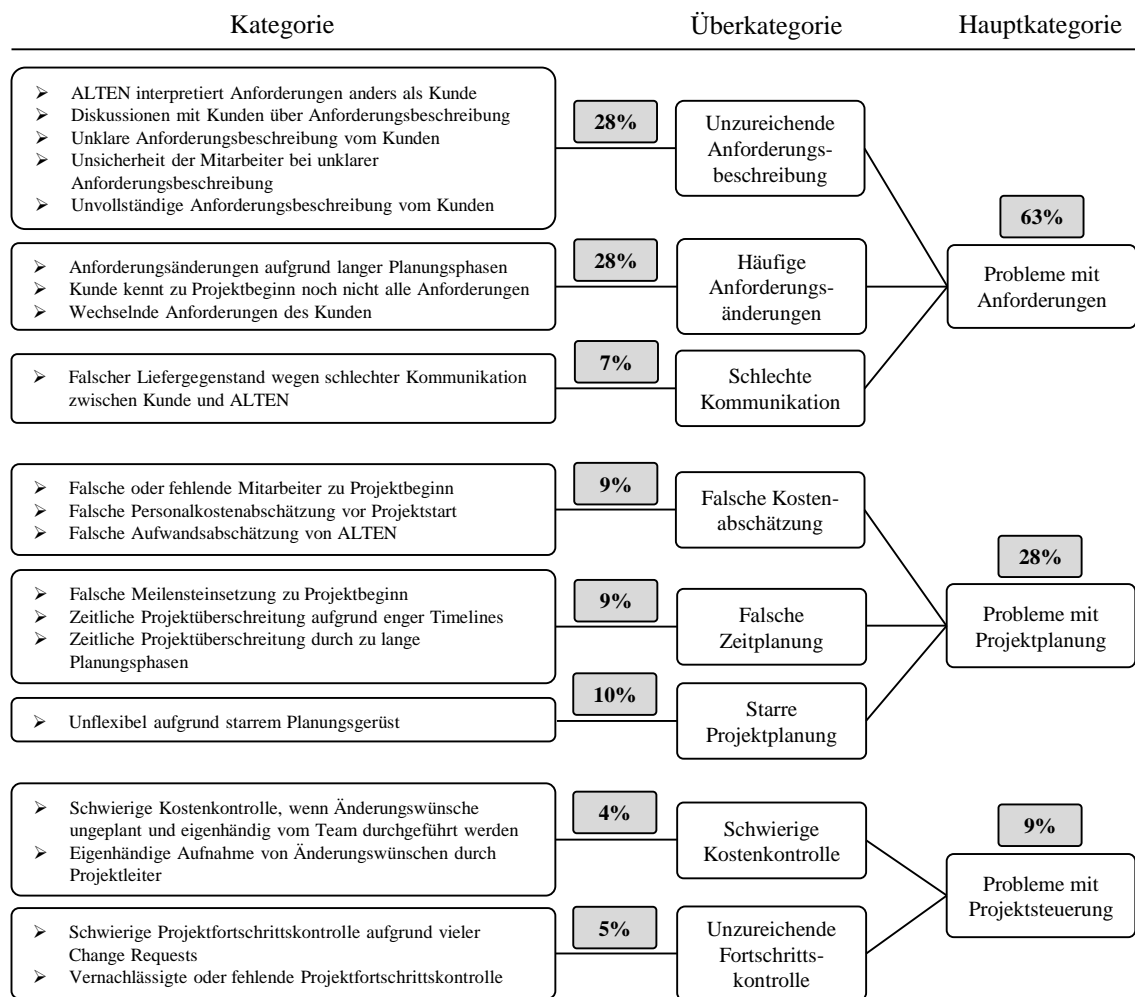


Abbildung 35: Kategoriensystem - Forschungsfrage 1
Quelle: Eigene Darstellung

Mit einer relativen Häufigkeit von 63 % zeigen *Probleme mit Anforderungen* den mit Abstand größten negativen Einflussfaktor für klassisch gemanagte Projekte. *Unzureichende Anforderungsbeschreibung* (28 %), *häufige Anforderungsänderungen* (28 %) und *schlechte Kommunikation zwischen ALTEN und Kunden* (7 %) bilden hierbei die Überkategorien.

Abbildung 36 enthält Auszüge aus den Experteninterviews, um oben genannte Punkte zu untermauern:

Hauptkategorie	Zitat
Probleme mit Anforderungen	<i>„Änderungen der Anforderungen. Wenn sich während dem Projekt sich Anforderungen ändern, dann ist das immer ein Problem, was mach ich jetzt damit.“</i>
	<i>„Also das Problem, was wir oft bei Projekten haben, ist, dass das Lastenheft oft sehr unklar definiert ist.“</i>
	<i>„Kommunikation ist aus meiner Sicht das A und O. Kommunikation zwischen Kunde und der Firma, wenn die nicht passt, wenn der Kunde nicht kommuniziert oder selber nicht bewusst ist, was er letztendlich an Output haben will.“</i>

Abbildung 36: Zitate der Experteninterviews zu Forschungsfrage 1

Diese von den Experten abgegebene Einschätzung deckt sich weitestgehend mit den in der Literatur gefundenen Misserfolgskategorien (siehe 3.2 Gründe für nicht erfolgreiche Projekte, Seite 35-37). Auch hier werden schlechte Kommunikation (bzw. fehlende Zusammenarbeit), unvollständige bzw. unklare Anforderungen und häufige Anforderungsänderungen als die größten negativen Faktoren beschrieben.

Neben den hauptsächlich extern, also vom Kunden verursachten Anforderungsproblemen, konnten auch interne *Probleme mit der Projektplanung* und der *Projektsteuerung* gefunden werden. Eine *falsche Kosten- und Zeitplanung* sowie *starre Projektplanung* werden hier mit einer kumulierten relativen Häufigkeit von 28 % angeführt. Eine *schwierige Kosten- und Fortschrittskontrolle* weist mit einer Summe von 9 % den geringsten Wert auf. Es muss jedoch angeführt werden, dass es hier zu einem Ursache-Wirkungs-Konflikt kommt. Betrachtet man die entwickelten Kategorien der Hauptkategorie *„Probleme mit Projektsteuerung“* so fällt auf, dass 75 % all dieser in direktem Zusammenhang mit Änderungswünschen stehen. Änderungswünsche allerdings, stehen in direktem Kontakt zu Anforderungsänderungen, welche der ersten Hauptkategorie *„Probleme mit Anforderungen“* zugerechnet werden. Auch bei Hauptkategorie zwei *„Probleme mit Projektplanung“* können vereinzelt Kategorien gefunden werden, die einen kausalen Zusammenhang mit *Anforderungen* aufweisen. Dies

verdeutlicht jedenfalls die Problematik mit *unzureichenden Anforderungsbeschreibungen* und *häufigen Anforderungsänderungen*.

8.2 Vorteile von agiler Projektentwicklung

In diesem Kapitel werden die Vorteile von agiler Projektentwicklung diskutiert. In Summe wurden dabei 26 Kategorien zu 8 Überkategorien und diese zu 3 Hauptkategorien zusammengefasst. Abbildung 37 zeigt dazu das entwickelte Kategoriensystem für Forschungsfrage zwei (siehe Kapitel 6).

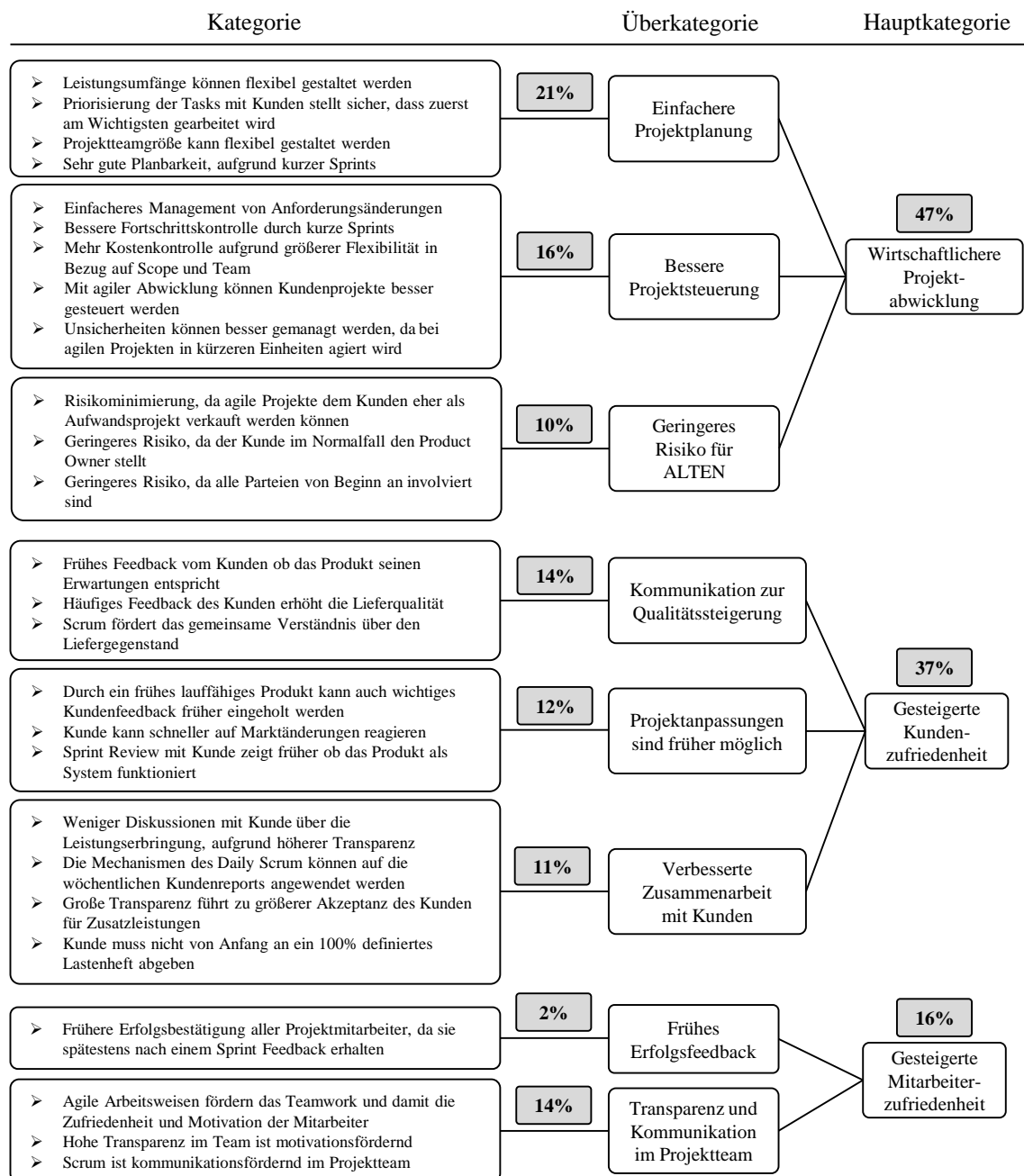


Abbildung 37: Kategoriensystem - Forschungsfrage 2

Quelle: Eigene Darstellung

Den größten Vorteil sieht ALTEN in einer *wirtschaftlicheren Projektabwicklung* mit einer relativen Häufigkeit von 47 %. Für diese Hauptkategorie konnten folgende drei Überkategorien gefunden werden: *Einfachere Projektplanung* (21 %), *Bessere Projektsteuerung* (16 %) und *Geringeres Risiko für ALTEN* (10 %). Vor allem in der flexiblen Gestaltung von Leistungsumfängen und Teamgrößen sieht ALTEN einen entscheidenden Vorteil gegenüber klassischer Projektabwicklung. Eine *bessere Projektsteuerung* zeichnet sich dadurch aus, dass Anforderungsänderungen einfacher gemanagt werden können und mehr Kostenkontrolle aufgrund höherer Flexibilität vorhanden ist. Da der Kunde im Normalfall den Product Owner stellt, kommt es zu einer deutlichen *Risikominimierung* bezüglich falscher Lieferung. Ein weiterer risikominimierender Faktor ist, dass bei agiler Projektabwicklung schon von Beginn an alle Stakeholder im Projekt involviert sind und es dadurch zu weniger negativer Überraschungen am Ende des Projektes kommt.

Neben diesem ökonomischen Aspekt nennt ALTEN mit einer relativen Häufigkeit von 37 % eine *gesteigerte Kundenzufriedenheit* als zweiten Erfolgsfaktor für agile Projektabwicklung. Hier wird vor allem *Kommunikation* (14 %) mit dem Kunden als entscheidend angesehen. Frühes, sowie häufiges Feedback vom Kunden wirken sich sowohl positiv auf die Lieferqualität, als auch auf die Erwartungen des Kunden aus. Ein weiterer Vorteil für den Kunden ist die Möglichkeit, *früher Projektanpassungen vornehmen zu können* (12 %). Größere Transparenz sowie häufigerer Austausch verbessern zudem die *Zusammenarbeit mit dem Kunden* (11 %).

Die letzte Hauptkategorie bildet eine *gesteigerte Mitarbeiterzufriedenheit* mit einer Nennung von 16 %. Verantwortlich dafür ist eine höhere Transparenz im Team und mehr interne Kommunikation. Dies erhöht zum einen die allgemeine Zufriedenheit und wirkt sich zum anderen positiv auf die Motivation der Teammitglieder aus.

In Abbildung 38 werden Aussagen von Experten in Bezug auf die zwei wichtigsten Hauptkategorien, wirtschaftlichere Projektabwicklung und gesteigerte Kundenzufriedenheit, dargestellt:

Hauptkategorie	Zitat
Wirtschaftlichere Projektentwicklung	<i>„Und dann, wenn man agil arbeitet, kann man denk ich auch besser darauf reagieren, weil man in kleineren Scheiben plant und dort schneller feststellt, wenn man aus dem Plan rausfällt.“</i>
	<i>„Wenn man aber alle involvierten Parteien von Anfang an dabei hat und die regelmäßig diesen Austausch haben, dann hat man am Ende glaube ich weniger Überraschungen.“</i>
	<i>„Bei der agilen SW Entwicklung ist der Product Owner meistens der Kunde. Da ist natürlich viel Verantwortung auf seiner Seite. Hier einen Fehler auf Alten Seite zu machen, ist sehr viel kleiner.“</i>
Gesteigerte Kundenzufriedenheit	<i>„Und mit jedem Sprint wird das Produkt noch einmal reifer und besser. Das heißt, dieses Thema Qualität und das, was der Kunde möchte, ist deutlich besser dadurch geworden.“</i>

Abbildung 38: Zitate der Experteninterviews zu Forschungsfrage 2

Ein Vergleich zur Literatur (siehe Kapitel 3.4 Vorteile eines agilen Ansatzes) zeigt sehr ähnliche Ergebnisse. Auch hier werden wirtschaftliche Faktoren (kürzere „Time to Market“, höhere Qualität und größere Effektivität) und persönliche Faktoren (zufriedenere Mitarbeiter) zu den größten Vorteilen gezählt.

8.3 Grundvoraussetzungen für agile Projektentwicklung

Dieses Kapitel beschreibt welche Grundvoraussetzungen für eine erfolgreiche agile Projektentwicklung gegeben sein müssen (siehe Forschungsfrage drei in Kapitel 6). Dazu wurden 17 Kategorien, 7 Überkategorien und 3 Hauptkategorien entwickelt, welche in Abbildung 39 dargestellt werden.

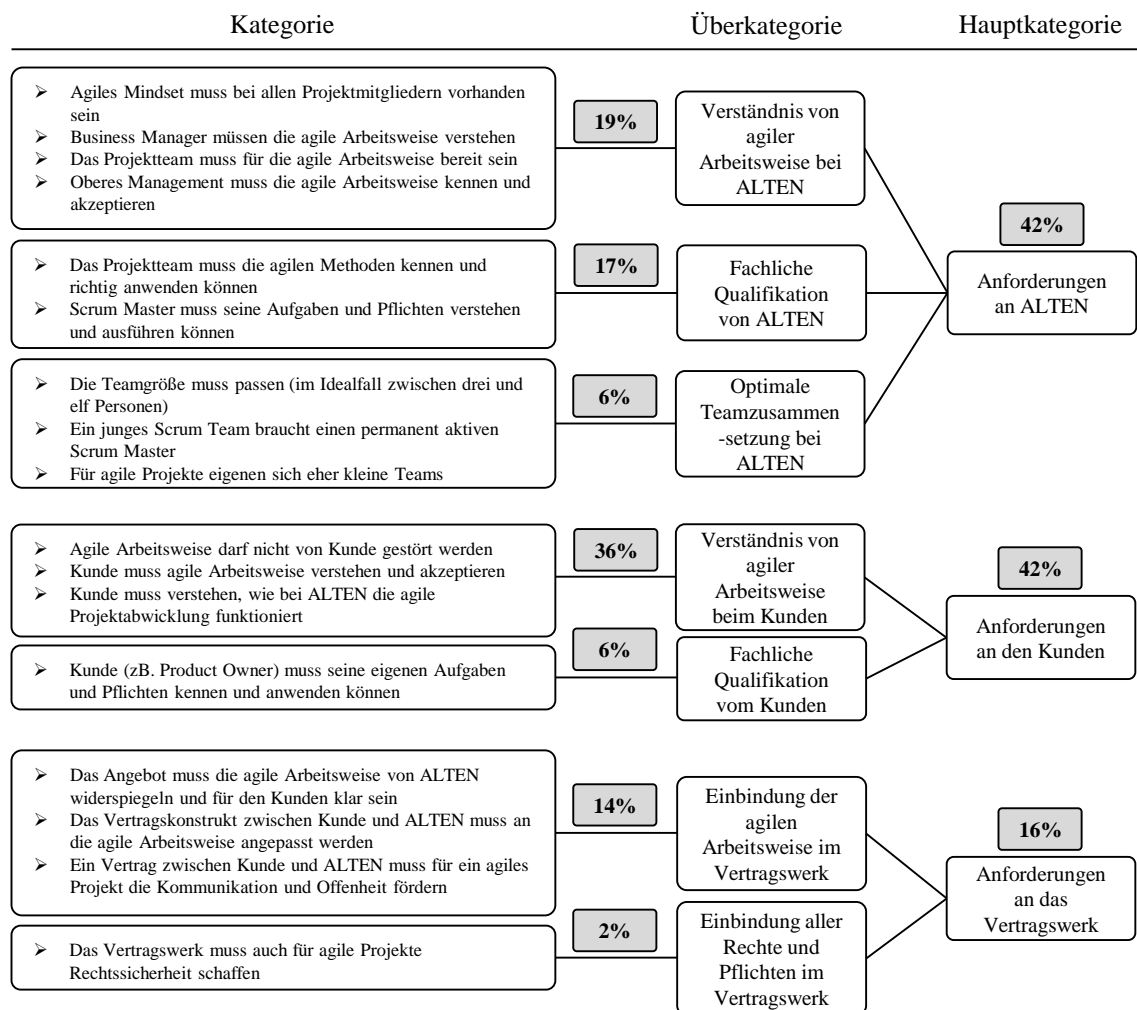


Abbildung 39: Kategoriensystem - Forschungsfrage 3

Quelle: Eigene Darstellung

Mittels Expertenbefragung und anschließender Inhaltsanalyse konnten die gefundenen Grundvoraussetzungen drei Hauptkategorien zugeteilt werden. Mit jeweils 42 % relativer Häufigkeit stellen *Anforderungen an ALTEN* und *Anforderungen an den Kunden* die wichtigsten Hauptkategorien dar.

Die Überkategorien *Verständnis von agiler Arbeitsweise* und *fachliche Qualifikation* können sowohl bei den Hauptkategorien für *ALTEN*, als auch des *Kunden* angefounden werden. Fasst man die Überkategorie *Verständnis von agiler Arbeitsweise bei ALTEN* (19 %) und *Verständnis von agiler Arbeitsweise beim Kunden* (36 %) beider Hauptkategorien zusammen, ergibt sich als relative Häufigkeit ein Wert von 55 %. Mehr als jede zweite Kategorie beschreibt also das *Verständnis von agiler Arbeitsweise* als die wichtigste Grundvoraussetzung für agile Projektentwicklung. Die Analyse zeigt ebenfalls, dass nicht nur das Projektteam, sondern auch Business Manager und das obere Management von *ALTEN* diese Arbeitsweise kennen und akzeptieren müssen. Dieser Grundsatz gilt ebenfalls für die Kundenseite. Auch diese muss die agile Arbeitsweise verstehen und akzeptieren. Zudem bedarf es einer gewissen fachlichen Qualifikation auf beiden Seiten: Das Projektteam (*ALTEN*), der Scrum Master (*ALTEN*) und natürlich der Product Owner (*Kunde*) müssen ihre Aufgaben und Pflichten kennen und auch richtig anwenden können. *ALTEN*-intern lässt sich dies sehr einfach in Form von Schulungen bewerkstelligen. Auf die *Anforderungen an den Kunden* hingegen kann nur begrenzt Einfluss genommen werden. Die einzige Möglichkeit dazu bietet der Vertrag bzw. das Angebot.

Folgende Abbildung zeigt einen Ausschnitt von Expertenaussagen zu den ersten beiden Hauptkategorien, um deren Wichtigkeit zu verdeutlichen:

Hauptkategorie	Zitat
Anforderungen an ALTEN	<i>„Sie müssen selbst wissen, was erwartet wird und proaktiv im Entwicklungsprozess teilhaben. Das ist das Wichtigste. Die Mitarbeiter sind der Kern in der agilen Entwicklung.“</i>
	<i>„Da muss jeder genau wissen, was er eigentlich darf, was er nicht darf, was er kann und was er tun oder lassen sollte.“</i>
Anforderungen an den Kunden	<i>„Problem ist, Agilität funktioniert aus meiner Sicht nur, wenn man auch das Mindset innerhalb der Firma ALTEN und auch beim Kunden hat.“</i>
	<i>„Der Kunde muss von Anfang an verstehen, dass er agil arbeitet. Das ist sehr wichtig. Sollte das für den Kunden nicht verständlich sein, was die agile Entwicklung bedeutet, dann ist er vielleicht nicht reif dafür.“</i>

Abbildung 40: Zitate der Experteninterviews zu Forschungsfrage 3

Die dritte und somit letzte Hauptkategorie wird durch *Anforderungen an das Vertragswerk* mit einer Nennung von 16 % gebildet. Dabei ist es wichtig, dass das Angebot die agile Arbeitsweise von ALTEN widerspiegelt und für den Kunden Klarheit schafft. Auch muss ein Vertrag die für ein agiles Projekt so wichtigen Punkte wie Kommunikation und Offenheit fördern und darf diese in keiner Weise einschränken.

8.4 Anpassungen und Änderungen

Dieses Kapitel behandelt mögliche Anpassungen bzw. Änderungen in Bezug auf die unterschiedlichen IQP-Modelle, die für ein agiles Projekt getroffen werden müssen, um dieses erfolgreich abwickeln zu können (siehe Forschungsfrage vier in Kapitel 6). In Summe konnten 17 Kategorien gefunden werden, die in 5 Überkategorien und 2 Hauptkategorien zusammengefasst wurden (siehe Abbildung 41).

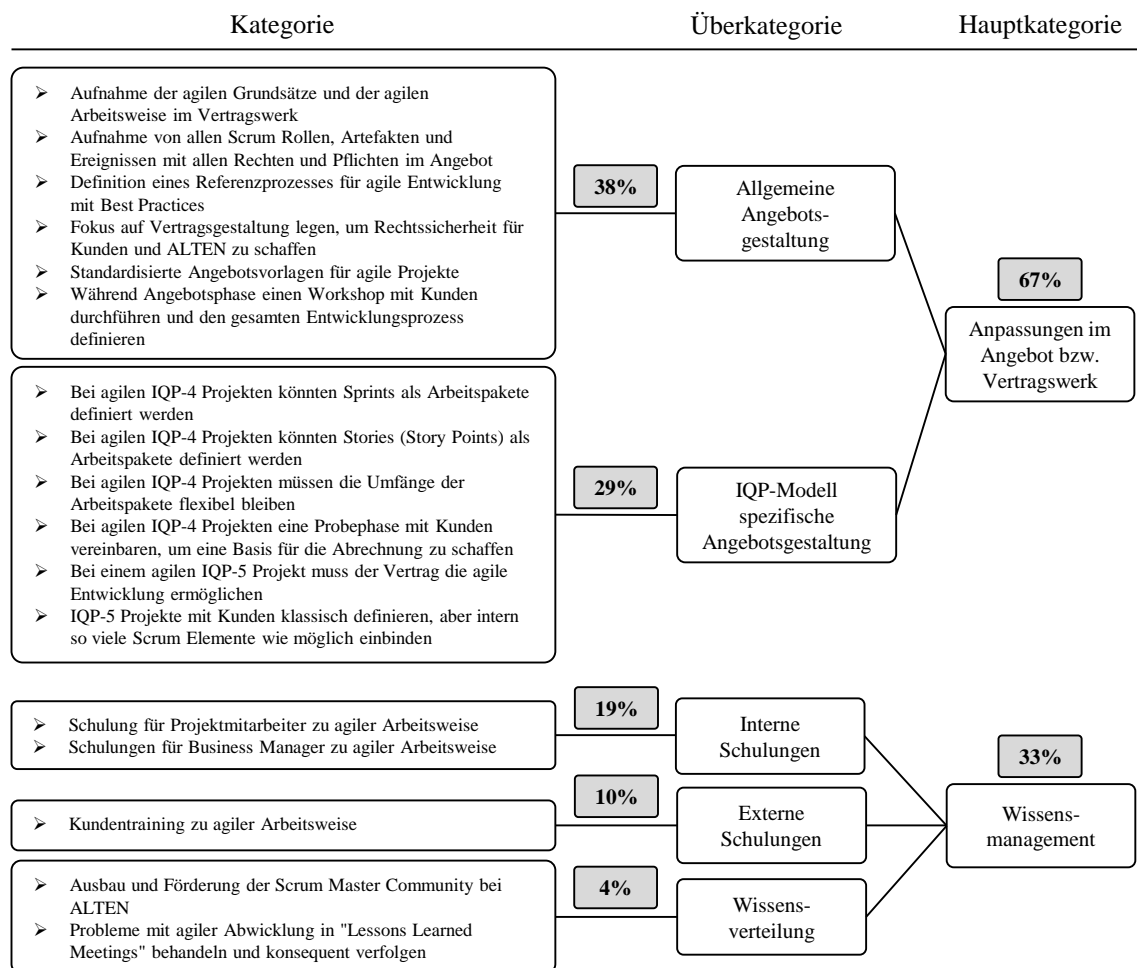


Abbildung 41: Kategoriensystem - Forschungsfrage 4
Quelle: Eigene Darstellung

Die von den Experteninterviews kategorisierten Ideen und Vorschläge zu Verbesserungen bzw. Anpassungen und Änderungen für zukünftige agile Projektabwicklung ergaben dabei zwei Hauptkategorien. ALTEN sieht einerseits die Notwendigkeit, *Anpassungen im Angebot bzw. Vertragswerk* (67 %) vorzunehmen und andererseits ein *Wissensmanagement* (33 %) zu betreiben bzw. weiterzuentwickeln.

Einige der durch Kategorien identifizierten Maßnahmen wurden bei ALTEN schon implementiert, werden zur Vollständigkeit aber trotzdem im Folgenden angeführt.

Mit *internen Schulungen (19 %)*, *externen Schulungen (10 %)* und *Wissensverteilung (4 %)* wurden drei Überkategorien zur Hauptkategorie *Wissensverteilung* entwickelt. Diese Überkategorien sind unabhängig von den IQP-Modellen und daher allgemeingültig. Neben internen Schulungen zu agiler Arbeitsweise für Projektmitarbeiter empfiehlt sich laut Expertenaussagen auch die Schulung von Business Managern, da diese die ersten Ansprechpartner der Kunden sind. Wie schon in Kapitel 8.3 angeführt, ist das Verständnis und die Akzeptanz des Kunden für agile Arbeitsweise essentiell und sollte nicht vernachlässigt werden. Plant man demnach eine Zusammenarbeit mit einem Kunden mit wenig oder ohne Erfahrung im Bereich agiler Projektabwicklung, empfiehlt es sich hier Kundentrainings durchzuführen. Folgende Abbildung zeigt ausgewählte Zitate zu dieser Hauptkategorie:

Hauptkategorie	Zitat
Wissensmanagement	<i>„Ansonsten ist es eher so, dass die Leute wissen wie das funktioniert, dass sie die richtige Ausbildung haben, auch, dass wir das dem Kunden erklären, dass er das versteht.“</i>
	<i>„Auch die Business Manager müssen diese agile Arbeitsweise verstehen, weil das Kundenfeedback auf eine andere Art und Weise kommt, direkter kommt und die Interaktion der Business Manager auch kaufmännisch eine Andere wird.“</i>

Abbildung 42: Zitate der Experteninterviews zu Forschungsfrage 4

Die zweite Hauptkategorie *„Anpassungen im Angebot bzw. Vertragswerk“* setzt sich aus zwei Überkategorien zusammen: *Allgemeine Angebotsgestaltung (38 %)* und *IQP-Modell spezifische Angebotsgestaltung (29 %)*. Es wird daher im Folgenden eine Unterteilung der Ergebnisse vorgenommen.

Allgemein

Die Experten führten an, dass vor allem ein starker Fokus auf die Vertragsgestaltung (wurde in den Interviews gleichgesetzt mit Angebotsgestaltung) zu legen ist, um Rechtssicherheit für den Kunden und ALTEN zu schaffen. Ein erster Schritt sollten standardisierte Angebotsvorlagen, speziell für agile Projekte sein. Hier ist es wichtig, dass neben den agilen Grundsätzen und der agilen Arbeitsweise auch alle Scrum Rollen,

Artefakte, Ereignisse sowie alle Rechte und Pflichten von ALTEN und Kunden in Bezug auf die agile Projektabwicklung verständlich dargelegt werden. Hier könnte die Definition eines Referenzprozesses mit Best Practices für mehr Klarheit auf Kunden- als auch auf ALTEN-Seite sorgen. Um eine noch bessere Zusammenarbeit mit dem Kunden zu erreichen und diesem die agile Projektabwicklung, wie sie bei ALTEN stattfindet, näher zu bringen, empfiehlt sich ein gemeinsamer Workshop schon während der Angebotsphase. Dabei sieht der Kunde im Detail wie ein Sprint von der Planung bis zum Review bzw. der Retrospektive durchgeführt wird und welche Mitwirkungspflichten er trägt. Unklarheiten können dabei schon vor dem eigentlichen Projektstart angesprochen und im besten Fall vollständig eliminiert werden.

IQP-Modell spezifisch

Bei den IQP-Modell spezifischen Ideen bzw. Vorschlägen für mögliche Anpassungen und Änderungen konnten nur Kategorien zu den IQP-Modellen vier und fünf gefunden werden. IQP-3 wurde einheitlich von allen Experten als sehr gut geeignetes Modell für agile Projektabwicklung beschrieben und benötigt daher keine spezifischen Anpassungen. Der Eignung der IQP-4 und IQP-5 Modelle für agile Projektabwicklung standen die Befragten geschlossen kritisch gegenüber und nannten auch hier weniger Kategorien als erwartet.

Um eine agile Projektabwicklung mit einem IQP-4 Modell durchführen zu können, führten die Experten an, dass die Umfänge der Arbeitspakete unbedingt flexibel bleiben müssen, da sonst die wichtigen Scrum-Elemente, wie das Review und die Retrospektive an Relevanz verlieren. Auch wurde erwähnt, dass man für ein agiles IQP-4 Modell Arbeitspakete mittels Sprints oder Story Points definieren könnte, diese Formen aber ein Konfliktpotential mit dem Kunden bergen.

Eine agile Projektabwicklung mit einem IQP-5 Modell (Festpreis) sahen alle Experten als problematisch an, da bei diesem Modell der Aspekt der Retrospektive fehlt und ein fixierter Leistungsumfang im Kontrast zur agilen Entwicklung nach Scrum steht. Ein hybrides Projektmanagement, die Projektabwicklung zum Kunden hin klassisch gestaltet, aber intern so viele Scrum Elemente wie möglich integriert, würde sich hier besser eignen.

Nachfolgende Abbildung enthält Expertenaussagen zu möglichen Anpassungen im Angebot bzw. Vertragswerk und soll die oben angeführten Interpretationen der Ergebnisse untermauern:

Überkategorie	Zitat
Allgemeine Angebotsgestaltung	„Das große Hindernis, das ich sehe, ist das Vertragswerk. Scrum erfordert viel Kommunikation und fordert Offenheit. Und wenn ich jetzt irgendwo einen Vertrag dazwischengeschaltet habe, der die Offenheit nicht intensiv fördert, sondern blockiert, dann wird es scheitern.“
	„Ich glaube, wir sollten auf einer Seite gucken, dass wir die Angebotsvorlagen für agile Projekte standardisieren, ich glaube da gibt es auch schon Ansätze, wo sich Leute Gedanken machen. Das man da halt einfach ein einheitliches Vorgehen hat und eine Vergleichbarkeit der Projekte.“
IQP-Modell spezifische Angebotsgestaltung	„Also ich würde bei IQP-5 und dem Zusammenspiel mit dem Kunden sagen, das ist nicht so geeignet dafür. Natürlich können wir intern immer agil und der Kunde kann für seine Sachen agil arbeiten, aber so das Zusammenspiel macht wenig Sinn.“
	„Sagen wir mal so, wenn ich es erlaube eine Retrospektive zu machen und gegebenenfalls diese Preise zu verändern, wenn ich feststelle, dass es doch komplexer war und ich das auch vernünftig begründen kann. Wenn ich diese Tür aufmache, dann würde auch IQP-4 funktionieren.“

Abbildung 43: Zitate der Experteninterviews zu Forschungsfrage 4

8.5 Agile Projektabwicklung mit Scrum für ein Festpreisprojekt

Dieses Kapitel adressiert die fünfte und somit letzte Forschungsfrage der Masterarbeit (siehe Kapitel 6). Im Gegensatz zu den Forschungsfragen eins bis vier wird zur Beantwortung dieser nicht auf ALTEN-interne Expertenmeinungen zurückgegriffen. Die im Folgenden vorgestellten Ergebnisse wurden zum Großteil mittels Literaturrecherche erarbeitet und sollen darstellen, wie eine agile Projektabwicklung mit Scrum für ein Festpreisprojekt (IQP-5 Modell) aussehen könnte und welche Vertragsinhalte berücksichtigt werden müssen.

In Kapitel 5 (Agile Vertragsgestaltung) wurden neben dem klassischen Festpreis vier agile Abwandlungen vorgestellt:

- 5.3.1 Klassischer Festpreis mit agilen Techniken
- 5.3.2 Festpreis mit garantiertem Minimalumfang
- 5.3.3 Money for Nothing, Change for Free
- 5.3.4 Der agile Festpreis

Welche dieser vier Varianten nun am ehesten für eine agile Projektabwicklung verwendet werden sollte, hängt hauptsächlich von den Präferenzen des Kunden ab. Fordert dieser einen klassischen Festpreis, so empfiehlt es sich, das Projekt wie in Kapitel 5.3.1 (Klassischer Festpreis mit agilen Techniken) angeführt, abzuwickeln. Dabei wird zum Kunden hin das Projekt klassisch definiert, ALTEN-intern aber, werden so viele agile Techniken wie möglich eingebunden (siehe Seite 53-54).

Wünscht der Kunde jedoch eine agile Projektabwicklung mit festem Preis, so empfiehlt sich eine Projektabwicklung in Anlehnung zu dem in Kapitel 5.3.4 (Der agile Festpreis) beschriebenen agilen Festpreis (siehe Seite 57-60).

Die Varianten „5.3.2 Festpreis mit garantiertem Minimalumfang“ und „5.3.3 Money for Nothing, Change for Free“ stellen Grundkonzepte zur agilen Abwicklung dar, weisen aber keinen eigenständigen Gesamtprozess auf. Diese könnten zum Beispiel als Erweiterungen in den agilen Festpreis oder den klassischen Festpreis mit agilen Techniken eingebunden werden.

Welche Aspekte für eine agile Vertragsgestaltung berücksichtigt werden sollten, beschreibt Kapitel 5.1 (Wichtige Aspekte für agile Vertragsgestaltung, Seite 48-51). Dabei können vier Punkte hervorgehoben werden, die den Minimalumfang an Vertragsinhalten darstellen:

- Vertrag muss unvollständige Funktionsbeschreibung erlauben
- Gemeinsames Verständnis
- Kooperative Problemlösung
- Scrum Elemente

Ein agiler Vertrag muss auf jeden Fall so ausgeführt sein, dass von vorher definierten Funktionen abgewichen werden kann und Anforderungen, die zu Beginn noch nicht bekannt waren, ohne Mehraufwand und Mehrkosten in die Entwicklung integriert werden können (siehe 5.1.1 Unvollständige Funktionsbeschreibung).

Da der Kunde bei agiler Projektabwicklung eng in die Entwicklung eingebunden werden muss, bedarf es eines gemeinsamen Verständnisses, wie bei ALTEN die agile Entwicklung funktioniert und auf welchen Grundsätzen diese basiert. Es sollten also zumindest die wichtigsten agilen Werte und Prinzipien sowie ein Referenzprozess zum besseren Verständnis des Kunden angeführt werden (siehe 5.1.2 Gemeinsames Verständnis).

Wie bei allen Projekten kann es auch bei agiler Abwicklung zu Problemen und Missverständnissen kommen. Ein Problemlösungsprozess, Eskalationsmechanismen und ein vorab definiertes Riskshare-Modell sollten hier nicht fehlen (siehe 5.1.3 Kooperative Problemlösung).

Einer der wichtigsten Punkte ist die Definition der Scrum Elemente, also der Rollen, Artefakte und Ereignisse. Hierbei sollten die Rechte und Pflichten von Scrum Master, Projektteam und Product Owner festgehalten werden. Neben dem Sprint Backlog stellt das Product Backlog das zentrale Element eines Scrum Prozesses dar. Wie diese zu führen sind und wer vorrangig für die Pflege verantwortlich ist, sollte gemeinsam festgelegt werden. Für ein gemeinsames Verständnis zum Scrum Vorgehensmodell und der

chronologischen Abfolge aller Ereignisse, empfiehlt sich eine visuelle Darstellung des Prozesses im Vertrag (siehe 5.1.4 - 5.1.6)

Grundsätzlich kann jeder Vertrag mehr oder weniger umfangreich ausgestaltet werden. Die oben angeführten vertraglichen Aspekte bilden dabei die Mindestanforderungen an einen agilen Vertrag und sollten auch bei einem guten Vertrauensverhältnis zum Kunden auf keinen Fall weggelassen werden.

9 Fazit

Im Rahmen der vorliegenden Arbeit sollte gezeigt werden, welche Anpassungen bzw. Änderungen für ein agiles Projekt in Bezug auf die unterschiedlichen ALTEN-internen Zusammenarbeitsmodelle (IQP-Modelle) und der Vertragsgestaltung getroffen werden müssen, um dieses erfolgreich abwickeln zu können. Auch wurde untersucht, mit welchen Hauptproblemen ALTEN bei klassischer Projektabwicklung derzeit konfrontiert wird und worin die Vorteile von agiler Projektabwicklung liegen. Die gewonnenen Ergebnisse sollen für ALTEN eine Bestandsaufnahme der aktuellen internen Situation aufzeigen und einen Grundstock an sinnvollen Handlungsempfehlungen bereitstellen, wie agile Ansätze für zukünftige agile Projekte implementiert werden könnten.

Um einen Überblick über das umfangreiche Themengebiet der agilen Projektabwicklung zu erhalten, wurde als Erstes eine ausführliche Literaturrecherche durchgeführt. Neben den Bausteinen des agilen Projektmanagements und den Grundlagen zur Scrum Methodik, wurden in einem Theorieteil der Masterarbeit die klassischen den agilen Ansätzen gegenübergestellt und dabei Gemeinsamkeiten sowie Unterschiede beleuchtet. Zudem war es notwendig, sich in das Thema der ALTEN-internen Zusammenarbeitsmodelle einzuarbeiten. Da für einen Ingenieurdienstleister wie ALTEN vor allem die Zusammenarbeit mit dem Kunden einen entscheidenden Erfolgsfaktor darstellt, wurde auch die Vertragsgestaltung für agile Projekte im Theorieteil näher betrachtet.

Da es sich bei den zu Beginn definierten Forschungsfragen um einen Mix aus firmeninternen und allgemeinen Fragestellungen im Bereich des klassischen und agilen Projektmanagements handelt, wurde die Entscheidung getroffen, diese sowohl mittels der schon angeführten Literaturrecherche, als auch firmeninternen Experteninterviews zu beantworten. Auf Basis der Literaturrecherche wurde für die sechs durchgeführten Interviews ein geeigneter Interviewleitfaden entwickelt, um die gewünschten Auskünfte zu den Forschungsfragen zu erhalten. Das dabei erhaltene Datenmaterial wurde vollständig transkribiert und einer qualitativen Inhaltsanalyse nach Mayring unterzogen. Für die ersten vier der fünf Forschungsfragen konnte anschließend aus den Resultaten der Inhaltsanalyse in Anlehnung an die Gioia-Methode ein Kategoriensystem entwickelt werden, welches als Interpretationsgrundlage für die Ergebnisse diente. Die

Beantwortung von Forschungsfrage fünf hingegen, erfolgte zum Großteil mit den aus der Literatur gewonnenen Erkenntnissen.

Die größten Probleme bei klassischer Projektabwicklung sieht ALTEN derzeit mit den vom Kunden initial definierten Anforderungen. Neben einer unzureichenden Anforderungsbeschreibung und häufigen Anforderungsänderungen des Kunden erwähnen die Experten auch einen schlechten Informationsaustausch mit dem Kunden als negativen Einflussfaktor. Langwierige Diskussionen über den Liefergegenstand, sinkende Kundenzufriedenheit und zeitliche Projektüberschreitungen können die Folge sein. Bei der agilen Projektabwicklung hingegen können Leistungsumfänge flexibel gestaltet werden, der Kunde wird über die gesamte Projektlaufzeit stark in die Entwicklung eingebunden und gibt regelmäßiges Feedback und Mitarbeiter haben die Chance, freier und selbstbestimmter zu arbeiten. Deshalb nennt ALTEN hier Vorteile zu einer wirtschaftlicheren Projektabwicklung sowie einer gesteigerten Kunden- und Mitarbeiterzufriedenheit. Nun kann aber nicht jedes Projekt agil abgewickelt werden. Es gibt gewisse Grundvoraussetzungen die eingehalten werden sollten, um agile Projekte erfolgreich abschließen zu können. Dazu zählt vor allem das agile Mindset: Das Projektteam und der Kunde müssen die agile Arbeitsweise verstehen und zu 100 % akzeptieren. Auch sollte das obere Management und die Business Manager verstehen, was unter agiler Entwicklung gemeint ist und wie sie funktioniert. Außerdem müssen alle in das Projekt involvierten Parteien, also das Development Team (ALTEN), der Scrum Master (ALTEN) und der Product Owner (Kunde) die agilen Methoden & Techniken anwenden können und noch wichtiger, deren eigenen Rechte und Pflichten kennen. Um die interne fachliche Qualifikation des Projektteams und des Scrum Masters zu gewährleisten sollten regelmäßig Schulungen und Zertifizierungen angestrebt werden. Mögliche fachliche Defizite auf Kundenseite könnte man mit gemeinsamen Workshops oder Kundentrainings reduzieren. Um für ALTEN aber auch für den Kunden Rechtssicherheit zu schaffen, sollten außerdem Anpassungen in der Vertragsgestaltung vorgenommen werden. Neben der Erstellung von standardisierten Angebotsvorlagen für agile Projekte, sollten in diesen die agilen Grundsätze, die agile Arbeitsweise und alle Scrum Rollen, Artefakte, Ereignisse, sowie in diesem Zusammenhang stehende Rechte und Pflichten von ALTEN und Kunden verständlich niedergeschrieben werden. Für das IQP-3 Modell konnten keine spezifischen Anpassungen gefunden werden, da sich dieses Modell schon sehr gut für agile Abwicklung eignet. Die Modelle IQP-4 und IQP-5

können jedoch als kritisch für die agile Abwicklung angesehen werden, da bei diesen der Leistungsumfang fixiert wird, für Scrum dieser aber variabel bleiben muss. Über die Experteninterviews konnten nur wenige konkrete Anpassungs- bzw. Änderungsvorschläge hierzu gefunden werden. Das IQP-4 Modell könnte dann funktionieren, wenn es gelingt, die Arbeitspakete und Umfänge flexibel zu gestalten. Das IQP-5 Modell wurde interviewübergreifend als das am wenigsten geeignete Modell für agile Projektentwicklung eingestuft, da der Aspekt der Scrum Retrospektive verloren geht und der fixierte Leistungsumfang im kompletten Gegenteil zur Scrum Idee, Umfänge zu priorisieren und nur jene mit höchstem Geschäftswert zu entwickeln, steht. Dennoch konnte in der Literatur ein Ansatz gefunden werden, um auch solche Projekte agil abwickeln zu können. Dieser unter dem Begriff „der agile Festpreis“ angeführte Ansatz wurde deshalb in der Masterarbeit in seinen Grundzügen behandelt.

Abschließend kann angemerkt werden, dass sich ALTEN auf einem guten Weg für zukünftige, erfolgreiche agile Projektentwicklung befindet. Die Interviews haben klar gezeigt, dass schon jetzt ALTEN-intern ein großes Know-how vorhanden ist und kontinuierlich an Verbesserungen (Angebotsanpassungen, Schulungen für Scrum-Master und Projektmitarbeiter) gearbeitet wird. Auch zeigte sich, dass aktuelle Defizite bei agilen Projekten realistisch reflektiert wurden und vorgeschlagene Anpassungen bzw. Änderungen der Experten im Einklang mit jenen in der Literatur angeführten, stehen. Zudem kann positiv erwähnt werden, dass bei ALTEN das obere Management in diesem Prozess aktiv beteiligt ist.

Literaturverzeichnis

- Baur, N., & Blasius, J. (2014). *Handbuch Methoden der empirischen Sozialforschung*. Wiesbaden: Springer-Verlag.
- Beck, K., Schwaber, K., Sutherland, J., Beedle, M., van Bennekum, A., Cockburn, A., . . . Thomas, D. (2001). *Agile Manifesto*. Abgerufen am 01. Jänner 2019 von <https://agilemanifesto.org/>
- Berwanger, J. (19. Februar 2018). *Gabler Wirtschaftslexikon*. Abgerufen am 13. Dezember 2018 von <https://wirtschaftslexikon.gabler.de/definition/werkvertrag-47440/version-270704>
- Bewernick, M., Schreyögg, G., & Costas, J. (September 2013). Charismatische Führung: Die Konstruktion von Charisma durch die deutsche Wirtschaftspresse am Beispiel von Ferdinand Piëch. *Schmalenbachs Zeitschrift für betriebswirtschaftliche Forschung*, S. 434-465.
- Cockburn, A. (2004). *Crystal Clear: A Human-Powered Methodology for Small Teams*. Boston: Addison-Wesley Professional.
- Deppermann, A. (2013). *Gespräche analysieren: Eine Einführung*. Wiesbaden: Springer-Verlag.
- Engel, C., & Quadejacob, N. (21. Jänner 2009). *Competence Site*. Abgerufen am 27. Jänner 2019 von <https://www.competence-site.de/studie-von-gpm-und-pa-consulting-fuenf-erfolgskriterien-fuer-projekte/>
- Fleig, J. (29. August 2016). *Business Wissen*. Abgerufen am 15. Jänner 2019 von <https://www.business-wissen.de/hb/mit-80-20-regel-und-abc-analyse-prioritaeten-setzen/>
- Gioia, D., & Corley, K. (1. Juni 2004). Identity Ambiguity and Change in the Wake of a Corporate Spin-off. *Administrative Science Quarterly*, S. 173-208.
- Heimrath, M. (2018). *Agiles Projektmanagement - Scrum für Einsteiger*. Hannover: Independently published.
- Helfferich, C. (2005). *Die Qualität qualitativer Daten: Manual für die Durchführung qualitativer Interviews*. Wiesbaden: VS Verlag für Sozialwissenschaften.
- Mayring, P. (2002). *Einführung in die qualitative Sozialforschung: Eine Anleitung zu qualitativem Denken*. Weinheim: Beltz.

- Mayring, P. (2010). *Qualitative Inhaltsanalyse: Grundlagen und Techniken*. Weinheim: Beltz.
- Michels, B. (2015). *Projektmanagement Handbuch - Grundlagen mit Methoden und Techniken für Einsteiger*. Leipzig: 2015, Benjamin Michels Self Publishing.
- Opelt, A., Gloger, B., Pfarl, W., & Mittermayer, R. (2018). *Der agile Festpreis*. München: Carl Hanser Verlag München.
- Pieper, F.-U., & Roock, S. (2017). *Agile Verträge*. Heidelberg: dpunkt.verlag GmbH.
- Pink, D. (2011). *Drive: The Surprising Truth About What Motivates Us*. Riverhead Books.
- Plewa, W. (2017). *Business Wissen*. Abgerufen am 05. Jänner 2019 von <https://www.business-wissen.de/artikel/scrum-so-erstellen-sie-gute-user-stories/>
- Powers, S. (2017). *Adventures with Agile*. Abgerufen am 01. Jänner 2019 von <https://www.adventureswithagile.com/2017/03/25/what-is-the-agile-mindset/>
- Preußig, J. (2015). *Agiles Projektmanagement*. Freiburg: Haufe-Lexware GmbH und Co. KG.
- Proagile. (2017). *Proagile*. Abgerufen am 25. Jänner 2019 von <https://proagile.de/unterschied-klassisch-agil/>
- Rubin, K. (2014). *Essential Scrum*. Rheinbreitbach: mitp Verlags GmbH & Co. KG.
- Sawyer, K. (2007). *Group Genius: The Creative Power of Collaboration*. Basic Books.
- Stacey, R. (1996). *Strategic Management and Organisational Dynamics: The challenge of complexity to ways of thinking about organisations*. Prentice Hall.
- Standish Group. (2015). *Standish Group*. Abgerufen am 27. Jänner 2019 von https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf
- Stärk, J. (2016). *Agiles Projektmanagement im Anlagen- und Maschinenbau*. Norderstedt: BoD - Books on Demand.
- Sutherland, J. (2008). *Jeff Sutherland*. Abgerufen am 03. Februar 2019 von <http://jeffsutherland.com/scrum/Agile2008MoneyforNothing.pdf>
- Takeuchi, H., & Nonaka, I. (Jänner 1989). The New New Product Development Game. *Harvard Business Review*.
- Timinger, H. (2017). *Modernes Projektmanagement*. Weinheim: WILEY-VCH Verlag GmbH & Co. KGaA.
- Wiedeker, M. (2018). *Projektmanagement für Anfänger*. Leipzig: Amazon Distribution GmbH.

Anhang

A. Gesamter Leitfaden der Experteninterviews

Wo liegen die Hauptprobleme bei klassischer Projektabwicklung?

1. Was sind Ihrer Meinung nach negative Einflussfaktoren für den Projekterfolg bei klassisch gemanagten Projekten?
2. Welche dieser Faktoren denken Sie, sind die Hauptgründe für nicht erfolgreich abgeschlossene Projekte?
3. In vielen Unternehmen gibt es einen Trend in Richtung agiler Abwicklung. Sehen Sie die treibende Kraft hier eher beim Kunden oder in der Alten Organisation?
4. Wo denken Sie, sieht der Kunden die Vorteile für sich bei agiler Abwicklung?
5. Wo sieht Alten die Vorteile bei agiler Abwicklung? Welche oben genannten negativen Einflussfaktoren könnte man mit agilem Vorgehen kompensieren?

Was sind Grundvoraussetzungen für erfolgreiche agile Projekte?

1. Denken Sie, gibt es gewisse Grundvoraussetzungen, die es benötigt, damit eine agile Projektabwicklung Sinn macht?
2. Welche der von Ihnen angeführten Punkte würden Sie zu den Knockout Kriterien für agiles Vorgehen zählen?
3. Wenn Sie an Ihr letztes Scrum Projekt denken, was lief hier besonders gut und wo sehen Sie Vorteile dieser Methodik?
4. Wenn Sie an Ihr letztes Scrum Projekt denken, was lief nicht so gut und wo sehen Sie Nachteile dieser Methodik?
5. Gab es ein Projekt, dass agil abgewickelt wurden, dass Sie im Nachhinein aber eher klassisch machen würden? Wo lagen hier die Gründe?

Wie gut eignen sich die unterschiedlichen IQP-Modelle für agiles Vorgehen?

1. Wie gut eignet sich aus Ihrer Sicht das IQP-3, 4 und 5 Modell für agiles Vorgehen nach Scrum?
2. Warum denken Sie, eignet sich dieses Modell (nicht) für die agile Projektabwicklung?

Welche Anpassungen oder Änderungen müssen getroffen werden um dem agilen Vorgehen zu genügen?

1. Wo sehen Sie Änderungs- bzw. Anpassungsbedarf in Bezug auf das IQP-3,4 und 5 Modell, um ein Projekt mit Scrum erfolgreich abzuwickeln zu können?

Wie sollte ein allgemeiner Scrum basierter Prozess aussehen?

1. Denken Sie an Ihr letztes agiles Projekt. Haben Sie dieses laut dem in der Literatur beschriebenen Scrum-Prozesses durchgeführt, oder gab es hierbei Abweichungen in Bezug auf die Meetings, Rollen und Artefakte?
2. Warum sind Sie vom Prozess abgewichen und ergaben sich dadurch Vorteile oder eher Nachteile aus Ihrer Sicht?
3. Wenn Sie an Ihr letztes nicht sehr erfolgreiches agiles Projekt denken, fällt Ihnen ein was hier nicht so gut gelaufen ist, in Bezug auf die Meetings, Rollen und Artefakte? Würden Sie es beim nächsten Projekt anders machen?

B. Interviewberichte

Interview 1

I: Was sind deiner Meinung nach negative Einflussfaktoren für den Projekterfolg für klassisch gemanagte Projekte?

B1: Negativer Einfluss ist, dass man sehr oft unflexibel ist. Man hat ein relativ starres Planungsgerüst, wo man halt einen Anfang und ein Ende hat und wo man dann, die Dinge oft versucht, parallel zu machen und am Schluss manches Mal, Themen nicht zusammenpassen. Also, da jeder so seinen eigenen Plan verfolgt, aber am Schluss dann feststellt, dass Dinge nicht zusammenpassen oder zeitlich komplett aus dem Rahmen kommen, weil man halt oft sehr lange Phasen hat, in so einer klassischen Projektplanung. Was dann noch mit dazu kommt ist, wenn man das Thema mit den Kunden anschaut, ist es sehr oft so, dass man bei so einer klassischen Projektabwicklung so eine Art Lastenheft hat, wo dann genau definiert sein sollte, was zu tun ist und man aber auf der Strecke relativ oft draufkommt, dass man eigentlich Dinge vergessen hat oder unklar sind und dann kommen ewige Diskussionen, die das alles erschweren und ja auch das Zusammenarbeitsgefüge erschweren dadurch.

I: Jetzt hast du mir Punkte genannt in Bezug auf den Kunden und in Bezug auf die Planung. Fällt dir vielleicht noch ein negativer Einflussfaktor für das Projektteam ein?

B1: Da ist es auch so, dass das Projektteam mit unklaren Bedingungen losstartet. Das es schon ein Ziel gibt, aber der Weg dorthin nicht klar ist, weil das Ziel oft zu weit in der Ferne ist und da braucht man halt oft eine starke Projektleitung und starke Prozesse, die man nicht immer hat. Da haben die Mitarbeiter oft eine Unsicherheit, speziell in der Anfangsphase.

I: Und von diesen genannten Faktoren, wo siehst du die Hauptgründe für nicht erfolgreich abgeschlossene Projekte? Liegt das eher, zwischen Alten und Kunde oder in der Projektplanung oder im Projektteam?

B1: Sehr oft ist es zum einen ein Erfahrungsthema, dass man das Projekt falsch einschätzt. Das ist auch relativ oft so, dass man den Umfang falsch einschätzt. Zu optimistisch zum Beispiel und dann in der Projektlaufzeit draufkommt, es ist eigentlich ganz anders als man es erwartet hat und was auch oft Thema ist, dass sich der Kunde was Anderes

erwartet als das, was man als Lieferant versteht. Das heißt, so dieses genaue definieren, was zu tun ist. Bei der klassischen Projektleitung ist es so, dass man eigentlich sehr autark arbeitet und immer wieder mal ein Zwischenergebnis abgibt und da ist es oft so, dass man etwas Anderes bringt als der Kunde erwartet hat, weil halt dieser Austausch nicht so intensiv ist.

I: Wie du weißt gibt es in vielen Unternehmen jetzt den Trend in die Richtung agile Abwicklung. Wo siehst du jetzt die treibende Kraft dafür? Kommt die eher von der Alten Organisation oder vom Kunden?

B1: Es hat einen Wechsel gegeben. Am Anfang ist es von der Alten Organisation gekommen, weil wir einfach gesehen haben, dass wir über den Weg unsere Projekte besser managen können und zwischenzeitlich ist es so, dass fast alle Kunden das fordern, dass wir agil entwickeln. Aber es heißt, wir haben jetzt schon konkrete Anforderungen vom Kunden, die zitiert eine agile Entwicklung erwarten. Da hat es sich ein bisschen geändert - von, dass wir einen Vorteil sehen - zu, die Kunden sehen nun auch einen Vorteil bzw. Nutzen dahinter.

I: Wo glaubst du, sieht der Kunde die Vorteile für sich bei dieser agilen Abwicklung?

B1: Die haben halt auch das Problem, dass wenn sie einen normalen klassischen Werkvertrag haben, am Anfang ein dickes Lastenheft abgeben müssen und sehr oft ist es so, dass sie es selbst nicht beschreiben können oder dass sie am Anfang inhaltlich noch gar nicht wissen, wie sich das Projekt entwickeln wird. Ganz ein einfaches Thema, man entwickelt einen Prototyp und kommt während der Entwicklung eigentlich erst drauf, was die Schwierigkeiten sind. Und deshalb ist es für sie eine gute Möglichkeit, sich aus dieser 100 % definierten Spezifikation zurückzuziehen und über diesen Weg quasi iterativ sich zum Ziel hin zu entwickeln. Ein zweiter großer Vorteil ist, dass man schon vom frühen Zeitpunkt an ein laufendes Produkt hat. Das heißt, man hat schon relativ früh etwas und man kann probieren, wird das am Schluss rauskommen was man sich vorgestellt hat. Ein weiterer Vorteil ist, da man agil unterwegs ist, kann man auch mit dem Lieferanten flexible Teamgrößen zusammenstellen oder auch die Umfänge flexibel definieren. Man hat über den Weg noch ein bisschen mehr das Thema Kosten unter Kontrolle.

I: Jetzt möchte ich nochmal kurz in eine spezielle Richtung gehen und zwar, wenn man jetzt den Unterschied aufgreift zwischen agil und klassisch, dann ist es doch so, dass beim

Klassischen der Kunde mal anfangs dieses Lastenheft definiert und dann gibt es bestimmt Kunden, die sagen, ok jetzt sehe ich mich nicht mehr in der Verantwortung irgendetwas zu tun, wohingegen bei agil er doch über den gesamten Dienstleistungsrealisierungsprozess immer in diesem Prozess involviert ist und immer dieses Kundenfeedback gefordert wird. Ist den Kunden das bewusst, dass sie bei agiler Abwicklung ein anderes Mindset benötigen als bei Klassischer?

B1: Das ist unterschiedlich, das hängt vom Kunden ab. Wir haben Kunden, bei denen ist genau die Problematik, dass die quasi agil arbeiten, aber dann nichts damit zu haben wollen und sich dann wundern, warum wir immer mit Themen daherkommen bzw. wir in eine Situation kommen, wo wir Dinge entwickeln, wo der Kunde sagt, dass wollen sie eigentlich gar nicht, weil wir halt aus der Diskussion heraus verstanden haben, dass ist die nächste Planung für den nächsten Sprint, dann aber erfährt, dass der Kunde unter Anführungszeichen nur mal laut gedacht hat. Das sind so die Kunden die das Thema agil toll finden, aber sich nicht richtig damit beschäftigt haben. Bei anderen Kunden ist es aber wieder so, die haben globale Scrum Teams geplant, wo sie selbst eingebunden sind. Da ist der Kunde der Product Owner und die dann halt über diesen Weg die Interaktion sehr stark fördern und deren Organisation auch danach ausgerichtet haben. Es ist also eine Frage der Reife, wie weit sich die Firmen auch schon Richtung agiler Entwicklung orientiert haben.

I: Du hast mir ja ganz am Anfang die negativen Einflussfaktoren genannt und die Hauptgründe, wenn Projekt nicht erfolgreich abgeschlossen werden, bei klassisch gemanagten Projekten. Jetzt haben wir ja gerade über die agile Vorgehensweise gesprochen. Denkst du, kann man mit der agilen Vorgehensweise alle oben genannten negativen Einflussfaktoren kompensieren, oder wo denkst du, wird es auch mit agil schwierig?

B1: Agil ist so das Thema, was wir hatten mit Unklarheiten oder dass man halt zu weit entfernte Ziele hat. Das kann man mit Sicherheit besser in den Griff kriegen, weil man halt in kürzeren Zeiten und in kürzeren Einheiten unterwegs ist. Was da passieren kann ist, dass man dann zum Beispiel diesen eigentlichen Projektinhalt und Projektende aus den Augen verliert, weil man dann immer nur in kurzen Etappen unterwegs ist. Das ist dieser Nebeneffekt dabei, habe ich auch schon beobachtet, dass dann oft die Teams gar nicht mehr wissen, wo das, was sie entwickeln ins große Ganze reinpasst. Und das Thema Richtung Kunde kann sein, dass sie im Rahmen der agilen Entwicklung noch weniger

genau definieren, was sie eigentlich haben wollen. Also das Problem, was wir oft bei Projekten haben, ist, dass das Lastenheft oft sehr unklar definiert ist. Und wenn man dann ins Agile reingeht und man sich von Einheit zu Einheit durch hantelt, kann es dann sogar so weit kommen, dass der Kunden keine Vorgaben mehr gibt und diese dann von uns kommen und dann am Schluss der Kunde draufkommt, das ist doch nicht das, was ich haben möchte. Also das heißt, da ist dann diese Klarheit, die vom Kunden kommen sollte, noch geringer. Er zieht sich also komplett aus dieser Spezifikationsverantwortung raus.

I: Wir haben somit Teil A abgeschlossen und ich gehe somit direkt in Teil B über. Hier geht es um die Grundvoraussetzungen für agiles Vorgehen. Da starte ich gleich mit der ersten Frage. Denkst du, gibt es überhaupt gewisse Grundvoraussetzungen, die es benötigt, dass eine agile Projektabwicklung Sinn macht?

B1: Zum einen muss die Mannschaft dafür bereit sein, so zu arbeiten. Es ist ganz eine andere Arbeitsweise, als die, die man bisher kennt. Und wenn man sich das bei uns anschaut, da haben wir ja verschiedene Themenbereiche. Und als Beispiel mechanische Konstruktion und als zweites Beispiel nehme ich SW Entwicklung. Bei der SW Entwicklung ist es ja sehr oft so, dass die Entwickler selber so dieses iterative Entwickeln ja kennen. Also es wird ein Teil der SW entwickelt, dann getestet, weiterentwickelt, wieder testen und sich halt iterativ vorwärts hanteln, dass sie am Ende zu einem Ergebnis kommen. Das heißt, da ist schon ein bisschen so ein Gedankengut dabei. Bei der agilen Entwicklung muss man halt dann mehrere Personen synchronisieren, dass man halt dann auf eine gemeinsame iterative Vorgehensweise kommt. Das sind so Themen, da ist es wahrscheinlich grundsätzlich besser geeignet, man muss aber trotzdem die Leute mit dieser Methode vertraut machen, dass sie alle von der gleichen Sprache reden und in der gleichen Art und Weise arbeiten. Wenn man es vergleicht mit der mechanischen Entwicklung, zum Beispiel irgendein Konstruktionsthema, das ist meistens so, dass man ein Bauteil konstruiert und man Vorgaben hat, wo dieses Teil reinpassen muss oder auch mechanische Vorgaben hat. Das ist meines Erachtens einfacher oder besser, wenn man vom Anfang zum Ende dieses Ding durchentwickelt und dann nicht immer diese Zwischenstufen macht. Das ist dann ein Aufwand der dann kommt, der einfach das Ganze zu umständlich macht. Und ich glaube auch, dass da die Denkweise von der Vorgehensweise noch sehr klassisch geprägt ist und sich die Mitarbeiter schwer tun in die flexible Arbeitsweise reinzukommen. Das sind so Grundeinstellungen zur Entwicklung.

I: Da hätten wir wieder das Mindset von Kunde und Alten natürlich. Du hast es jetzt auch angesprochen, dass wenn eine vollständige Definition von Projekttasks vorhanden ist, also man weiß was man zu tun hat, dann eignet sich agil dementsprechend nicht so stark. Sieht du irgendwelche Grundvoraussetzungen in Bezug auf die Vertragsgestaltung oder Angebotsphase.

B1: Das ist ja auch so das Thema, wenn es um Angebote oder Verträge geht, dass halt der Kunde gerne hätte, zum einen agil entwickelt, aber zum anderen einen festen Preis vom ersten Tag an hat. Das wäre somit das Schönste. Man sagt, das Ganze kostet zum Beispiel 100.000 € und am Weg dorthin kann ich noch flexibel meine Wünsche und Ziele verändern, was aber dazu führen kann, dass es keine 100.000 € mehr sind, sondern zum Beispiel 500.000 €. Weil man halt das Thema Vertrag und agil nicht immer 100 % zusammen kombinieren kann. Dieses starre Fixpreis-Angebot passt halt zu dieser agilen Arbeitsweise nicht.

I: Du hast mir nun ein paar Grundvoraussetzungen genannt. Welche von den angeführten Punkten würdest du zu den Knockout Kriterien für agiles Vorgehen zählen?

B1: Definitiv die Vertragsgestaltung. Das ist sicher ein Thema. Von der Kundenseite ist es theoretisch angenehm, von der Lieferantenseite natürlich ist es eine Grundvoraussetzung, wenn die Vertragsgestaltung nicht zur flexiblen Arbeitsweise passt, kann man es nicht machen. Das ist dann ein wirklich kritisches Thema und da werden beiden Seiten nicht glücklich, weder der Kunden noch der Lieferant, weil man immer herumstreitet und herumkämpft. Also das ist meines Erachtens wirklich ein Knockout Kriterium. Bei den anderen Themen, wie Mitarbeitertraining, Bekanntheit und Mindset würde ich jetzt nicht so sehen. Das sind eher Themen, die man auch dem Team noch beibringen kann, sind jetzt keine Knockout Kriterien meines Erachtens.

I: Wenn du an dein letztes Scrum Projekt denkst, was lief hier besonders gut und wo siehst du die Vorteile dieser Methodik?

B1: Das eine ist ein Qualitätsthema. Da ist es so, dass man in der Entwicklung 14-tägig oder monatlich, je nachdem wie die Sprints sind, immer ja schon ein lauffähiges Produkt hat. Das heißt zum einen ist es so, dass man schon relativ früh weiß, funktioniert das System in sich, weil es sind ja verschiedene Teams, wo verschiedene Leute verschiedenste Teilmodule entwickeln und zum anderen kriegt man einen Vorteil, da man

vom Kunden schon rückgemeldet kriegt, ist es das was er haben möchte. Und mit jedem Sprint wird das Produkt noch einmal reifer und besser. Das heißt, dieses Thema Qualität und das, was der Kunde möchte, ist deutlich besser dadurch geworden. Wenn man sich den Aspekt Planbarkeit anschaut, ist es so, dass man diese kurzen Phasen sehr gut planen kann. Also das heißt, ich kann gut sagen in 14 Tagen, was ungefähr kommen wird. Wo man aber darauf achten muss, dass man das Endziel nicht aus den Augen verliert. Weil oft ist es so, man bewegt sich mit den Sprints halt in eine Richtung, nach links, nach rechts und dann kommt man drauf, einige Hauptfeatures sind noch gar nicht drinnen und in 2 Wochen haben wir schon unseren Endtermin. Da braucht man immer jemanden, der das ganze Thema steuert und halt auch so dieses Ziel im Kopf behält. Und vielleicht den dritten Aspekt, wenn man in Richtung Kosten geht oder Preis auch geht, man hat eine deutliche größere Transparenz dem Kunden gegenüber. Der weiß halt immer, was los ist. Man kann zwar zum einen drinnen nicht sehr viel verstecken, dass man sagt, man macht deutlich mehr, als man eigentlich getan hat, das geht nicht, aber im Gegenzug ist es so, dass jede Zusatzleistung oder Zusatzumfang dem Kunden direkt bekannt ist und man kann dann direkt verhandeln, ob man das jetzt macht oder nicht macht, ob er das Geld dafür hat oder wie auch immer. Also da ist eine Akzeptanz vorhanden, wenn es Veränderungen gibt, was man bei so einem Fixpreisprojekt nicht hat. Weil der zahlt halt seinen Betrag und was da drinnen passiert, von dem will er eigentlich nix mehr hören.

I: Und jetzt die Gegenfrage. Wenn du wieder an dein letztes Scrum Projekt denkst, was lief hier nicht so gut und wo siehst du eventuelle Nachteile dieser Methodik?

B1: Was jetzt wirklich eine Schwierigkeit war, ist jetzt dieses Zusammenspiel mit dem Kunden, der jetzt nicht in der Scrum Methodik drinnen war. Das heißt er ist halt mit dieser Fixpreisvorstellung unterwegs, sagt aber er will Scrum haben, das heißt er hat einfach andere Vorstellungen was agil bedeutet und das ist ein starker finanzieller Aspekt. Dem Kunden hat es zwar Spaß gemacht, weil er immer sieht was wir tun und monatlich festlegen kann wo es hingehet, aber wo wir dann mit flexibler Rechnung gekommen sind, hat er sich schwergetan. Das ist ein ganz wesentlicher Punkt dabei.

I: Denkst du, hätte man das mit einem Kundentraining besser gestalten können, dass man mit Ihm im Vorfeld detailliert darüber spricht und so etwas auch im Vertragskonstrukt festhält?

B1: Auf alle Fälle. Auf alle Fälle. Wir haben ja am Anfang immer so ein internes Kickoff und ein Kunden Kickoff. Und bei dem Kunden Kickoff sollte man solche Dinge genauestens festlegen. Also ein Training mit dem Kunden zu machen, also was heißt das konkret, damit der auch versteht was diese Arbeitsweise bedeutet und gegebenenfalls sich auch noch dagegen entscheiden kann, falls er sagt, das ist nicht „seins“.

I: Dann komme ich gleich zur letzten Frage in diesem Kapitel. Gab es Projekte die agil abgewickelt wurden, die du aber mit deinem jetzigen Wissensstand nicht mehr agil machen würdest, sondern klassisch? Wenn ja, wo lagen die Gründe dafür?

B1: Ist mir so jetzt eigentlich nicht bekannt. Das einzige, was wir mal hatten, habe ich jetzt schon leicht angesprochen, das war das Projekt mit dem Widerspruch zwischen Fixpreis-Bestellung bei Kunden und agiler Entwicklung bei uns. Also da wäre im Nachhinein betrachtet, entweder was du vorher angesprochen hast, im Kickoff mit dem Kunden das besser zu besprechen, oder dann vielleicht wirklich nach Fixpreismethodik abzuwickeln, also, weil man einfach diesen Widerspruch hatte. Aber sonst bei den anderen Projekten muss ich sagen, würde ich es jetzt nicht anders machen.

I: Dann gehe ich gleich über zum nächsten Teil C. Da geht es um die Eignung der unterschiedlichen IQP-Modelle. Wie gut eignet sich aus deiner Sicht das IQP-3 Modell für agiles Vorgehen a la Scrum?

B1: IQP-3 ist bei uns das bestgeeignetste Modell, weil da ist es so, dass man ja eigentlich eine Mannschaft definiert, die dann immer auch von der Abrechnungsweise, monatsweise abgerechnet wird und immer auch monatsweise in diesem Modell die Ziele und Abnahmen definiert. Das heißt, eigentlich passt das sehr gut zu dieser Scrum Methodik dazu. Man hat immer ein definiertes Entwicklerteam, das Scrum Team kann man es auch gleich nennen in diesem Fall und definiert halt immer so in diesen Einheiten seine Ziele und Ergebnisse. Da ist es von dem ersten Tag an schon dem Kunden bekannt, dass wir jetzt nicht ein Fixpreis-Thema haben, sondern das halt einfach in Form von Größe einer Mannschaft das ganze Thema dann mitbewerten. Und deshalb ist das wirklich sehr gut auf das Thema zugeschnitten.

I: Fallen dir noch weitere Punkte ein, warum es sich gut für agil eignet?

B1: Vielleicht noch ganz kurz. Die Vertragsgestaltung passt genau dazu, dann das Thema Zielsetzung und Abnahme passt genau dazu und dann auch so die Flexibilität das Team in kurzen Einheiten zu gestalten.

I: Wie gut eignet sich aus deiner Sicht das IQP-4 Modell für agiles Vorgehen a la Scrum?

B1: IQP-4 ist so, dass man vordefinierte Arbeitspakete festlegt. Zum Beispiel Testen: klein, mittel, groß oder SW Entwicklung: einfach, mittel, schwer. Da ist es gedacht, dass man immer wiederkehrende Einheiten hat, die man dann macht. Man kann das auch für die agile Entwicklung verwenden, in dem man diese Arbeitspakete auf die Sprints zuschneidet, ist insofern aber komplizierter, weil man im Vorfeld diesen Leistungskatalog sehr genau definieren muss. Und wie agil schon sagt, kann sich dieser Leistungskatalog, den man dann in diesen Sprints schon hat, immer wieder mal verändern. Das heißt, einmal ist es vielleicht eine komplexe Funktion, das nächste Mal ist es eine mittelkomplexe Funktion und das nächste Mal ist es eine leichte Funktion und wir haben vielleicht nur leicht, mittel und komplex aber nicht mittelkomplex definiert. Und dementsprechend hat man dann immer eine gewisse Unschärfe dabei. Prinzipiell geht's, das heißt man kann ja auch dann immer diese Arbeitspakete nehmen für Sprint, es ist nur mehr an Kontrolle oder mehr an Eignung der Definition der Arbeitseinheiten. Da ist etwas mehr an Aufwand dahinter.

I: Wenn es wiederkehrende Tätigkeiten sind und ich es schon das erste Mal gemacht habe, dann weiß ich ja, um was es da geht und was es zu machen gibt. Agil ist ja eher in dem Bereich, wo man sagt, man kennt zwar ungefähr das Ziel und die Ausgangsbasis, aber wie man dann zum Ziel hinkommt ist noch nicht 100 % klar. Wohingegen beim IQP-4 Modell ist das ja doch schon ziemlich stark definiert, oder?

B1: Das Thema ist, man fängt dann an, diese wiederkehrenden Arbeitspakete sehr schwammig zu definieren. Also es gibt dann sehr oft ein Arbeitspaket, das nennt sich einfach: Entwicklung mittleren Umfangs. Und da ist nicht beschrieben, was heißt mittlerer Umfang. Heißt das, ich arbeite jetzt 10 h dran, heißt das, ich arbeite jetzt eine mittelkomplexe Funktion aus, heißt es, ich arbeite jetzt 200 Zeilen Codes ab. Das heißt, man fängt dann an, das sehr schwammig zu definieren, damit es halt für alle Fälle irgendwie passt. Weil wie du es vorher angesprochen hast, dass es wirklich definierte wiederkehrende Pakete sind, das hat man in diesem Fall ja nicht, sondern man hat ja immer nur so Arbeitseinheiten, die man halt definiert, die einen gewissen Umfang haben.

Im Prinzip versucht man, eigentlich ein IQP-3 Modell auf IQP-4 umzuwandeln, aber wo man halt sagt, man hat keine fixe Mannschaft in dem Zeitrahmen, sondern Einheiten von Leistungsgrößen oder eine Leistungsmannschaft, die man da definiert und hat aber nicht mehr ein echt definiertes Thema, was immer wiederkehrend gemacht wird.

I: Wie würdest du jetzt IQP-4 genau einstufen. Eher nicht geeignet, schwammig, ...

B1: Ich würde sagen, eher schwammig. Wie gesagt, der eigentliche Hintergedanke von IQP-4, wiederkehrende Pakete immer wieder gleich zu machen, der geht verloren. Man kann aber trotzdem irgendwie so Art Arbeitseinheiten definieren, die schwammig definiert sind. Das heißt, man kann es schon damit machen, ist für den Kunden oft besser, weil er bei IQP-3 den Eindruck hat, dass er eine fixe Mannschaft zahlt, egal, ob die was tun oder nicht. Und so hat er den Eindruck, dass er die Umfänge besser gestalten kann. Jetzt hat er zum Beispiel weniger zu tun, jetzt braucht er nicht mehr die 5 Leute, sondern jetzt zieht er 3 Tickets, wo jedes Ticket eine Person als Leistungsumfang ist. Das heißt, für den Kunden ist es ein bisschen flexibler von den Kosten her. Er hat zumindest den Eindruck, dass es flexibler ist, ist aber im Managen vom Ablauf komplexer, weil ich auf beiden Seiten schauen muss, passt das zu dem was wir jetzt aktuell machen. Sofern ist es schwieriger, aber es geht auf alle Fälle.

I: Wie gut eignet sich das IQP-5 Modell aus deiner Sicht für das agile Vorgehen?

B1: Gar nicht eigentlich. Also man hört ja schon am Namen, Fixpreis mit flexibler Entwicklung ist ja schon ein Widerspruch an sich. Ist aber so, dass der Kunde das sehr oft fordert, weil er sagt, er hat ein Budget von so und so viel und will halt in dem Rahmen das Projekt gemacht haben, aber er will es halt so haben, dass er es flexibel gestalten kann. Das heißt, das ist so für uns die Schwierigkeit einen Weg zu finden, wo man Beides unter bekommt.

I: Im nächsten Kapitel geht es um die Anpassungen die notwendig wären, oder sinnvoll wären, um IQP-3, 4 oder 5 doch für das agile Vorgehen fit zu machen. Du hast ja erwähnt, IQP-3 eignet sich sehr gut für das agile Vorgehen, würdest du trotzdem noch Änderungsbedarf sehen oder ein Verbesserungspotential, um noch bessere Ergebnisse zu erzielen in Bezug auf den Kunden, das Projektteam, die Organisation oder die Prozesslandschaft?

B1: Was sicher ist, ist das Thema Prozessgestaltung bei uns. Wir haben ja die IQP Modelle bei uns schon beschrieben, aber eher so im Sinne von Abrechnung und Angebotsgestaltung. Wie man aber konkret jetzt das in der Praxis in der Entwicklung umsetzt, da haben wir noch relativ wenig beschrieben, bzw. festgelegt von der Prozessseite her. Das ist eher intuitiv, wie es halt die Leute dann machen. Beim IQP-3 ist es ja dann so, dass es schön dazu passt, insofern nicht so schwierig ist. Aber trotzdem, da wäre es ganz schön, wenn wir bei uns so ein Art Referenzprozess hätten, oder Vorschlag, oder Best Practice, wie man es auch immer nennen möchte. Wo man sieht wie macht man idealerweise bei IQP-3 dann die agile Entwicklung. Aber prinzipiell denke ich so grundsätzlich ist es ein sehr gutes Modell, wo man jetzt nicht so viel herumschrauben muss. Das ist eher so, dass man Leuten was an die Hand gibt, um zu sehen wie es am besten oder effizientesten funktioniert.

I: Siehst du Änderungsbedarf in Bezug auf den Kunden?

B1: Dieses Thema Kickoff mit dem Kunden gemeinsam, also was wir da schon angesprochen haben kurz. Das Thema Training finde ich eine super Idee, das man dem Kunden beim Projektstart noch einmal einfach klar macht, wie das Ganze abläuft. Was es genau im Detail bedeutet, auch für Ihn. Dass man einfach ein gleiches Verständnis hat für das Thema. Das Thema Abrechnung ist ganz gut über das Modell festgelegt, das passt eigentlich ganz gut.

I: Du würdest jetzt an der Vertragsgestaltung oder Angebotsphase speziell für ein agiles Projekt bei IQP-3 nichts ändern? Oder es wäre nicht zwingend notwendig?

B1: Genau. Da würde ich nichts sehen jetzt direkt. Das passt wirklich ganz gut.

I: Auch nicht vielleicht die Scrum Nomenklatur in den Vertrag mit rein zu geben?

B1: Also wir haben es momentan drinnen, dass wir, wenn wir ein Scrum Projekt haben, ein gewisser Textblock drinnen ist. Das sind meistens so 2, 3 Seiten, da wird mal grundsätzlich die Scrum Methodik beschrieben, dann auch noch ein bisschen Referenz was wir dazu schon gemacht haben. Also das ist schon im Angebotstext bei Scrum Projekten schon mit drinnen.

I: Gehen wir gleich zu IQP-4 über. Da haben wir gesagt, es hat eine mittlere Eignung für Scrum. Dementsprechend könnte es ja Anpassungen oder Änderungen geben, die man machen könnte, damit das agile Vorgehen noch besser funktioniert.

B1: Das Kundentraining sowie Scrum Team Training gilt für alle IQP Modelle, muss also nicht nochmal im Detail angeführt werden. Was bei der Vertragsgestaltung sehr wichtig ist, ist die Arbeitspakete zu definieren. Was ich vorher angesprochen habe, die immer sehr weich besprochen werden, so groß, mittel, klein. Das, was alles bedeuten kann und das ist glaub ich ein Thema, wo man sich wirklich Gedanken machen muss, wie macht man das bei dieser Scrum Entwicklung. Da habe ich jetzt auch keine spontane Lösung dafür, aber das ist ein Thema, wo man konkret sich überlegen kann, was ist da die beste Definition der Arbeitspakete oder Arbeitseinheiten, dass es für den Kunden passt, dass es für uns passt, dass es aber auch keine Widersprüche gibt und es einigermaßen gut definiert ist. Weil es ist generell so bei IQP-4, der Erfolg bei IQP-4 hängt sehr stark von dieser Leistungskatalog- bzw. Arbeitspaketkatalogdefinition ab. Und noch einmal mehr, wenn es ein agiles Projekt ist, weil man ja eigentlich diesen Katalog nicht starr definieren kann, sondern man braucht eine Flexibilität dahinter, sollte aber trotzdem einen starren Weg finden, weil man sonst in diese Einheiten nicht viel reingießen kann. Und diesen Spagat bzw. diese Optimierung zu finden, ist ein wichtiges Thema.

I: Fällt dir bei IQP-4 noch irgendwas bezüglich Prozesslandschaft ein?

B1: Das ist das gleiche wie bei IQP-3. Also auch hier ist es so, dass das Modell sehr stark auf Angebot und Abrechnung ausgelegt ist, von der Beschreibungsseite her. Da wäre auch noch wichtig, diese Verbindung zu schaffen, wie es dann konkret in der Entwicklung aussieht. Also auch so Best Practice oder Grundgerüst, damit man sagen kann, so oder so macht man es bei IQP-4.

I: Dann gehen wir gleich zum letzten Punkt über, IQP-5. Da haben wir schon gesagt, dieses flexible und starre schneidet sich, das funktioniert nicht. Aber vielleicht gibt es Möglichkeiten, um dieses doch dem agilen Vorgehen anzupassen. Siehst du irgendwo Möglichkeiten?

B1: Da ist ja der Widerspruch zwischen wie wir intern arbeiten und dem, wie wir zum Kunden hin agieren. Der Kunde hätte gerne, dass wir flexibel arbeiten, will aber starr abrechnen. Das ist der Weg, wo man sich überlegen müsste, wie kann man die beiden Dinge zusammenbringen, wie kann man das in dem Rahmen machen. Zum Beispiel, wir

arbeiten agil, aber nur bis zu einem gewissen Budgetrahmen. Er hat zum Beispiel auf 100.000 € bestellt. Wir arbeiten agil, solange bis die 100.000 € aufgebraucht sind, er kriegt regelmäßig eine Aussage, wie viel er noch übrig hat und dann kann der quasi gestalten im Rahmen seines Fixpreises, wie viel das wir agil noch weiterentwickeln bzw. ob man rechtzeitig das Ganze beendet. Das heißt, auch wieder Richtung Transparenz zu gehen.

I: Der Kunde stellt ja am Anfang gewisse Features, die er haben will und bei dieser Scrum Methodik ist es ja sehr wichtig zu priorisieren. Da geht es ja eigentlich immer darum, zu schauen was ist am wichtigsten, was hat den größten Geschäftswert für den Kunden und da versucht man dann natürlich auch mit jenen Features zu starten, die den höchsten Wert haben. Wäre es dann nicht zum Beispiel möglich, im Vertrag so eine Art „exchange for free“ zu hinterlegen? Also der Kunden sagt, er möchte doch noch etwas Anderes haben. Alten sagt, dass kannst du gerne haben, dafür musst du aber etwas aus deinen Anforderungen streichen, das ungefähr den gleichen Aufwand hat.

B1: Das ist eine gute Idee. Das finde ich sehr gut. Was damit aber auch verbunden ist, ist, dass er immer auch den Vertrag mit uns verändern muss. Wenn man es rein formell sieht, ist in diesem Vertrag vom Werkvertrag inhaltlich beschrieben, was wir zu tun haben. Und beim IQP-5 ist das ja sehr detailliert beschrieben. Das heißt, wenn wir so einen Tausch machen, Funktion 1 machen wir nicht, stattdessen gleichwertig Funktion 2, dann muss man halt jedes Mal den Vertrag anpassen, rein formell gesehen. Aber prinzipiell ist es möglich. Finde ich keine schlechte Idee. Dann muss man kein Geld verhandeln, weil das ist oft sehr unangenehm, für beiden Seiten. Sondern man diskutiert über Leistungsumfänge. Ok, dann ist uns das nicht so wichtig, dann machen wir lieber das Andere, hat aber auf den Endpreis keine Auswirkung. Immer, wenn es um Kosten oder Preise geht, ist es auch für den Kunden oft schwer intern zu argumentieren.

I: Dann gehen wir zum letzten Kapitel E über. Hier geht es um den allgemeinen Scrum basierten Prozess. Wenn du an dein letztes agiles Projekt denkst, hast du das laut dem in der Literatur beschriebenen Scrum Prozess durchgeführt, oder gab es hierbei Abweichungen in Bezug auf die Meetings, die Rollen und Artefakte?

B1: Im Detail kann ich jetzt nicht alles beantworten. Aber wo ich es weiß, das ganze Backlog Thema wird so betrieben wie es in der Literatur beschrieben ist. Wir haben auch

meistens Product Owner und Scrum Master. Manches Mal ist es aber so, dass dies in einer Person ist, vor allem bei kleineren Projekten.

Interview 2

I: Was sind deiner Meinung nach negative Einflussfaktoren für den Projekterfolg bei klassischer Projektabwicklung?

B2: Meiner Ansicht nach, derzeit, sind die größten Probleme A: Technologiewachstum, B: wechselnde Anforderungen der Kunden. Beispiel: Ein Kunde möchte ein Produkt entwickeln. Die Entwicklung dieses Produktes dauert nach Einschätzung ca. 1 Jahr. Das ist ein sehr kurzfristiger Entwicklungsprozess, normalerweise hat man 3 Jahre. In diesem einem Jahr kann A: Technologie gewechselt werden. Bsp: Du hast einmal Android, du hast einmal iOS. Du schreibst klassisch ein Lastenheft. Da drinnen beschreibst du nicht nur was gemacht werden müsste, sondern auch welche Technologien dafür verwendet werden sollten. Dann stellt sich heraus, dass es eine bessere Alternative gibt um das Programm zu schreiben. Was noch schlimmer ist, ist B, dass der Kunde sagt, wir haben eine Marktstudie durchgeführt und es ist nicht mehr interessant oder relevant, dass dieses Produkt die Features A, B und C, die wir als hochpriorisiert haben, sondern lediglich D, E und F entwickelt werden. Oder eine Marktstudie besagt, dass die Zielgruppe nicht mehr 14-25 Jahre alt ist, sondern 18-65, weil doch mehr ältere Menschen Smartphones benützen. Durch die sehr schnelle Umgestaltung, kann ich in einem klassisch gemanagten Projekt diese Änderungen nicht vornehmen. Weil wir starten bei A und hören erst bei Z auf. Wenn du ein Jahr später mit Z fertig bist, dann entspricht das nicht den Anforderungen vor einem halben Jahr, sondern den Ausgangsanforderungen. Das ist das größte Problem, das ich hier sehe. Es kann auch sein, dass sich in der Pilotphase herausstellt, dass das was entwickelt worden ist, nicht das ist, was die Zielgruppe braucht. Man hat es gelb entwickelt, die Zielgruppe möchte aber eine andere Farbe. Früher hat man die Pilotphase erst nach Abschluss des Projektes gestartet. Alphatester waren nur beim Abschluss des Projektes involviert. Hier haben wir die Möglichkeit durch SCRUM in Zyklen MVP (Minimum Value Product) zu erstellen und die Features für diese MVPs bereitzustellen, damit die Testgruppen die Produkte testen können.

I: Du hast jetzt ein Paar negative Einflussfaktoren genannt. Welche von denen würdest du zu den Hauptproblemen zählen?

B2: Falsches Verständnis der Kundenwünsche. Nicht, dass die Kundenwünsche falsch aufgenommen werden, sondern, dass diese in Wochen oder Monaten ganz anders sein können.

I: In vielen Unternehmen gibt es jetzt einen Trend in Richtung agile Abwicklung. Siehst du den Trend eher bei Alten oder beim Kunden?

B2: Mittlerweile ist es beidseitig. Agile SW Entwicklung ist von Alten seit 2017 aktiv gepusht worden in sämtlichen Projekten die ich jetzt kenne. Leider kann ich nicht das gesamte Spektrum von Alten überblicken. Aber das was ich bis jetzt erlebt habe, möchte Alten proaktiv, dass die Projekte in agiler Weise umgesetzt werden.

I: Befürwortet das auch der Kunde?

B2: Ja, definitiv. Lediglich was mich etwas stutzig macht, ist, dass nicht jeder versteht, was agil bedeutet. Der Kunde möchte das, weil es oben das Signal gibt, mach mal agil, aber nicht versteht, was agil bedeutet. In den meisten der Fälle. Das hat aber auch damit zu tun, dass wir größere Kunden haben, die jetzt in den letzten 25 Jahren auf ihre eigene Weise gearbeitet haben. Die sagen, warum sollten wir etwas ändern, obwohl es bis jetzt super funktioniert hat.

I: Wo sieht Alten die größten Vorteile bei der agilen Abwicklung? Welche oben genannten negativen Einflussfaktoren könnte man mit der agilen Abwicklung kompensieren?

B2: Bei der agilen SW Entwicklung ist der Product Owner meistens der Kunde. Da ist natürlich viel Verantwortung auf seiner Seite. Hier einen Fehler auf Alten Seite zu machen, ist sehr viel kleiner. Zweitens ist die Hauptverantwortung auf Kundenseite. Das ist natürlich für uns ein enormer Vorteil. Ein weiterer Vorteil sind viel schnellere Rückmeldungen vom Kunden. Du hast Feedback vom Kunden, sowie Sprintabnahmen (2 oder 4 Wochen). Gehen wir in die richtige Richtung, machen wir tatsächlich, was von uns erwartet wird. Das ist natürlich ein sehr großer Vorteil für Alten. Seitens Kunden kann man sehr schnell auf Marktänderungen reagieren. Man kann die neuesten Trends sofort mitnehmen. Hier sehe ich einen klaren Vorteil für den Kunden.

I: Denkst du gibt es überhaupt gewisse Grundvoraussetzungen die es benötigt, damit eine agile Projektabwicklung Sinn macht?

B2: Bei Immobilien ist es immer Lage, Lage, Lage. Bei der agilen SW Entwicklung ist es Team, Team, Team. Die agile SW Entwicklung stellt das Team in den Vordergrund. Teams über Lastenheft. Aber auch das Wort Selbstmanagement ist sehr wichtig. Eine weitere Grundvoraussetzung ist eine bestimmte Größe des Teams. Mit 2 Mitarbeiter kannst du kein agiles Projekt aufsetzen. Aber auch nicht mit 11. Ab 3 – 8 macht es definitiv Sinn. Die Grundvoraussetzung ist das Team selbst. Das Team muss agil sein. Die Gedanken müssen agil sein. Sie müssen selbst wissen, was erwartet wird und proaktiv im Entwicklungsprozess teilhaben. Das ist das Wichtigste. Die Mitarbeiter sind der Kern in der agilen Entwicklung. Ebenfalls wird vorausgesetzt, dass der Product Owner weiß wohin es gehen soll. Dieser ist auch ein Teil des Teams. Das wird sehr oft missverstanden. Es ist oft nicht leicht, dem Kunden zu vermitteln, dass wir alle zusammen in einem Team sind. Wir arbeiten alle zusammen. Dieser Aspekt wird sehr oft vernachlässigt.

I: Denkst du ist bei einem agilen Projekt die Angebotsphase oder Vertragsgestaltung anders zu machen als bei einem klassischen Projekt?

B2: Definitiv. Viel einfacher. Es muss ein gewisses architektonisches Skelett gebaut werden, auf dem agil aufgesetzt werden kann. Aber sobald diese Voraussetzung erfüllt ist, ist natürlich auch die Vertragsgestaltung viel einfacher. Einen Vorteil sehe ich dadurch in den definierten Rollen. Der Product Owner schafft an, die Entwickler entwickeln und der Scrum Master versucht zwischen allen Parteien zu vermitteln.

I: Ich greife jetzt ein bisschen vor. Gibt es dieses architektonische Skelett schon für agiles Vorgehen?

B2: Agile Strukturen müssen gegeben sein, sonst ist das ein chaotischer, selbstentscheidender Organismus, der macht was er möchte. Selbstverständlich muss dieses Skelett vorhanden sein. Meistens wird dies mit Jira oder Trello gemacht, die eine gewisse Struktur reinbringen. Da wird gesagt, was wir machen möchten, da wird konkret unterteilt, was gemacht werden muss, in welchem Zeitraum. Also diese Struktur muss vorhanden sein. Agil heißt nicht, dass wir die Wilden sind und wir nach Lust und Laune entwickeln. Hier sind auch Regeln drinnen, die befolgt werden müssen.

I: Gibt es deiner Meinung nach Knockout Kriterien für agiles Vorgehen?

B2: Ja die gibt es definitiv. Der Kunde versteht nicht was agil bedeutet. Für diesen ist agil kleine Lastenhefte und fixierte Aufwandsabschätzung. Die Aufwandsabschätzung ist

aber nur eine Estimation. Es ist nicht in Stein gemeißelt. Ein weiteres großes Knockout Kriterium ist, wenn das Team selbst nicht agil ist. Das Team muss sich selbst organisieren und die Kommunikation im Team muss stimmen. Wenn das Team davon ausgeht, dass der Scrum Master und Product Owner die Leitung übernehmen, dann kann man hier nicht mehr von Scrum sprechen.

I: Denkst du, gibt es auch Knockout Kriterien in Bezug auf die Angebots- und Vertragsgestaltung?

B2: Die Praxis zeigt, dass die Vertragsgestaltung bei einem ineffektiven Projekt zu Problemen führen kann. Wenn alles Geforderte geliefert wird, dann gibt es hier natürlich auch keine Reibereien in Bezug auf den Vertrag. Der Kunde muss von Anfang an verstehen, dass er agil arbeitet. Das ist sehr wichtig. Sollte das für den Kunden nicht verständlich sein, was die agile Entwicklung bedeutet, dann ist er vielleicht nicht reif dafür. Vertragsgestaltung ist, glaube ich, nicht das größte Problem, sondern das Mindset der Menschen.

I: Wenn du an deine letzten Scrum Projekte denkst, was lief hier besonders gut und worin siehst du den Vorteil dieser Methodik?

B2: Dass die Mitarbeiter mitentscheiden können wie etwas entwickelt wird. Das gibt Ihnen mehr Freiheit. Vorausgesetzt er ist ein erfahrener Entwickler. Ein weiterer Vorteil ist, dass wir schnell was umsetzen können, dass der Kunde auch rechtzeitig die Richtung ändern kann. Das ist etwas anspruchsvoll, hat aber auch etwas Lebendiges. Man genießt den Tag und denkt nicht jeden Tag über den Tod nach, wie es beim Lastenheft sehr oft der Fall ist. Was passiert in 2,5 Jahren.

I: Hast du auch Nachteile gesehen?

B2: Nachteile auf Seiten des Kunden sind, dass dieser nicht versteht, dass agil nicht kleine Lastenhefte sind und das auch bei agiler Vorgehensweise manchmal ein Code oder Sachen auf Alten Seite angepasst oder geändert werden müssen. Oft fragt der Kunde dann, warum habt ihr das nicht gleich so gemacht. Für den Kunden ist eine agile Entwicklung eine eierlegende Wollmilchsau. Wenn er möchte, kann er ein Team kleiner oder größer machen. Wenn neue Features kommen, können wir diese gleich umsetzen. Wenn aber dann auf Alten Seite Änderungen notwendig sind, weil Informationen nicht vorhanden waren, dann möchte der Kunde nichts damit zu tun haben. Kommt auch vor.

Woher kommen diese Bugs? Ich dachte eigentlich, dass wir mit Sprint 17 eine fertige Version haben. Warum haben wir noch Bugs drinnen? Ja, weil aufgrund der nicht voll zu Ende gedachten Architektur Probleme aufgetreten sind. In einem HW Entwicklungsprojekt entwickelt man oft nur 20-30 % der Teile neu, viele können übernommen werden. In der SW Entwicklung ist das Produkt meist für den Kunden spezifisch entwickelt. Dafür gibt es oft vorher keine Lösungen, da muss man sehr oft erst eine Lösung dafür entwickeln.

I: Gab es Projekte, die du im Nachhinein eher klassisch machen würdest?

B2: Ja, gab es definitiv. Aber hierfür muss eine Vorarbeit geleistet werden. Ein Lastenheft muss erstellt werden. Um ein Lastenheft zu erstellen, brauchst du eventuell 2 Wochen Zeit, oder eventuell 3 Monate. Das kommt auf die Größe des Projektes an. Bei kleineren Projekten macht das schon Sinn, dass man so Mini-Lastenhefte erstellt, aber bei größeren Projekten, alles was in den nächsten 3 Jahren entwickelt werden soll, macht es meiner Meinung nach weniger Sinn. Wenn du für öffentliche Strukturen, Regierungen zum Beispiel, dann macht das eventuell schon Sinn, nach Lastenheft vorzugehen. Du hast ein Dokument, was das enthält, das am Ende rauskommen soll. Bei der agilen Methode hast du das nicht. Du hast einen Vertrag, der sagt, dass du agil entwickelst und die Agilität wird durch die Story Points bzw. durch Sprints dargestellt.

I: Wie gut eignet sich aus deiner Sicht das IQP-3 Modell für agiles Vorgehen?

B2: Perfekt. Du rechnest nach geleisteter Arbeit ab und dementsprechend kannst du ein Team aufstocken oder abstocken.

I: Wie sieht es mit IQP-4 aus?

B2: Bei IQP-4 kann ich mir das auch vorstellen. IQP-4 sagt aber, dass es wiederkehrende Aufgaben sind. Ist ein Sprint jetzt eine wiederkehrende Aufgabe oder nicht? Das ist die Frage die ich mir auch schon des Öfteren gestellt habe. Für mich eventuell ja, weil ein Sprint ist zum Beispiel etwas, das ich messen kann, alle 2 oder 4 Wochen. Und der Kunde ruft dieses Ticket zyklisch ab. Diese Tickets könnten auch nach Schwierigkeitsgrad definiert werden. Leichte, mittlere und schwierige Aufgaben. Und dann können diese vom Kunden nach Bedarf abgerufen werden. Wenn du aber hier plötzlich das Team vergrößern musst, damit du die Aufgaben bewältigen kannst, dann

funktioniert das nicht mehr. Und die agile SW Entwicklung ist auch nicht bedarfsgesteuert, sondern eine kontinuierliche Entwicklung.

I: Wie sehen sie die Eignung von IQP-5?

B2: IQP-5 ist für mich ein typischer Werkvertrag. Das hat meiner Meinung nach nichts mit Scrum zu tun. Ich lasse mich gerne des Besseren belehren. Aber IQP-5 ist das klassische Lastenheft, wo man alles vorher definiert. Weiteres wird bei IQP-5 ein Start und Ende definiert. Dies ist jedoch bei agil nicht so einfach zu sagen. Bei agil machst du schon gewisse Abschätzungen, die aber vertraglich nicht festgesetzt sind.

I: Wenn du an dein letztes agiles Projekt denkst, hast du dieses nach dem in der Literatur genannten Prozess durchgeführt, oder gab es hier Abweichungen in Bezug auf die Meetings, Rollen und Artefakte?

B2: Die Artefakte bleiben unangetastet, die Rollen sind ebenfalls klar. Wie der Scrum Prozess läuft, ist nicht immer rund. Hier muss immer zwischen Inhouse und externe Projekte unterschieden werden. Bei internen Projekten kann man eher nach Literatur vorgehen. Bei externen Projekten gibt es einfach mehr Variablen. Oft wünscht der Kunde eine Verschiebung von Meetings oder auch intern gibt es immer wieder Abweichungen. Die Sprintdauer wurde manchmal vom Kunden verändert, aufgrund von eigenen Terminen oder Messen. Sprint Reviews wurden manchmal weggelassen. Manchmal wünschte der Kunde auch Änderungen während dem eigentlichen Sprint. Irgendwann wurden diese Änderungen dann Normalität. Aber manchmal muss man einfach dem Wunsch des Kunden nachkommen. Aber das große Problem ist, dass der Kunde nicht zu 100 % versteht, was agil bedeutet. Er sagt was zu tun ist und will das sofort umgesetzt haben. Scrum funktioniert zu 100 %, wenn du für dich selbst ein Produkt entwickelst. Wenn du ein Produkt im Namen des Kunden entwickelst und der Kunde Änderungen haben möchte, dann sind wir in erster Linie Berater. Aber den Kunden zu seinem eigenen Glück zu zwingen, indem wir Ihm sagen, wie er den Prozess handeln muss, können wir leider nicht.

I: Also deiner Meinung nach ist es eine Gratwanderung, um einerseits den Kunden zufrieden zu stellen und andererseits mit den Änderungen nicht zu tiefe Schnitte im Projekt zu hinterlassen. Denkst du, dass jede Abweichung vom Scrum Prozess zu Nachteilen führt?

B2: Abweichungen vom Prozess heißt schlechte Planung. Also ja, ich denke, dass es zu Nachteilen führt. Der Product Owner, also der Kunde hat nie verstanden was die Rolle des Scrum Masters ist. Wenn ich sage nie, dann meine ich auch nie. Jedes Mal, wenn der Kunde ein agiles Projekt bestellt, geht er davon aus, dass der Scrum Master der Projektleiter ist. Dieser ist aber nach agiler Entwicklung nie der Projektleiter. Scrum Master ist derjenige, der Kaffee kocht. Für den Kunden als auch für das Team. Er erarbeitet Lösungen im Falle von Problemen und schafft Hindernisse aus dem Weg. Aber es ist nicht seine Aufgabe das Projekt zu leiten. Er bestimmt nicht die Ziele, die Velocity oder auch Deliverables hängen nicht vom Scrum Master ab. Das ist aber bis jetzt nie verstanden worden.

I: Denkst du will es nicht verstanden werden oder wird es einfach nicht verstanden?

B2: 80/20. Aber nicht die guten 80/20. 80 % verstehen es nicht und 20 % möchten es nicht verstehen. Hier möchte ich nochmals auf die Hauptprobleme kommen. Vielleicht ist ein weiteres großes Problem, dass der Product Owner seine Rolle nicht versteht.

I: Gibt es abschließend noch Änderungswünsche, Ideen von deiner Seite?

B2: Vielleicht wäre in der Angebotsphase ein eintägiger Workshop zur agilen SW Entwicklung nicht schlecht. Was ist die agile SW Entwicklung? Wie arbeitet Alten? Ab 100, 200, 300k würde sich das anbieten. Es ist vor allem ein Vertrauenszeichen für den Kunden, das er versteht, dass wir unser Zeug verstehen Best Practices darzustellen. Wenn wir jetzt auf eine andere Ebene kommen wollen, dann müssen wir einen Mehrwert bei gleichem Preis dem Kunden anbieten. So ein Workshop wäre ein Siegel der Qualität. Jetzt könnte diese Idee natürlich auf taube Ohren stoßen. Denn wer hat schon Zeit für das. Aber wenn wir auf eine andere Ebene kommen wollen, dann müssen wir uns halt diese Zeit nehmen. Es würde auch unser Risiko minimieren. Es gibt immer Nischen, die gefüllt werden müssen. Ein Vertrauen sowie Transparenz von Anfang an zu schaffen, kann nur ein Vorteil sein.

Interview 3

I: Was sind deiner Meinung nach negative Einflussfaktoren für den Projekterfolg bei klassischer Projektabwicklung?

B3: Requirements sollten ja, bis auf wenige Ausnahmen, anfangs immer zur Verfügung stehen, aber im Automobilbereich hat man bis kurz vorm SOP noch Anforderungen und das ist manchmal sehr schwer zum handlen. Weil das hat natürlich Einfluss, wenn es ein spezielles Requirement ist, schlägt es durch auf andere Requirements. Das halte ich für ein großes Problem. Permanente Anforderungsänderungen. Auch die Art wie sie entwickeln. Das sie auch anfangs Richtung Feature Complete zielen, also sie haben ein relativ grobes Anforderungsgerüst und sagen, den Rest müsst ihr mit eurer Erfahrung selbst ausfüllen. Dann darf man das entwickeln, dann sagt man irgendwann mal, man ist jetzt Feature Complete und danach beginnt die Integration ins Fahrzeug und da beginnt dann das große Zahnklappern. Und dann merkt man, welche Anforderungen anfangs falsch gestellt wurden und welche jetzt alle nachkorrigiert werden und dieser Ansatz, dass ich eben Features nicht bis zum Ende entwickle und dann die nächsten Features draufsetze, sondern dass ich irgendwann mit einer Buggy Software Feature Complete bin und dann versuche, innerhalb einer zweiten Phase die Fehler raus zu machen, macht es dann auch schwierig vorhersagbar zu werden. Irgendwann erklärt man dann, jetzt ist gut genug, weil SOP ist vor der Tür.

I: Gibt es noch andere Einflussfaktoren, die du hier siehst?

B3: Es ist nicht immer ganz einfach, wenn man ganz geradlinig klassisch entwickelt und dann die Change Requests reinprasseln, noch genau festzustellen als Projektleiter, wie weit man eigentlich ist. Also die Fortschrittsermittlung ist nicht immer ganz simple. Und in dem Zusammenhang auch noch, was haben wir budgetmäßig überhaupt im Topf und die Abgrenzung. Es sind oft Änderungen an einem bestehenden Requirement und wenn man das Team nicht richtig steuert, dann sagen die ganz schnell mal, ja das machen wir kurz mit. Und dieses kurz mitmachen summiert sich halt und am Ende platzt dann das Budget. Das ist auch ein Problem, das ich sehe, das wir in der Vergangenheit auch sehr viel hatten.

I: Siehst du den Trend der agilen Entwicklung eher beim Kunden oder bei Alten?

B3: Ich würde sagen bei Alten kommt sie eher vom Kunden, ist kundengetrieben. Es gibt auch Ansätze von unten, dass Projektleiter für sich sagen, ich möchte mein Team jetzt begrenzt, sozusagen vom Kunden versteckt, aber intern agil steuern. Das ist aber dann eher die persönliche Initiative des Projektleiters und nicht von Alten gesteuert. Das sind aber dann auch eher Elemente aus der agilen Entwicklung.

I: Also das Grobkonstrukt ist nach wie vor klassisch, aber man versucht möglichst viele agile Techniken miteinzubringen?

B3: Genau, was man ebenso unter der Haube unterbringen kann. Das ist dann eben die Eigeninitiative des Projektleiters, ansonsten kommen eben die Vertragsanforderungen vom Kunden, dass er sagt wir kaufen bei euch ein Team ein, dass bei uns folgendermaßen in unserer agilen Wertschöpfung eingegliedert wird und da fügt ihr euch bitte mit ein.

I: Warum fordert der Kunde Agilität und wo sieht der Kunde die Vorteile?

B3: Wenn er es richtig verstanden hat, dann vor allen Dingen aus der leichten Aussteuerung. Das er eben nicht am Anfang ein komplettes Pflichtenheft über den Tisch werfen muss, sondern dieser ganze Change-Management Prozess der nachher ohnehin passiert, leichter gemanagt werden kann. Hat ja auch Einfluss auf den Einkauf, Lastenhefte müssen genehmigt werden, müssen in den Einkauf, also der ganze Prozess ist länger. Ich glaube halt auch, dass einige Kunden den Hype folgen und man merkt, dass sich viele noch gar nicht intensiv damit auseinandergesetzt haben.

I: Wie viele Kunden verstehen den agilen Ansatz?

B3: Ich glaube, dass es zwischen Verstehen und der Implementation in einem Unternehmen noch ein riesen Sprung ist. Verstehen tun es viele, aber wenn man agil entwickeln möchte, beginnt es im Management und ich glaube da machen es sich viele Firmen zu einfach und versuchen das in Projekten zu etablieren, ohne dass das Management abgeholt wird und das verursacht Probleme.

I: Wo sieht Alten die Vorteile agiler Abwicklung?

B3: Was Agiles alles leichter macht, kann man auch alles im klassischen Projektmanagement etablieren, aber was im agilen Frameworker schon eingelebt ist, ist häufigere Releases an den Kunden und frühes Feedback. Der Kunde ist permanent eingebunden. Also wie gesagt, das lässt sich auch alles im klassischen Projektmanagement etablieren, auch wenn ich jetzt nicht agil, sondern eher iterativ inkrementell ausliefere. Dann habe ich auch bei einem Projekt über 2 Jahre das entsprechende Feedback. Also man kann auch bei klassischen Verträgen und klassischem Projektmanagement diese frühen Feedbackschleifen etablieren, aber bei Scrum ist es halt ein wesentlicher Bestandteil.

I: Also Kundenzufriedenheit und höhere Qualität. Könnte man das so sehen?

B3: Qualität wird nicht zwangsläufig höher. Also das einzige was ich habe, wenn ich eine Definition of Done habe, ich habe eine definierte Qualität, die ich bei jedem Release ausliefere aber auch das erziele ich mit klassischem Projektmanagement genauso. Ich sehe noch einen Vorteil. Man hantelt sich von Sprint zu Sprint. Und wenn jetzt festgestellt wird, dass ein Requirement oder Änderungen vom Kunden gewünscht werden, verschiebt ja letztendlich das Backlog, was da ist und da hat man viel mehr Möglichkeiten zu priorisieren und wir haben mehr Möglichkeiten, unsere erkennbaren Mehrwerte zu positionieren.

I: Denkst du, gibt es gewisse Grundvoraussetzungen, die es benötigt, damit eine agile Projektabwicklung Sinn macht?

B3: Ja die gibt es. Beteiligung des Managements sowohl beim Kunden als auch bei Alten. Die Manager müssen bis relativ weit oben wissen, was das heißt und mitspielen. Auch die Business Manager müssen diese agile Arbeitsweise verstehen, weil das Kundenfeedback auf eine andere Art und Weise kommt, direkter kommt und die Interaktion der Business Manager auch kaufmännisch eine Andere wird. Das muss der Business Manager halt verstehen und mittragen.

I: Gibt es auch Grundvoraussetzungen auf Kundenseite?

B3: Ja, der größte Hemmschuh, so wie wir es erleben, der Fachbereich ist meistens relativ schnell bereit mit frühen Feedback und Agilität zu arbeiten, aber wer da immer von seiner Macht beschränkt wird, ist der Einkauf. Der ist halt gewohnt, immer noch 10 % raus zu handeln und wenn der Scope nicht mehr fest ist, kann er das eben nicht. Dann ist einer seiner wesentlichen Stellschrauben weg. Ich denke, die Beteiligungskette an einem Projekt, die wir so oder so haben, wie Management, Einkauf, die müssen die agilen Prinzipien verstehen. Das muss bewusst sein, klar sein, was da wie in welcher Form geliefert wird und was sich gegebenenfalls auch durch Veränderung des Scopes daraus ergeben kann. Den Leuten muss das klar sein, ansonsten geht das schief. Die Erwartungshaltung ist damit eine ganz andere. Das zweite große Hindernis, das ich sehe, ist das Vertragswerk. Scrum erfordert viel Kommunikation und fordert Offenheit. Und wenn ich jetzt irgendwo einen Vertrag dazwischengeschaltet habe, der die Offenheit nicht intensiv fördert, sondern blockiert, dann wird es scheitern.

I: Könntest du das bitte noch näher beschreiben?

B3: Ein Beispiel, wenn wir jetzt ein IQP-3 Projekt machen, wo wir sagen wir mal kein komplettes Team stellen, sondern der Kunde selber noch 2 Experten in dem Team platzieren möchte und wir den Rest stellen. Mal angenommen, wir stellen den Scrum Master, dann wäre es mir ganz recht, wenn der Kunde den Product Owner stellt, weil er den Scope definiert. Spannend jetzt aber eine Retrospektive im Team, wenn da zwei Leute vom Kunden drinnen sitzen, dann kann man so eine ehrliche Aussprache, was hat funktioniert, was hat nicht funktioniert, nicht mehr so durchführen, wie man das hätte gekonnt, wenn die alle von der gleichen Firma sind. Das zweite ist das Abrechnungsmodell. Wenn jetzt die Kundenspione sagen, ihr anderen macht ja nichts, wird sich das womöglich abrechnungstechnisch niederschlagen und es wird sich vor allem auf die Teammoral niederschlagen und die Kommunikation im Team. Also die Offenheit die man sich eigentlich wünscht, hätte in dem Fall sofort irgendwelche Vertragskonsequenzen und wird dementsprechend anders gemanagt werden müssen, als in einem homogenen und gleichberechtigten Team.

I: Welche der genannten Voraussetzungen würdest du als Knockout Kriterien bezeichnen?

B3: Ein Kriterium haben noch gar nicht genannt. Das ist, wenn agil nicht agil genug ist. Also, wenn ich zum Beispiel in der Bug Fixing Phase bin, dann kann ich keine Sprintlänge von einer Woche Länge definieren, weil morgen kommt der höchstpriorisierte Bug rein, dann wird meine komplette Planung hinfällig. Deswegen gibt es ja die situativen Ansätze wie Kanban. Das wäre für mich das größte Knockout Kriterium, wenn ich noch nicht mal eine Sprintlänge Ruhe habe und den Scope festhalten kann. Das ist eigentlich das Wichtigste, danach eben solche Teamsetup Szenarien. In welcher Umgebung agiert das Team und wie ist es zusammengesetzt. Also eben vertragliche oder organisatorische Themen. Manchmal auch persönliche Themen, dass ich Leute im Team habe, die mit dieser Offenheit nicht umgehen können oder sich nicht gegenseitig mit Respekt begegnen können. Wichtig ist ebenfalls, dass der Scrum Master aktiv beim Team ist und sein Ohr an der Schiene hat. Ein erfahrenes Team kann auch ohne Scrum Master funktionieren, aber ein junges Team, das noch nie zusammengearbeitet hat und Scrum nicht kennt, braucht einen Scrum Master, der führen kann.

I: Wenn du an dein letztes Scrum Projekt denkst. Was lief hier besonders gut und wo siehst du die Vorteile dieser Methodik?

B3: Die Projekte, die ich hatte, sind alles Scrum Projekte unter der Haube. Das heißt der Kunde hatte kein Scrum bestellt und ich habe dann halt das Team intern so gemanagt mit dem, was ich so noch zur Verfügung hatte. Die größten Vorteile für mich waren immer das frühe Kundenfeedback und dass ich die Reports mit den Mechanismen von Scrum, also die Kundenkommunikation konnte ich schon danach gestalten. Also, in dem ich halt die Fragen vom Daily auf den wöchentlichen Report ausgedehnt habe. In dem Sinne, was haben wir in der letzten Woche erreicht, was planen wir nächste Woche zu erreichen und was sind unsere Hindernisse. Dass das eben ein schönes Fragenset war, um es auch in dem Report an den Kunden zu kommunizieren. Das Einfordern von klaren Prioritäten, also wirklich unique Ranking, von den nächsten interessanten Themen, habe ich als sehr hilfreich empfunden. Haben auch die meisten Kunden klaglos mitgespielt und verstanden warum. Ansonsten vor allen Dingen die Kohäsion im Team. Also durch die regelmäßigen Retrospektiven, durch das ständige Verbessern im Team und auch durch das Aufbrechen der klassischen Rollen und das Mehrdenken in kurzfristigen und erreichbaren Zielen, nicht irgendwo in den nächsten 5 Monaten ist der nächste Milestone, sondern dieses gezielte, gemeinsame Arbeiten am nächsten Feature. Das bringt einfach mehr Zufriedenheit.

I: Wie war das Feedback vom Entwicklungsteam an sich bezüglich dieser Scrum Methodik?

B3: Sehr gut gefallen hat ihnen immer der Fokus, den das Team hatte. Das es halt gemeinsam am nächsten Sprint Ziel arbeitet. Was auch heißt, dass ich mich darauf verlassen kann, dass sich der Entwickler darauf verlassen kann, dass das, was er gerade entwickelt, schon morgen oder übermorgen vom Tester schon getestet wird. In der klassischen Entwicklung ist es halt ganz häufig so, dass der Entwickler schon lange vergessen hat, oder das Team verlassen hat, bevor der Tester die Funktion überhaupt testet und sich dann wieder als Entwickler erinnert, warum habe denn das vor einem halben Jahr so gemacht, habe ich mir dabei was gedacht oder habe ich da jetzt wirklich einen Fehler gemacht. Ist sehr viel schwieriger, als wenn ich es erst gestern gemacht habe. Vorteil hier wäre eine schnellere Erfolgsbestätigung.

I: Gab es auch schlechte Erfahrungen mit dieser Methodik?

B3: Ja sicher, wenn jemand aus dem klassischen Bereich kommt, dann reingesetzt wird, ohne Schulung und meistens ist es so, das geschieht ja von heute auf morgen, dass dann die Schulung gar nicht so möglich ist, dann gibt es Reibungsverluste erst mal. Gut die ersten zwei, drei Sprints werden sowieso scheitern mit einem Team, das Scrum nicht kennt. Aber definitiv und ich habe bisher noch nicht erlebt, dass wegen Scrum sich in einem Team Leute an die Gurgel gegangen sind, aber Scrum bringt diese Konflikte wegen der intensiveren Kommunikation eher an die Oberfläche.

I: Muss ja nicht unbedingt ein Nachteil für das Projekt sein.

B3: Nein, aber man kann sie halt nicht mehr unter den Teppich kehren.

I: Gab es agile Projekte, die du im Nachhinein klassisch machen würdest?

B3: Da ich ein Fan von Scrum bin, nein. Ich sehe Scrum halt eher als Methodenbaukasten und es gibt Elemente, die ich in jeden Fall anwenden würde. Wenn ich eben weiß, der Kunde akzeptiert nicht die Vertragsgestaltung im Sinne eines Backlogs und dann Feature Änderungen einfach durch Backlog Definitionen, weil es halt als klassisches IQP-4 oder 5 ausgelegt ist, dann lass ich die Elemente halt weg, mach sozusagen die Gestaltung zum Kunden klassisch und nehme die Elemente wo es um Teamkohäsion geht und gemeinsames Lernen noch mit rein, also ScrumBut. Ich bin ein Fan von Hybrid. Das ist halt das einzige, was ich im realen Leben bei den Projekten, die ich bisher hatte, umsetzen konnte. Wenn der Kunde die Scrum Values verstanden hat, dann könnte man auch Scrum in seiner reinen Form machen.

I: Wie gut eignet sich aus deiner Sicht das IQP-3 Modell für agiles Vorgehen?

B3: Hervorragend, weil bei IQP-3 verkaufen wir ja im Prinzip Time & Material von Personen. Das heißt wir verkaufen keinen Scope. Der Scope und das komplette Management des Scopes kann im Projekt flexibel gehandhabt werden, ohne aufwendige Change Requests. Aus meiner Sicht ist also IQP-3 die ideale Form um die Scrum Welt umzusetzen, weil wir diskutieren nicht über Werthaltigkeit und Wertsteigerung und letztendlich konkrete Bezahlung. Das einzige worüber wir mit dem Kunden reden, ist über die Teamzusammensetzung, ob der Kunde mit einem Mitarbeiter zufrieden oder unzufrieden ist und das halte ich für legitim. Das ist eine Offenheit die ein Team aushalten muss. Insofern ist IQP-3 die ideale Form für Scrum.

I: Wie eignet sich IQP-4?

B3: Managebar. Also da ist es eine Frage der Vertragsgestaltung, ob wir sagen der Kunde bezahlt für ein Team von n-Leuten über die Dauer eines Sprints, ohne dass wir den Scope festzurren, dann ist es ja eigentlich ein paketierte IQP-3. Der Kunde zahlt halt nicht mehr pro Stunde, sondern er bezahlt für ein Scrum Team von 7 Leuten, für den Zeitraum von einer Sprintlänge von 2 Wochen, oder so, als ein Paket, kann das bestellen und man muss dann halt das Sprintziel festzurren und am Ende liefern. Kann man machen. Andere Möglichkeit von IQP-4 ist tatsächlich so etwas von Story Point zu vereinbaren, also Bezahlung per Story Point. Dann kommt man schon wieder in die Diskussionspunkte, wenn sich der Wert eines Story Points im Laufe eines Projektes allmählich schleichend verschiebt. Wer definiert, wieviel ist ein Story Point? Also die Lehre von Scrum sagt an der Stelle zwar, dass legt letztendlich das Team fest, aber dann haben wir wieder die vertraglichen Probleme. Dann ist wieder Geld involviert und dann gehen die Diskussionen los.

I: Wie sieht es mit dem IQP-5 Modell aus?

B3: Da würde ich Scrum nur unter der Haube verwenden. Also tatsächlich in der Kommunikation mit dem Kunden klassisches Projektmanagement machen und halt Elemente wie interne Gestaltung des Teams, Retrospektiven, das interne Schaffen von schnellen Feedbackschleifen zwischen Entwicklern, Testern und anderen Gewerken im Team. Das würde ich halt immer noch umsetzen können, aber nach oben hin würde ich es halt dann klassisch managen müssen. Weil IQP-5 heißt für mich, dass ein Scope festgezurret ist. Bei IQP-5 kann ich bestenfalls noch ein ScrumBut realisieren.

I: Siehst du bei IQP-3 noch Verbesserungspotential, Änderungsbedarf für agiles Vorgehen?

B3: Also normalerweise haben wir ja wenig Einfluss. Wir kriegen ja die Ausschreibungsunterlagen, die dann mehr oder weniger schon ein IQP-3 oder irgendwas Entsprechendes fordern. Wenn der Kunde sagt, wir wollen Scrum, dann wird der auch sein entsprechendes Vorgehensmodell auf uns runter brechen wollen. Dann hat der schon seine Struktur und wir betten uns in diese ein. Insofern haben wir im Normalfall nicht furchtbar viel Spielraum. Wenn wir jetzt einen Kunden hätten, selber sozusagen ein Projektteam aufsetzen möchte und das mit uns gestalten möchte, dann haben wir wieder mehr Spielraum. Ich weiß gar nicht ob wir schon eine geeignete Vorlage dafür haben bei

Alten. Sollte es noch keines geben, könnte man ein Template mit dem ersten Kunden erstellen.

I: Wenn du jetzt Template erwähnst, beziehst du das auf das Angebot?

B3: Ich beziehe es auf das Angebot. Letztendlich ist der Vertrag ja Ausschreibung + Angebot + Bestellung. Das Einzige von den drei Dingen, auf das wir einen Einfluss haben ist das Angebot, weil das von uns kommt. Die Ausschreibung kommt vom Kunden, da haben wir normalerweise keinen Einfluss, die Bestellung kommt vom Kunden, auch das macht der nach seinen Prozessen. Das Einzige was wir formen können, ist das Angebot und dementsprechend wäre das der Teil, den wir dann durch Template oder Beispiele füttern könnten und sagen könnten, wenn der nächste Kunde um die Ecke kommt, der genau sowas will, da hätten wir was.

I: Wie integriert der Kunde das Angebot in den Vertrag?

B3: Ganz einfach. Er macht eine Bestellung und verweist in der auf das Angebot. Also unser Business Management ist sowieso gehalten, wenn eine Bestellung kommt, drauf zu prüfen, ob der Verweis auf das Angebot drinnen ist. Weil sonst wäre alles, was wir da reingeschrieben haben, rechtlich nicht bindend, wenn wir die Bestellung akzeptieren.

I: Du hast ja erwähnt, dass es den ein oder anderen Kunden gibt, der nicht 100 % weiß, was Scrum genau bedeutet. Gibt es hier vielleicht Anpassungsbedarf?

B3: Ja, also die schwierigste Rolle die der Kunde wahrnehmen muss, eigentlich die einzige Rolle, die der Kunde wahrnehmen muss in solchen Projekten, ist ja die des Product Owners und insofern kann es hilfreich sein, entweder in das Angebot unsere Anforderungen an den Product Owner festzuzurren. Was wir schon an Erfahrungen gesammelt haben ist, dass es sinnvoll ist da auch mit reinzuschreiben, dass wir eventuell einen Product Owner Proxy bei uns installieren. Dann halt für zusätzliche Kosten, der den Kunden an der Stelle entlastet. Denn dieser dreifache Spagat zwischen Mach, Verfügbarkeit und Sachkenntnis über das Produkt, den kriegt man in der Schönheit nie. Insofern muss man immer in der einen oder anderen Richtung Abstriche machen und das kann man, je nachdem wo die Abstriche liegen, kann man das eben abfangen, indem man Product Owner Proxy von uns beim Team etabliert der dann die Kommunikation managt.

I: Siehst du beim IQP-4 mögliche Anpassungen oder Änderungen, die eine Verbesserung für agiles Vorgehen bringen könnten?

B3: Eigentlich relativ wenig. Das, was ich eben über die Mitwirkung des Kunden über den Product Owner gesagt habe, gilt hier auch. Dadurch, dass es paketbasiert bezahlt wird, ist es sogar noch wichtiger, dass man dann dafür sozusagen ein Paket vorsieht, Product Owner Proxy, dass optional in Absprache abgerufen werden kann.

I: Siehst du beim IQP-5 mögliche Anpassungen oder Änderungen die eine Verbesserung für agiles Vorgehen bringen könnten?

B3: Ja und nein. Wenn der Kunde Scrum nicht kennt, würde ich nicht darauf bestehen, Scrum zu machen. Aber ich kann es trotzdem versuchen, auch wenn ich es nicht so nenne, die wesentlichen Elemente mit rein zu bringen. Ich kann immer noch fragen, wie häufig möchtest du denn ein Release von uns sehen, wie häufig kannst du die angucken und wie häufig kannst du uns Feedback geben. Ich würde vom Kunden immer noch eine klare Priorisierung der Anforderungen fordern, auch wenn der Scope festgezurr ist. Ich würde auch sein genaues Prozessmodell fordern und ich würde immer noch fragen, welches Qualitätslevel forderst du denn für welche Art von Release und wann willst du die jeweils haben. Das ist eigentlich dann alles und den Change Request Prozess mit ihm am Anfang festzurren. Dann habe ich eigentlich alle Elemente die mir in Scrum besonders wichtig sind auch im klassischen Projektmanagement abgedeckt.

I: Also du würdest diese Punkte mit ins Angebot aufnehmen?

B3: Das ist ja nicht einfach, wir schreiben ein Angebot und der Kunde frisst oder stirbt. Das sind ja normalerweise mehrere Runden in denen wir miteinander diskutieren. Ich würde versuchen möglichst viele Elemente davon mit dem Kunden abzusprechen und wenn er es für sinnvoll hält mit aufzunehmen. Und wenn er es nicht für sinnvoll hält, dann weiß ich, woran ich bin. Dann darf ich halt das Element nicht umsetzen und bin dann wieder bei ScrumBut.

I: Passiert das häufiger, dass der Kunde einen Festpreis a la Scrum fordert?

B3: Einen Festpreis a la Scrum habe ich bisher noch nicht gesehen. Wenn, dann war es ein IQP-4 mit Deckel. Also das da drinnen vereinbart ist, was weiß ich, wir bezahlen pro Teamsprint so und so viel Euro und wir gehen davon aus, dass wir 20 Teamsprints brauchen und genauso viel Budget stellen wir bereit. Also der Kunde hat das Recht vorher

abzubrechen, aber wenn das Geld alle ist, muss er sich halt rechtzeitig um einen Nachfolgeauftrag bemühen. Also das ist eher so ein IQP-4 mit Deckel. Ein echter IQP-5 mit festem Scope und nach Scrum, da hätte ich von Alten her Bauchschmerzen. Ich würde dann meine Bedenken anmelden.

I: Denkst du, wäre es allgemein für die Zukunft sinnvoll, Änderungen oder Anpassungen bei der Prozesslandschaft durchzuführen?

B3: Nein.

I: Auch nicht bei den Verfahrensanweisungen und Arbeitsanweisungen?

B3: Nein, der Grund ist folgender: Alten hat momentan den Schwerpunkt seines Geschäftes bei 40 % auf IQP-1, also Arbeitnehmerüberlassung und zu 60 % auf den Anderen. Davon mit Sicherheit ein Großteil mit IQP-3. Das heißt, wenn der Kunde sagt, macht Scrum, dann machen wir Scrum. Wenn der Kunde sagt, macht nicht Scrum, machen wir nicht Scrum. Das ist IQP-3. Insofern müssen wir dafür sorgen, wenn der Kunde sagt, macht Scrum, dass wir ausreichend Leute haben, die wissen, was das heißt. Wir müssen aber deswegen nicht unserer Prozesse ändern, weil wir ja eh nach den Prozessen des Kunden arbeiten. Alles was wir brauchen, ist mehr Sachverstand im Schreiben des Vertrags. Dadurch ändern sich aber keine Prozesse. Und mehr Sachverstand zu Scrum in der Umsetzung, aber auch da schreiben wir keine Prozesse, weil das machen die nach den Prozessen des Kunden. In IQP-4 und IQP-5, da kann man darüber reden, wenn wir das machen. Also wie gesagt, beim IQP-4 vielleicht. In diesem zweiten IQP-4 plus Modell, wo der Kunde halt tatsächlich nach Story Points bezahlt, da müssen wir womöglich schon ein Prozessmodell definieren, wie wir im Team arbeiten wollen. Und das intern definieren. Solange es außer Scrum keine anderen Prozessforderungen gibt, oder nur die alten ISO9000, müssen wir auch hier nichts tun. Da kann der Projektleiter das einfach definieren, umsetzen und muss den Prozess noch nicht mal dokumentieren. Wenn wir jetzt Scrum in einem IQP-4 Projekt umsetzen wollen, das noch andere regulatorische Anforderungen hat, dann entstehen daraus Prozessdokumentationspflichten und wenn wir in diesem Umfeld mit Scrum arbeiten wollen, dann müssen wir Scrum dokumentieren. Also den Prozess, den wir vorhaben nach Scrum zu leben, sauber hinschreiben, nachweislich danach leben und auch danach abprüfen. Also die Forderung kommt dann nicht durch Scrum, sondern die anderen Prozesse, die wir ebenfalls anziehen müssen.

I: Wie siehst du das bei IQP-5?

B3: Genauso. Schlimmer noch.

I: Du hast ja erwähnt, man könnte so Bausteine für die Angebotsphase erstellen. Also wenn der Kunde Scrum fordert, dass die Business Manger auf Templates zugreifen können, oder?

B3: Ja, das ist eine Änderung, die hat Alten schon hinter sich. Wo wir es halt dann noch nicht haben, ist in der Projektdurchführung, aus den besagten Gründen, weil diese Anforderungen, das zu dokumentieren, kommt nur durch die Hintertür.

I: Wenn du an dein letztes Projekt denkst, hast du dieses nach dem in der Literatur genannten Prozess durchgeführt, oder gab es Abweichungen in Bezug auf die Meetings, Artefakte und Rollen?

B3: Immer. Ich habe noch nie ein Projekt nach der reinen Lehre von Scrum abgewickelt. Inzwischen betrachte ich mich als Experte im ScrumBut. Meetings hatten wir relativ strikt, was wir nicht geschafft haben, war das Timeboxing. Also die Sprintlängen schwankten ein wenig, was an dem Terminkalender des Kunden lag, der die Rolle des Product Owners innehatte. Der Aufbau von Dailys und Rollen schwankte, da war das Problem, die Verfügbarkeit des Scrum Masters. Das Projekt funktionierte erst dann, als wir einen Scrum Master, der Vorort beim Team sitzen konnte, frei schaufeln konnten. Die Meetings hatten wir, wir hatten auch eine Retrospektive. Halte ich für das wichtigste Meeting. Rollen hatten wir. Wir hatten Team, Product Owner beim Kunden und Scrum Master. Mehr fordert Scrum ja nicht. Artefakte. Wir hatten ein Backlog, wir hatten ein Sprint Burn Down, wir hatten ein Product Burn Up. Also von dem was Scrum hier fordert, hatten wir eigentlich fast alles, trotzdem war es aus meiner Sicht keine Scrum, weil eben das Metaprinzip der festen Sprintlänge verletzt war und weil es ein IQP-5 war, in dem wir relativ viel Zeit mit dem Nachverhandeln verbracht haben. Und das ist etwas, das Scrum nicht beschreibt und nicht kennt. Das sagt halt nur, hier ist das Backlog, das ist eindeutig priorisiert und das man dann über Features nachverhandelt oder nicht, kennt zwar Scrum nicht, verbietet Scrum aber auch nicht.

I: Welche von den Abweichungen haben die größten Nachteile mit sich gebracht?

B3: Was das Projekt in der Anfangszeit fast gesprengt hätte, war eben, dass der Scrum Master nicht adäquat besetzt war, auch wenn die Rolle definiert war, es hat halt einfach nicht funktioniert. Und was danach der größte Hemmschuh war, war eigentlich die Inflexibilität des Vertrages, weil der Scope halt festgezurrert war und dann das frühe Feedback, das halt viele Änderungswünsche brachte, was ja in Scrum wünschenswert ist, bei klassischen Verträgen aber Ärger macht, weil man die dann nachverhandelt oder entsprechend umpriorisiert.

I: Gehen wir nochmals auf die letzten Projekte ein. Was ist hier sehr gut gelaufen in Bezug auf die Meetings, Rollen und Artefakte?

B3: Als Gewinn und identitätsstiftend fand ich am wichtigsten das Daily und die Retrospektive. Das Planning Meeting ist notwendig und die Demo ist extrem hilfreich, wenn man einen aktiven Product Owner hat.

I: Gibt es Punkte, die du bei einem neuen Projekt ganz anders machen würdest?

B3: Nein. Es gibt zwar einige Ecken, wo ich im Verlauf eines Projektes in eine Ecke gerannt bin, mit dem das Team nicht glücklich war. Das kam aber meist in der Retro raus und haben wir dann den Verlauf des Projektes selber noch umgebogen. Also insofern, nein. Der große Vorteil der Retro ist, dass ich Teamsorgen einfangen kann und gleich verbessern kann. Das Einzige was ich feststelle, Projekte bei Alten werden immer größer. Das heißt, Skalierung wird ein Thema. Es ist also eher so ein Wachstumsthema. Bis jetzt kam ich immer noch mit dem klassischen Setup von Scrum, einem Scrum Team aus. Demnächst werden Projekte locker 20, 30, 40 Leute haben. Da muss man sich schon über korrektes Skalieren ernste Gedanken machen.

I: Gibt es von deiner Seite Ideen oder Wünsche für Änderungen?

B3: Ich formuliere es positiv. Vor einem Jahr gab es hier maximal eine Handvoll Scrum Master, inzwischen gibt es zertifiziert ca. 30, 40. Also innerhalb von einem Jahr, haben wir hier einen riesen Sprung gemacht und wir sorgen dafür, dass die Scrum Master untereinander in Kontakt bleiben, dass jeder nicht nur für sich lernt, sondern wir auch einen Wissensaustausch untereinander betreiben. Wir haben jetzt allmählich eine kritische Größe, wenn jemand nicht nur die Ausbildung als Scrum Master hat, sondern dann auch tatsächlich in einem Projekt landet, in dem er das anwenden darf, seine Erfahrung auch den anderen Scrum Master mitteilen könnte. In dem Fall wäre mein

Feedback, einfach diese Community weiter hegen und pflegen. Einfach nur, damit dieser Gedanke der lernenden Organisation da über das Storytelling weiter funktioniert.

I: Fällt dir noch etwas in Bezug zum Kunden ein?

B3: Das würde ich sagen, hängt von Fall zu Fall ab. Wenn der Kunde uns schon konkret vorlegt, wie er Scrum leben möchte, dann haben wir da relativ wenig Einfluss darauf, dann können wir uns entweder danach richten oder nicht. Wenn wir einen Kunden gewinnen, der vorher kein Scrum und kein agil kennt und wir den mit den Ansätzen überzeugen können, dann ist es sinnvoll während der Vertragsverhandlung mit ihm seine Rolle zu klären und dann eben im Vertrag ein Fallback aufzunehmen, wenn der Kunde diese Rolle nicht wahrnehmen kann. Weil das eine ist es, ihn zu trainieren, das zweite ist auch, dass er das Kapa (Kapazität) dafür hat. Wenn er es dann auch tatsächlich tun muss.

Interview 4

I: Was sind deiner Meinung nach negative Einflussfaktoren für den Projekterfolg bei klassischer Projektabwicklung?

B4: Negative Einflussfaktoren, wenn ich es rein nach Aufwand mache, habe ich eigentlich relativ wenig Risiko, wenn es länger dauert und der Kunde zahlt, wäre mir eigentlich relativ gleichgültig. Schwierig wird es halt, wenn es nach Festpreis ist. Man kann sich verschätzen, man braucht länger, man hat nicht die richtigen Leute und kommt dadurch letzten Endes, sei es zeitlich oder mit den Kosten nicht dahin, wo der Kunde einen auch sieht. Dadurch muss man halt länger arbeiten, nacharbeiten oder Strafe zahlen, weil man Meilensteine überzieht. Das sind so negative Sachen. Einflussfaktoren sind, habe ich das Staffing richtig, habe ich da wirklich die richtigen Leute, die ich brauche, habe ich die Prozesse, damit ich rechtzeitig gegensteuern kann, also so KPI, Reporting spielt da eine Rolle. Ansonsten, wie ist der Kunde, auch wie kooperativ ist der. Es gibt Kunden, die sind so chaotisch, da kann ich super aufgestellt sein, aber die ziehen dann einen mit ins Chaos und obwohl man selbst alles richtig angeht, kann ein Projekt dann doch noch im Chaos enden. Gerade wenn der Kunde sich dann auch noch selber viel einmischt und einen gar nicht so machen lässt, wie man es selber besser machen will. Änderungen auch. Wenn das sehr dynamisch ist, der Kunde ständig Änderungen will, teilweise auch bewusst und gezielt Sachen einsteuert, die er gerne noch mitmachen will, die gar nicht Teil des Angebots waren und hat dann vielleicht einen schwachen

Projektleiter, der es nicht schafft, das abzuwehren. Am besten bei einem klassischen Projekt ist, ich mach es nach Aufwand, da kann ich relativ viel abfedern. Klar, ich kann immer noch Pech haben, mir werden die Leute krank oder kündigen, das sind auch Sachen, die ich nicht wirklich 100 % selber steuern kann, aber ich kann dafür sorgen, dass die Mitarbeiter motiviert sind und gute Stimmung haben.

I: Welche Faktoren würdest du zu den Hauptgründen für nicht erfolgreich abgeschlossene Projekte zählen?

B4: Als Vertriebler hat man seine Ziele, ich muss unbedingt noch ein Projekt gewinnen und biete dann unter Preis an oder bin zu optimistisch. Oft werden ja Projekte angeboten und hab die Leute noch gar nicht. Ich gehe davon aus, ich brauche 2 Leute und die kosten ungefähr so viel und biete das so an und gewinnen das dann auch, weil der Einkauf mich noch runtergehandelt hat und ich habe das dann gewonnen und am Ende sind die Leute dann doch teurer. Am Ende brauche ich dann vielleicht doch mehr Leute, brauch ich drei statt zwei. Da kann man die besten Projektmanager haben, das kriege ich schlecht in dem Zielpreis realisiert. Viele Probleme sind schon am Anfang da. Ansonsten wenn das aber alles richtig abgeschätzt ist und das kann man mit Erfahrung, ich sag mal ich habe schon 100-mal einen Außenspiegel konstruiert, ich werde auch für den 101sten so viel Leute brauchen. Wenn der Kunde dann noch Last Minute ständig Änderungen wünscht und 1000 Features redefiniert, die gar nicht angeboten waren, haut das mit der ursprünglichen Planung, die am Anfang gut war, nicht mehr hin. Also entweder wird der Fehler gleich am Anfang gemacht, da ist es sehr schwer hinten raus auszugleichen oder während der Durchführung, dass ich die Sachen nicht richtig tracke oder die Meilensteine falsch setze oder eben Änderungswünsche, die vom Projektleiter einfach aufgenommen werden.

I: Siehst du die treibende Kraft für agiles Vorgehen eher bei Alten oder beim Kunden?

B4: Ich denke, dass letzten Endes nicht Alten das unbedingt pusht, sondern einfach der Markt wird immer schnelllebiger, die Entwicklungszyklen werden immer kürzer und dynamischer. Neue Player im Automobilsektor, wie Tesla, Google oder Apple setzen etablierte OEMs unter Druck. Diese müssen sich dann natürlich auch neu erfinden. Das akzeptiert der Kunde nun mal nicht, wenn er bestimmte Features nicht hat, die der neue Konkurrent aber schon hat. Klassisch wandern die Entwicklungszyklen halt viel zu starr und das muss sich halt ein Stück weit ändern. Ich kann nicht ein Lastenheft auf 3 Jahre

schreiben und im Voraus schon alles definieren was kommt. Wenn plötzlich ein Tesla neue Features drinnen hat, die ich vorher noch nicht mal auf meiner Roadmap hatte, dann bringt es einem nichts, wenn ich den Plan starr 3 Jahre lange befolge. Ich muss da halt umpriorisieren. Weil sonst habe ich zwar das Auto das ich vor 3 Jahren definiert habe, aber das will dann vielleicht keiner mehr, weil sich der Markt gedreht hat. Da muss man halt dann sehr schnell reagieren können, Features, die dann vielleicht doch nicht so wichtig sind nach hinten schieben und andere nach vorne ziehen und das geht halt mit agilen Projekten deutlich besser. Also ich denke der Endkunde ist der Treiber. Unsere Kunden, also Tier1 und OEMs sind vom Markt getrieben und vom Wettbewerb der halt stärker wird. Und Alten muss das bedienen. Also ich denke nicht, dass wir unsere Kunden damit treiben.

I: Welche Vorteile siehst du hier für Alten?

B4: Grundsätzlich, Alten kommt ja aus der Arbeitnehmerüberlassung, wo man zeitbasiert die Leute verleiht. Die Alternative dazu sind die Werkverträge, die immer stärker werden. Früher waren es 30 %, mittlerweile ist es umgekehrt. Das ist für uns halt eine Chance auch größere Projekte zu machen, länger laufende Projekte und damit mehr Wachstum für uns zu generieren. Da kann diese agile Vorgehensweise schon ein Ansatz sein, um auch dem Kunden das besser zu verkaufen und gleichzeitig für uns das Risiko im Griff zu behalten. Man könnte ja jetzt auch schon sagen, ok, wir machen einfach riesig große Festpreisprojekte mit 20 oder 50 Mann, aber da hast halt immer das Risiko. Was ist, wenn wir nicht die richtigen Leute haben, was ist, wenn wir länger brauchen als geplant und über diese agilen Ansätze kann man das doch so ein bisschen von IQP-5 nach IQP-3 schieben, um das doch nach Aufwand abzurechnen. So best of both worlds. Einerseits große Gewerke, die beliebig verlängert werden können, wo wir auch das Staffing in der Hand haben, aber trotzdem das Risiko für uns soweit minimieren, dass es trotzdem ein Aufwand ist und der Kunde alles zahlt. Und wenn es halt länger dauert, weil er ständig Änderungen hat, er dann auch letzten Endes auch zahlt und nicht wir.

I: Denkst du gibt es gewisse Grundvoraussetzungen, die es benötigt, damit eine agile Projektabwicklung überhaupt Sinn macht?

B4: Intern kann man agil relativ unkompliziert machen, ich glaube da gibt es wenige Voraussetzungen. Na klar, das Team selber, die Mitarbeiter sollten geschult sein, die sollten wissen, was das überhaupt ist, was das bedeutet, welche Rollen. Ich kann nicht

mit einem Mann ein agiles Projekt machen. Also wenn man von agil redet, habe ich auch verschiedene Rollen. Einen Scrum Master, Product Owner und die Entwickler. So um die 7,8 Leute im Team. Wenn es zu klein ist, dann hat das aus meiner Sicht nicht viel damit zu tun. Wenn es jetzt mit Kundenprojekten ist, Grundvoraussetzung ist da natürlich, dass der Kunde das akzeptiert, dass er es versteht, das ist halt Teil des Vertriebs, dass man den Kunden die Vorgehensweise erklärt. Manche Kunden kennen das schon, die sagen, ja super, habe ich auch schon woanders so gemacht. Dann ist die Grundvoraussetzung auch schon erfüllt, dass der Kunde da mitspielt, wenn er es überhaupt mitkriegt. Also wenn der Kunde sagt, IQP-5, macht es wie ihr wollt, solange die Ergebnisse stimmen, kann man das intern machen wie man will. Wenn man eher IQP-4 oder 3 macht, wo der Kunde auch mehr Transparenz hat und das auch sieht, dann muss er das dann auch akzeptieren.

I: Ich hake ganz kurz ein. Macht das überhaupt Sinn, wenn ich sage, machen wir IQP-5 und der Kunde will aber nichts damit zu tun haben. Wie mache ich das dann mit meinen Reviews?

B4: Die kann ich ja intern machen. Scrum wurde ja nicht für die unterschiedlichen IQP Modelle gemacht. Ich habe ein Lastenheft auf drei Jahre und ich habe einen festen Preis, da kann ich natürlich durch so kleine agile Sachen trotzdem für mich den Fortschritt besser tracken, weil, wenn ich einfach nur den großen Plan aufsetze und hab 3 Meilensteine, der erste nach einem Jahr, der zweite nach dem zweiten Jahr und der letzte nach 3 Jahren, dann habe ich dazwischen natürlich immer 12 Monate, wo ich gar nicht weiß, bin ich jetzt im Plan oder nicht. Und Scrum, oder diese agile Herangehensweise ermöglicht mir natürlich schon über webbasierte Programme wie Jira oder Trello meine Aufgaben zu visualisieren. Manche machen das auch mit Post-it Zetteln auf einem physischen Board. Ich habe eine gute Transparenz im Team, jeder sieht so ein bisschen was macht der Andere, ich habe als Mitarbeiter auch das Gefühl ich schaffe was, ich sehe wirklich was von In Progress nach Done geschoben wird. Das ist schon ein bisschen motivierend. Wenn ich dann kleinere Zeiteinheiten mache und das tracke eben auch, wieviel Aufgaben habe ich pro Sprint, dann habe ich automatisch einen KPI messbar, wie ist mein Fortschritt. Wenn ich sonst 3 große Arbeitspakete habe muss ich sagen, ok, wieviel Prozent sind denn vom ersten Paket fertig. Habe ich 10 % oder 15 %, ich muss den Projektstrukturplan ja irgendwie runterbrechen in Subtasks. Im Prinzip ist diese agile Methode oder Scrum erstmal nur eine Vorgabe, dass man große Arbeitspakete runterbricht in Teilaufgaben, die man dann natürlich zählen und messen muss und tracken

kann. Da gibt es ja die schönen Burn-Down-Charts, wo man dann genau sieht, wieviel Aufgaben pro Zeiteinheit auf Done geschoben wurden.

I: Fallen dir noch andere Grundvoraussetzungen ein?

B4: Der Mitarbeiter muss das kennen, der Kunde, wenn er involviert ist, sollte das dann natürlich auch kennen. Tooling und Lizenzen gehören auch dazu, wenn man es nicht grad ganz klassisch mit Post-it's machen will. Das ist ja bei uns alles gegeben. Welche Voraussetzungen könnte es sonst noch geben?

I: Wie siehst du zum Beispiel die Angebotsgestaltung? Denkst du könnte das vielleicht Einflüsse haben?

B4: Der Einkauf ist oft weniger technisch als der Fachbereich, die das kennen und sogar unterstützen, dann kann natürlich sein, der Einkauf prüft das und kann da überhaupt nix mit anfangen. Oder man hat einen Einkauf, der kennt das schon ganz gut, aber auch da können Rückfragen kommen: ich versteh das nicht, geben sie mir einfach wie viele Stunden sie brauchen. Weil der Einkauf will das immer vergleichbar haben, der will am Ende immer wissen, der Lieferant bietet 1000 Stunden zu dem Stundensatz an und der macht 1000 Stunden zu einem anderen und der ist der Günstigere. Das mit dem Ganzen, ich verkaufe jetzt 10 Sprints oder 300 Story Points, das ist für manche Einkäufer etwas verwirrend, wenn der damit noch nie gearbeitet hat, da hatten wir auch schon mit manchen sehr viele Runden drehen müssen, um das Ding klar zu machen.

I: Denkst du macht es Sinn, agil zu entwickeln, wenn von vornherein klar ist, was zu machen ist?

B4: Wenn man es wirklich nach Lehrbuch macht, mit Dailys usw., hatte ich selbst in meiner Entwicklerzeit auch, dann ist das auf jeden Fall kommunikationsfördernd. Hier kriegen alle mit was sind die Auswirkungen bei anderen. Man lernt daraus. Wenn ich als Tester schon mal weiß, dass die Jungs der Requirements nochmal eine Änderung vornehmen müssen, weil der Kunde Last Minute Änderungen hat, dann muss ich dran denken, dass sich das bei mir auch auswirken wird. Das man nicht alle klassischen Wasserfallmethodik hat, ich mach mal alle Requirements und wenn die dann 1000-mal gereviewed und durch sind, dann fang ich an mit der Implementierung und erst, wenn das fertig ist, starte ich mit Testen. Und dann zieht sich das immer. Wenn man aber alle

involvierten Parteien von Anfang an dabei hat und die regelmäßig diesen Austausch haben, dann hat man am Ende glaube ich weniger Überraschungen.

I: Gibt es deiner Meinung nach Knockout Kriterien, die gegen agiles Vorgehen sprechen?

B4: Grundsätzlich kommt das ja aus der SW Entwicklung, wo ja sehr viel kurze Release Zyklen da sind, wo ich jeden Sprint alle 2 Wochen was entwickeln kann und zeigen kann. Wenn es jetzt Komplettprojekte aus anderen Bereichen sind, wir haben es ja auch schon für HW Entwicklung, bei PCB Boards, auch da konnte man es ein Stück weit implementieren. Auch für Testing Sachen kann man es anwenden. Aber es gibt sicher Projekte, die völlig anders geartet sind, wo man Schwierigkeiten hat, Aufgaben zu definieren. Mir fällt jetzt spontan nichts ein, aber diese ganzen Sachen, Fabrikplanung, die sind vielleicht doch nicht ganz agil. Wenn viele Parteien dazukommen, da kann ich nicht spontan Sachen ins Backlog schieben oder rausziehen, weil da vielleicht zig andere Parteien mit dran sind. Da muss man vielleicht doch eine andere Projektmanagement Methodik gehen. Die ich hatte, gingen alle, weil ich halt sehr viele SW Projekte mach. Auch bei einer Konstruktion sehe ich Schwierigkeiten.

I: Denk an dein letztes Scrum Projekt, was lief hier besonders gut und wo siehst du für dich die Vorteile dieser Methodik?

B4: Also wenn man es so macht, wie ich es am liebsten habe, also eher IQP-3, IQP-4 mäßig, dann kann man halt dem Kunden das so transparent machen, dass letzten Endes schon ein Stundenaufwand dahinter ist. Wir nutzen das halt schon stark für die Abrechnung. Man kann sagen, ok, der Kunde, auch wenn es jetzt ein Werkvertrag ist, wo es offiziell egal ist mit wie vielen Leuten wir das bearbeiten, aber trotz allem weiß er, wir arbeiten mit 2 Leuten und haben 2 Leute kalkuliert. Wenn man dann merkt, ok, wir haben den Fall, dass einer längerfristig ausfällt, krank ist, sagen wir mal für 2 Monate und es waren 12 Monate geplant, dann kann ich relativ leicht ausrechnen, wenn ich weiß, jeder Monat hat so und so viel Stunden, wir haben immer so ein bisschen so einen Faktor oft gemacht. So, Story Point entspricht 4 Stunden Arbeit, kann ich die ganze Zeit ausrechnen, wie viele Story Points gehen mir denn verloren. Das kann ich sehr schön dem Kunden dann transparent machen und dann überlegen was heißt das Ganze. Oder wenn ich mal eine Spitze habe, kann ich einen mehr dazu nehmen. Grade wenn ich eine Änderung habe, muss ich mit dem Kunden immer verhandeln. Was müsste der Kunde mehr zahlen oder was muss ich nachlassen? Und wenn man vorher diese 2 Mann für ein Jahr berechnet hat,

dann habe ich ja eine gewisse Anzahl an Stunden und kann dann genau ausrechnen, wenn jetzt einer ausfällt, wie viele Stunden habe ich weniger und wie viel weniger kann ich machen. Am Ende des Tages kann ich halt mit einer Ressource im Projektzeitraum viele kleine Aufgaben machen oder wenige Große. Und wenn sich das ändert, kann ich mit dem Kunden halt regelmäßig die Ressourcen für den nächsten Sprint planen. Also die Kapa (Kapazität) Planung. Was ist alles im Backlog, sind da viele komplexe Tasks, oder weniger komplexe Tasks und da meine Ressourcen ja beschränkt sind, kann ich genau sagen, ich kann so und so viel Kleine und so und so viel Große nehmen. Wenn eine Ressource ausfällt muss ich fragen, was wollen wir streichen. Man hat eine gewisse Deckelung oder Budget an Story Points, klassisch hätte man das nach Zeit gemacht, aber das ist ja im Werkvertrag für die Abrechnung nicht zulässig, weil es halt schon nach Leistung geht, da kann man sich dann schon mit dem Kunden einigen. Eine Ressource ist ausgefallen, ok, wir können uns halt für den Sprint, nur weniger Sachen aus dem Backlog nehmen, um den nächsten Sprint zu planen. Der Kunde hat eine ganz klare Erwartungshaltung, er will immer so viel wie möglich, aber wenn man ihm das so transparent macht, es können im nächsten Sprint nicht genauso viele Tasks bearbeitet werden wie im Letzten, dann kann man das halt besser abrechnen. Man sieht dann halt auch noch wie viel Budget man noch hat für den Rest des Jahres. Um nochmal auf die Frage zurückzukommen ist halt, dass man mit dem Kunden weniger Diskussionen hat, ob die Leistung nun erbracht oder nicht erbracht wurde, oder wie viel, oder angenommen, man wird nicht fertig, wie viel man dann abziehen müsste am Ende. Man kann es sehr leicht messbar machen. Da gehört ja auch noch diese ganze Komplexitätsabschätzung für einzelne Tasks dazu, wo man auch mit dem Kunden sich ja im Sprint zusammensetzt, ok, es kommt ein neuer Task, wo man das Team mit involviert, jeder macht da so seine Schätzung, ok, ist das jetzt ein komplexer Task mit mehreren Story Points oder weniger. Man hat dann immer den Kunden auch im Boot und kann sich da ein Stück weit darauf verlassen, dass es ein gemeinsames Verständnis dafür gibt. Man kann das auch alles im klassischen Projektmanagement machen aber Scrum gibt einem so ein bisschen das Regelwerk und die Methoden, um das alles systematisch zu machen. Es ist auch eine Vergleichbarkeit zwischen den unterschiedlichen Projekten.

I: Gab es bei deinen letzten Scrum Projekten auch Nachteile dieser Methodik?

B4: Also wir hatten ein IQP-4 Projekt, haben 4 Personen zur Verfügung gestellt und es gab keine Mindestabrufmenge. Am Ende war es halt so, dass der Kunde weniger

abgerufen hat. Also das war so ein Ticketsystem, wo wir eine Hotline bedient haben und wenn keiner anruft oder eine E-Mail schreibt, dann ist halt nix zu tun. Da kann man halt Tickets oder Aufgaben definieren, aber wir können uns die nicht selber generieren und der Kunde konnte uns auch nicht mehr geben. Da haben wir uns halt ein Ei gelegt, wir konnten halt nur unser Staffing anpassen, in dem wir sagen, ok, dann müssen wir unsere geplanten Ressourcen halt runterfahren. Am Ende hat man weniger Umsatz gemacht als geplant. Man will ja auch nicht zu früh Leute abziehen, weil wenn sie dann doch gebraucht werden, ist blöd. Weil man arbeitet die da ein und am Ende, wenn sie einmal raus sind, dann sind sie IC, Inter-Contract, dann stehen sie für andere Projekte zur Verfügung. Für solche Sachen haben wir uns auch schon überlegt, sowas von Anfang an in das Angebot zu schreiben, dass es bei sowas auch eine Mindestabnahme gibt.

I: Wie genau schaut ein Scrum Prozess bei so einem Ticketprozess aus?

B4: Wir haben da einen an der Hotline gehabt, der da Sachen aufnimmt. Also erstmal alles, was reinkommt geht ins Backlog. Wir haben dann gesagt, Quality of Service, wir garantieren dem Kunden, dass jedes Ticket innerhalb der und der Zeit bearbeitet wird oder zumindest den Status, es wir bearbeitet, kriegt. Und dadurch ist schon mal alles ein Backlog. Die haben dann geguckt, dass alles im Team verteilt wurde. In dem Fall hat es jetzt nicht viel Sinn gemacht, Dailys zu machen. Es war eher ein ToDo Stapel.

I: Also ein ScrumBut?

B4: Ja, wir haben selten was nach Lehrbuch. Für uns ist es in vielen Fällen schon ein Mittel zum Zweck. Also wenn ich wirklich ein IQP-5 SW Entwicklungsprojekt über mehrere Jahre, ich habe ein großes Team und wo der Kunde völlig frei ist, da kann ich für mich selber Scrum machen, um genau diese Vorteile die Scrum ja hergibt zu nutzen. Aber für viele Projekte ist es für uns halt ein Mittel zum Zweck gewesen, um eben das auch für die Abrechnung zu nutzen. Der Unterschied Werkvertrag, Arbeitnehmerüberlassung ist genau das, nehme ich Leistung ab oder nehme ich Zeit ab. Leistung ist immer blöd, wenn man die Leistung nicht erbringt und der Kunde darauf pocht. Eigentlich wollen wir eher nach Zeit machen, weil die Mitarbeiter werden nach Zeit bezahlt, jeder hat eine 40 Stunden Woche, das heißt, egal was er in den 40 Stunden schafft, er kriegt sein Gehalt. Und unser Ziel ist natürlich möglichst viel dieser 40 Stunden dem Kunden in Rechnung zu stellen. Das heißt, wenn der Kunde uns mit Arbeit zuschüttet, oder auch nicht und die einfach nur dasitzen und darauf warten, dann wollen

wir natürlich trotzdem irgendwie das abrechnen. Sonst kriegen wir nicht die Marge hin, die wir als Ziel haben. Und das ist oft dahinter. Über diesen Scrum Mechanismus kommt man halt schon ein Stück weit weg, ok, ich gebe meinem Kunden jeden Monat nur einen Stundennachweis, der Mitarbeiter war von dann bis dann da und das ist der Stundensatz, weil das geht nicht im Werkvertrag. Aber man kann das schon geschickt verpacken und sagen, wir unterhalten uns jeden Sprint, alle 2 Wochen über die Leistung. Die Leistung der einzelnen Tasks werden mittels einem abstrakten Komplexitätsfaktor oder Story Points Bewertung halt abgeschätzt und die werden abgerechnet. Also man redet schon über Leistung und der Kunde hat auch die Möglichkeit, wenn er irgendeinen Task partout nicht akzeptieren will, nochmal zurück zu gehen. Also man hat schon das unternehmerische Risiko, was für den Werkvertrag charakteristisch ist, aber trotz allem, wenn doch Änderungen sind und der Kunde mitspielt, kann man nochmal abrechnen. Also wenn es dann doch Eigenverschulden war, man hat nicht hingehört, dann kann es schon sein, dass der Task nicht abgerechnet wird, aber sehr oft treibt das ja der Kunde. Beim klassischen Werkvertrag geht jede Änderung über einen Change Request, vielleicht noch über den Einkauf und nochmal ein Nachbesserungsangebot. So kann man das viel besser auf Working Level heben und mit dem Fachbereich aushandeln, weil man kann Aufgaben hin und her tauschen, ohne jedes Mal ein komplett neues Lastenheft zu machen, das ist glaub ich im Interesse aller Seiten, dass das wirklich schlank geht. Wenn man sich gut mit dem Kunden versteht, was auf Working Level sehr oft der Fall ist, dann findet man oft einen pragmatischen Weg. Das ist so ein bisschen der Vorteil, den wir hier sehen.

I: Noch eine Frage zu dem Change Request. Betrifft das hauptsächlich die IQP-5 Modelle?

B4: Ja, also das ist so dieses klassische, ich habe einen festen Preis und ich habe 5 große Arbeitspakete, einen Einzelpreis und irgendwie ein Arbeitspaket kommt komplett raus. Und es steht aber im Angebot genau, wir machen das Arbeitspaket und das ist so groß und kostet so viel. Und wenn das jetzt rausfällt und was Anderes reinkommt, oder 2 fallen weg und werden durch ein anderes ersetzt, erstmal gilt ja immer der Vertrag und je nachdem wie die Prozess des Kunden sind, kann es schon mal sein, dass man alles neu bewerten muss. Da muss man gucken, ich habe jetzt aber an dem Paket bevor der Kunde das gestrichen hat aber schon Zeit investiert. Und da muss man sagen, wieviel denn und das wieder verrechnen. Das kann beliebig komplex werden und indem man halt diesen anderen Ansatz wählt, indem man diese agilen IQP-3, IQP-4 Projekte macht, wo man

eigentlich ein festes Budget hat irgendwo und der Einkauf auch abgesegnet hat, aber die ganzen einzelnen Arbeitspaketinhalte, also was wird denn konkret gemacht in dem Projekt, das ist halt gar nicht im Vertrag geregelt und dadurch kann der Kunde dich einerseits auch nicht darauf festnageln und andererseits hast du durch die kleineren Zeiteinheiten und detaillierteren Beschreibungen viel mehr Möglichkeiten Anpassungen vorzunehmen.

I: Gab es ein Projekt, das du im Nachhinein nicht agil machen würdest?

B4: Das Ticket Projekt hätte man eventuell anders machen können.

I: Wie gut eignet sich aus deiner Sicht das IQP-3 Modell für agiles Vorgehen a la Scrum?

B4: Eigentlich ist das für mich das Beste. Es ist nach Aufwand. Wenn mehr Ressourcen im Projekt gebraucht werden, mehr Stunden, wir haben immer so einen festen Faktor wie ich schon sagte, 4 Stunden entspricht einem Story Point. Dann kann ich mit einem Mitarbeiter am Tag 2 Story Points machen. Wenn er einen zweiten Mitarbeiter noch dazu oder dritten, dann kann ich genau ausrechnen so und so viel Story Points mach ich und rechne nach Aufwand ab. Wenn ich da eine bestimmte Sache entwickle und das ist nichts, dann mach ich im nächsten Monat weiter, hab mehr Zeit, krieg mehr Story Points, ich rechne die ab und reich das komplett weiter. Das ist eigentlich am besten. Das andere ist für mich das Ticketbasierte, wo eben das Risiko besteht, dass die nicht abgerufen werden. Ist halt dann schon eher kritisch zu betrachten. Da muss man dann ein gutes Verhältnis zum Kunden haben. Das heißt, man muss sich relativ sicher sein, dass die dann auch kommen oder eben Klauseln machen, dass bei Unterschreitung einer Grenze doch bezahlt wird. Und das andere ist IQP-5. Da ist es letzten Endes dem Kunden auch egal ob wir Scrum machen oder nicht. Er hat seinen Festpreis und wie wir es dann intern bearbeiten, habe ich halt dann nur für mich die Vorteile, das ich halt besser kommuniziere und da besser merke, wenn ich nicht mehr On Track bin. Aber letzten Endes nutzt es nichts, wenn ich sag, ich habe einen Festpreis, 1 Million, 12 Monate, ich krieg mit meinen Ressourcen das einfach nicht entwickelt, was der Kunde haben will. Er verweigert mir am Ende die Abnahme, es ist völlig egal, wie viele Stunden oder Story Points ich gemacht habe. Wenn ich merke nach der Hälfte der Zeit, ich muss alles über den Haufen werfen und neu entwickeln, bei IQP-5 kann ich dem Kunden das nicht in Rechnung stellen. Er sagt immer noch, du hast das ganze Ding bei dem Preis angeboten, egal ob du nochmal 50 Mann raufsetzten musst die das Last Minute alles nochmal neu machen, das ist ihm dann egal.

Da habe ich jetzt nicht die Möglichkeit, egal ob ich Scrum mache oder nicht, dem Kunden das weiter zu verrechnen.

I: Aber auch bei IQP-3 kann der Kunde die Abrechnung verweigern, wenn er mit dem Ergebnis des Sprints nicht zufrieden ist, oder?

B4: Ja klar, er kann theoretisch dann sagen, ok, das was für den Sprint geplant war, wurde nicht geliefert. Jetzt muss man gucken, woran lag das denn. Habe ich vielleicht doch die falschen Leute? Auch im zweiten Monat, wenn die das nicht hinkriegen, muss man überlegen, muss ich die Leute austauschen, damit ich am Ende das doch noch liefern kann. Leute können ja auch mal einen schlechten Tag gehabt haben, kann ich das vielleicht im nächsten Monat mit Überstunden aufholen. Wenn der Kunde sagt, am Ende interessiert mich nur, was nach 12 Monaten, also am Projektende da ist, dann kann ich das ja ausgleichen oder aufholen oder hol mir noch einen Studenten mit rein. Wenn ich aber einfach ein Problem mit meinen Mitarbeitern hab, dass die das fachlich nicht können oder wie auch immer, dann kann ich dem Kunden nur sagen, so sieht es aus, wir versuchen die Leute auszutauschen und Ersatz zu finden und dann hast du eine neue Mannschaft. Mit der kannst du dann versuchen das aufzuholen. Letzten Endes, egal wie du es drehst und wendest, du wirst dann schon draufzahlen. Wenn der Kunde genau sieht, dass die Leute es nicht hinkriegen, dann tu ich mir schwer ihm den Aufwand zu verrechnen. Im Gegensatz zum AÜ, wo der Kunde ganz klar sich die Leute aussucht, macht er es ja bei den Werkverträgen nicht. Das heißt, wir sind selber in der Pflicht, richtig qualifizierte Leute rein zu setzen. Wenn ich nur Leute nehme, weil sie gerade verfügbar sind, dann ist das keine gute Basis für eine Zusammenarbeit. Das funktioniert auf lange Sicht einfach nicht. Wenn der Kunde selber durch bestimmte Vorgehensweise die Verzögerung verursacht hat, dann muss man sich auch wieder unterhalten. Wenn es aber ganz klar unser Verschulden ist, dann werden wir da draufzahlen.

I: Wenn der Kunde ein agiles Festpreisprojekt fordert, wie siehst du da die Eignung eines IQP-5 Modells?

B4: Ist ok. Die Frage ist halt, bei IQP-5 und Festpreis, theoretisch müsste es dem Kunden egal sein, wie du es bearbeitest. Am Ende hat er einen Preis, ein Budget, was er zahlt, wenn bestimmte Abnahmekriterien erfüllt sind und eigentlich müsste es ihm egal sein. Aber es gibt schon Kunden, die aus welchen Gründen auch immer selber auch agil

arbeiten und sich wünschen, dass sie nach jedem Sprint Reports kriegen, dann kann man es so danach machen.

I: Wenn Ihr allgemein Scrum Projekte durchführt, wer stellt dann den Product Owner?

B4: In der Regel ist der Kunde der Product Owner, weil es ist sein Produkt, sein System. Bei uns ist in der Regel der Scrum Master, der das Team steuert, schaut das sie arbeitsfähig sind. Das ist bei uns dann auch der Projektleiter der sich mit dem Kunden unterhält.

I: Hat es schon mal ein Festpreisprojekt gegeben, bei dem der Product Owner vom Kunden gestellt wurde oder ist das sowieso immer der Kunde?

B4: Nein muss nicht, also wie gesagt, es kann sein, dass der Kunde selbst nicht agil ist und gar keinen Product Owner bereitstellt. Der kann ja einfach nur seinen Repräsentanten, seinen Projektleiter auf seiner Seite stellen, der uns die Requirements rüber wirft und sagt, wie ihr das macht mit Scrum, habe ich keine Ahnung, macht das mal so, wie ihr denkt. Da musst du dann natürlich den Product Owner, der das repräsentiert was der Kunde eigentlich will, stellen.

I: Wenn man aber dann keine Reviews mit dem Kunden macht, dann weiß man ja wieder nicht ob es in die richtige Richtung geht oder was genau der Kunde fordert?

B4: Wenn der Kunde keinen Product Owner nach Scrum bereitstellt, sondern nur einen Projektleiter auf seiner Seite, auch mit dem wird man sich regelmäßig über den Projektfortschritt unterhalten und dem zeigen, was wir gemacht haben. Es muss ja kein klassischer Product Owner nach Scrum Definition sein. Das kann man ja Proxy (Stellvertreter) nennen, das kann ja auch auf unserer Seite der Projektleiter sein, einer der sich halt regelmäßig mit dem Kunden unterhält und versteht, was der will und intern das im Team repräsentiert. Bei enger Zusammenarbeit stellt normalerweise schon der Kunden den Product Owner und definiert uns das gleich ins Backlog rein, was zu tun ist. Aber muss nicht, wenn der Kunde selbst nicht agil ist und keine Jira kennt, dann schickt der uns halt eine E-Mail und unser Product Owner macht das, was in der E-Mail steht in Jira ins Backlog. Der übersetzt das quasi ein bisschen.

I: Würde es deiner Meinung nach Sinn machen, Änderungen oder Anpassungen für das IQP-3 Modell vorzunehmen, um es noch besser für agiles Vorgehen zu machen?

B4: Ich glaube, wir sollten auf einer Seite gucken, dass wir die Angebotsvorlagen für agile Projekte standardisieren, ich glaube da gibt es auch schon Ansätze, wo sich Leute Gedanken machen. Das man da halt einfach ein einheitliches Vorgehen hat und eine Vergleichbarkeit der Projekte. Das sehe ich auch bei dem webbasierten Tool Jira. Hier müssen wir auch noch ein bisschen versuchen das zu standardisieren, um den Projektstart zu beschleunigen. Der Kunde bekommt ja am Ende des Monats immer einen Leistungsnachweis, wo er abzeichnen kann, ja passt alles super, könnt ihr in Rechnung stellen. In der Regel sind das immer Wordvorlagen in unserem internen Alten Vorlagen Pool. Da wollten wir mal ein Tool machen, das diese Dokumente über Jira automatisch generiert. Das heißt ich drück einen Knopf jeden Monat und er zieht mir alles raus, was in der Done Spalte ist bei Jira. Also alles, was der Kunde schon digital akzeptiert hat, das wird dann exportiert, ein automatisches Dokument generiert und da sind dann alle Tasks mit ID, Name und Aufwand gelistet und eine Summe darunter und wo der Kunde dann nur unterschreiben muss. Weil das ist aktuell, was bei uns ein Projektleiter halt monatlich macht und das ist fehleranfällig, wenn man beim Filtern oder Exportieren was vergisst. Da wurde mal eine Arbeit darübergemacht, das Tool aber nicht ganz fertig wurde. Das wäre auch was, was man mal weiter entwickeln könnte, um den ganzen Prozess ein bisschen zu automatisieren.

I: Wo siehst du mögliche Anpassungen bei IQP-4?

B4: Hier könnte man versuchen, auch wenn ein IQP-4 geplant wäre, so zu schreiben, dass es doch ein IQP-3 ist. Also, dass man doch für Mehraufwände trotzdem auch bezahlt wird. Wenn man jetzt ein neues Thema hat und man hat sich beim Ticket doch um das Doppelte verschätzt, dann gibt es natürlich Kunden die sagen, war doch so abgeschätzt. Wenn ihr länger braucht, kann der Kunde auch sagen, der Mann den ihr da dran habt, ist zu langsam. Hier konnte man auch ins Angebot reinschreiben, wir machen so eine Probephase am Anfang, wo wir erst nach dem ersten Monat messen, auswerten, wie viel Aufwand ist es tatsächlich und das dann als Basis für die weitere Abrechnung verwenden. Um letztendlich auch da das Risiko zu vermeiden. Oder wir machen ein Vorprojekt mit dem Kunden, um zu schauen wie lange brauchen wir für ein einfaches, mittleres und komplexes Ticket. Und wenn wir die Zeiten haben, dann skalieren wir das hoch und gehen davon aus, dass das so bleibt. Letzten Endes geht es ja darum, unser Risiko zu minimieren und keine Überraschungen auf uns zukommen und wir am Ende draufzahlen.

I: Wenn du an dein letztes Projekt denkst, hast du dieses laut dem in der Literatur genannten Scrum Prozess durchgeführt oder gab es hier Abweichungen in Bezug auf die Meetings, Rollen oder Artefakte?

B4: Das Daily Scrum wird oft gar nicht gemacht. Oft sitzen die zusammen und tauschen sich aus, aber nicht nach Lehrbuch. Auch diese Schätzung für jede Story wird nicht immer gemacht. Die meisten haben das Lehrbuch halt nie gelesen und machen das eher sehr pragmatisch. Oft wird auch eine Retrospektive und Sprint Planung in einem gemacht. Es wird alles sehr simple gehalten. Wenn das schon sehr gut läuft, dann steuert der Kunde auch die Sachen ins Backlog ein. Man muss auch gar nicht jeden Monat ein Meeting machen unbedingt, wenn es gut läuft. Für manche Kunden ist das auch nur ein Mittel zum Zweck. Manchmal wollen die auch nur wissen, ja ich habe die Leute, die arbeiten durch und mach das nach dem Prozess damit ich Werkvertrag konform bin. Wenn es gut läuft natürlich, wenn es Probleme gibt, dann gucken sie genauer hin und gucken auch auf die Aufwandsabschätzung.

I: Wenn man in Maßen vom Prozess abweicht, denkst du, dass es keine Vor- oder Nachteile mit sich bringt.

B4: Ich glaub eher, wenn man zu sehr nach Lehrbuch geht, wäre das eher hinderlich für unsere Projekte. Wir sind da sehr pragmatisch. Wenn ich ein eigenes Produkt entwickle, dann kann ich natürlich Scrum so leben, wie es für ein SW Team gedacht ist. Aber was Scrum halt mit sich bringt, ist, dass man sich öfter mit dem Kunden austauscht und sehr früh merkt, wenn man gegensteuern muss. Das schlimmste ist, wenn man in eine komplett falsche Richtung entwickelt. Und Scrum zwingt einen halt das regelmäßig anzuschauen und auch ein Feedback vom Kunden zu bekommen, ist das wirklich, was er haben will. Ja sonst kann er ja die Abnahme nach dem ersten Sprint schon verweigern. Wenn man das wirklich klassisch macht, dann ist es stärker in der Verantwortung des Projektleiters. Das Ziel ist ja das gleiche, man will schon merken, ob es in die richtige Richtung geht und man nennt es halt dann nicht so. Man muss es ja nicht Scrum Master oder Product Owner oder Sprint. Einfach nur die Idee dahinter ist gut.

I: Gibt es noch Wünsche oder Ideen von deiner Seite?

B4: Ich glaube halt, was wir sehr viel machen ist, wir schicken sehr viele Consultants auf Schulungen, die wissen am Ende dann halt alles und am Ende machen wir es aber dann nicht so nach Lehrbuch und sind dann vielleicht enttäuscht. Moment mal, das habe ich ja

anders gelernt, warum macht ihr das jetzt so, das ergibt doch gar keinen Sinn. Also eigentlich müsste man eine Schulung definieren, wie wir Scrum machen, damit die dann auch das richtige Verständnis haben, weil ich glaube das Eine hat nicht immer 100 % mit dem Anderen zu tun. Das ist an Scrum angelehnt für unsere agilen Projekte, aber das ist nicht Scrum nach Lehrbuch, wie sich die Erfinder von Scrum das vorgestellt haben, weil ich denke das ist nicht immer 100 % übertragbar. Auf der anderen Seite ist es da aus meiner Sicht viel wichtiger, dass die Business Manager, die bei uns ja den Vertrieb machen und solche Angebote mitschreiben und dem Kunden das auch verkaufen müssen, die müssen eigentlich eher verstehen, warum und wieso und wie das funktioniert, damit die das auch verkaufen können. Aktuell ist es oft so, ein Business Manager geht hin zum Kunden, ja wir machen agile Projekte, ganz toll und so und ich habe mal den Projekt Direktor mitgebracht, der erklärt ihnen das mal. Und ich kann das auch, aber wir haben nicht so viele Projekt Direktoren wie Business Manager, das würde da viel mehr Sinn machen, wenn die mehr geschult werden. Das kriegen die natürlich nicht in einer Standard Scrum Schulung. Aber so wie wir das machen, wie das funktioniert, Best Practices, das kann man sicherlich stärker intern schulen. Dass die das auch dem Kunden so verkaufen und wir das hinterher so auch abarbeiten können, weil dieser agile Ansatz hat auch bei nicht SW Projekten Vorteile für uns, wie Risikominimierung zum Beispiel. Das muss man halt dem Kunden verkaufen, dass es nicht nur Vorteile für uns bringt, sondern auch für ihn. Das bringt bestimmt mehr, als alle Consultants zu Scrum Master auszubilden. Das würde ich mir wünschen, wenn wir mehr in diese Richtung steuern.

Interview 5

I: Was sind deiner Meinung nach negative Einflussfaktoren für den Projekterfolg bei klassischer Projektabwicklung?

B5: Staffing, Kommunikation, Zeit. Bei Staffing ist es oft so, dass wir Projekte starten, wo wir die vorhandenen Kapazitäten bzw. alle Leute noch nicht aufgerampft haben. Hinzu kommt jetzt bei klassischen Projekten, dass wir immer öfter die Qualifikation der Mitarbeiter nicht sofort herstellen können aufgrund des ganzen Fachkräftemangels. Dann die Kommunikation zum Kunden ist oft ganz ausschlaggebend dafür, dass das Verständnis vom Projekt, Input, Output und das, was letztendlich das Team liefern soll, oft differieren. Oft redet man am Thema vorbei. Und drittens Zeit. Oft sind es Projekte, die sehr zeitkritisch sind. Man wird unterbeauftragt, weil man halt Dienstleister ist und

Timelines da sind, die aufgrund mehrerer Faktoren uns einfach in Verzug bringen können und wir dann halt Probleme kriegen. Das sind so aus meiner Sicht die Hauptprobleme. Plus, wenn man jetzt eben den finanziellen Punkt auch noch betrachtet, ist gerade bei den OEMs in den klassischen Themenfeldern, so wie wir arbeiten im klassischen Projektmanagement auch der Preisdruck. Das ist aus meiner Sicht so das, was mich am meisten beschäftigt.

I: Fällt dir in Bezug auf Anforderungen noch was ein?

B5: Früher wurden die Lastenhefte sehr allgemein gehalten, jetzt muss das etwas detaillierter werden, weil wir in die Werkvertragsgeschichten gehen. Aufgrund der ganzen Gesetzesnormen, die da jetzt greifen. Das heißt, die Lastenhefte werden ein bisschen detaillierter, aber oft ist es so, dass die halt nicht ausreichen um abzuschätzen, was hinten raus eigentlich an Leistungen gefordert wird. Wir kommen ja aus dem Time & Material Umfeld, also letztendlich aus dem Body Leasing und jetzt ist es so, dass der Kunde letztendlich mit uns plötzlich Gewerke vor der Tür hat, die dann bei uns abgewickelt werden und da tut er sich schwer die Themen auf den Punkt zu bringen, was erwaarte ich, was muss der Auftragnehmer bringen, was soll die Leistung sein, wie bewerte ich die Leistung. Das ist auch ein ganz großes Thema. Was habe ich jetzt an Leistung gebracht. 160 Stunden sind nicht gleich 160 Stunden. Und da gibt es ganz oft Reibereien. Also Erwartungshaltung und letztendlich der Anspruch und das was rauskommen soll, ist ganz oft ein Thema, wo man sich darüber streitet.

I: Welche dieser Faktoren sind Hauptgründe für nicht erfolgreich abgeschlossene Projekte?

B5: Kommunikation ist aus meiner Sicht das A und O. Kommunikation zwischen Kunde und der Firma, wenn die nicht passt, wenn der Kunde nicht kommuniziert oder selber nicht bewusst ist, was er letztendlich an Output haben will, oder wenn das Lastenheft bewusst so gehalten ist, dass man sich bewusst ein paar Hintertüren offen lässt, ist es dann meistens zum Projektende oder wenn halt wirklich mal was schiefläuft, oder die Timeline nicht passt, das sind die Gründe warum Projekte kritisch werden.

I: Siehst du die treibende Kraft der agilen Entwicklung eher beim Kunden oder bei Alten?

B5: Ich sehe die treibende Kraft auf der Homepage im Web. Ich komme aus der Informatik und habe damals das Wasserfallmodell kennengelernt. Jetzt kenn ich agil seit

Jahren aus der Informatik. Das ist mit allen Dingen gerade, es ist on vogue mit Fachbegriffen um sich zu schmeißen ohne Verstand zu haben, was eine agile Entwicklung überhaupt bedeutet. Ich glaube ganz einfach, dass der Kunde, grade im Automotive Umfeld, sehr viel agil ins Lastenheft schreibt, aber sehr festpreis- und wasserfallmodellmäßig im Projekt unterwegs sein möchte. Das heißt, die treibende Kraft ist meiner Sicht die Auswirkung und auch letztendlich die Geschwindigkeit die man steigern muss, um IT Projekte ans Ziel zu bringen. Weil wir werden immer schneller, wir müssen immer schneller werden. Es ist nicht so wie früher, dass man alle 7 Jahre ein Auto gebaut hat. Man muss permanent Software entwickeln. Das heißt, die sind schon irgendwo gezwungen anders zu arbeiten, aber letztendlich ist es so, dass ich die treibende Kraft nicht dem Kunden zusprechen würde. Also ich glaube, es ist auf beiden Seiten. Man beschäftigt sich ja damit, weil man einfach nicht mehr drum rumkommt. Natürlich ist es bei uns so, dass wenn wir jetzt Projekte oder Lastenhefte haben, wo agile Entwicklung drinnen ist, wir uns primär auch damit beschäftigen. Wir gehen auf Anforderungen vom Kunden ein. Wenn er heute Scrum sagt und morgen Less, Safe, oder was auch immer, dann müssen wir uns da anpassen. Wir sind schon bei Alten so unterwegs, dass wir die Kundenbedürfnisse decken und uns an den Anforderungen orientieren und weniger selber Sachen entwickeln, weil wir nicht in der Lage sind, unsere Prozesse dem Kunden aufzuzwingen. In unseren Inhouse Projekten können wir machen wie wir wollen, aber wenn man für einen OEM arbeitet, muss man sich schon an seinen Prozessen orientieren.

I: Wo denkst du, sieht der Kunde und Alten Vorteile bei der agilen Entwicklung?

B5: Also der Kunde sieht natürlich Vorteile darin, dass er mit einer agilen Entwicklung, wie es nach Scrum ist, das Gewerkcharakter-Thema voll- und ganzumfänglich gesetzteskonform umsetzen kann. Er hat einen Ansprechpartner, Single Point of Contact, die Mitarbeiter sitzen bei uns. Das heißt, er steuert die nicht und sie entwickeln. Er muss sich einfach zusammensetzen mit dem Product Owner von mir aus oder dem Scrum Master und definiert die nächsten Schritte und muss sich um den Rest nicht kümmern. Das heißt, er kann die Verantwortung für eine Entwicklung rausgeben und hat somit mehr Zeit, sich um andere Dinge zu kümmern. Der Vorteil für Alten ist generell im Werkvertrag, dass wir unsere Kapazitäten selber steuern. Dass wir unsere Mitarbeiter so einsetzen, wie es eigentlich passt. Dass wir unsere Mitarbeiter auch intern durch solche agilen Thematiken, oder agile Teams bei uns Inhouse auch besser ans Unternehmen binden, da sie ja bei uns sitzen, weil wir ja natürlich die dann cross-interdisziplinär in

vielen Dingen nebenbei laufen und ausbilden können. Also die Vorteile für Alten sind ganz klar, Knowhow Aufbau, Transfer, Mitarbeiterbindung, Agilität im Einsatz der Kräfte und dann letztendlich auch die Wirtschaftlichkeit. Ein Projekt was wir hier abwickeln ist definitiv, wenn es richtig gesteuert wird, wirtschaftlicher, als wenn 5 Leute vor Ort sitzen und der Kunde jede einzelne Stunde zusammenzählen kann.

I: Denkst du, dass mit agilem Vorgehen alle negativen Einflussfaktoren, die du anfangs erwähnt hast, kompensiert werden können?

B5: Ganz im Gegenteil. Das, was ich gerade erlebe, ist, dass es im Augenblick viel mehr Gesprächsbedarf gibt, als in klassischen Projekten. Die agile Entwicklung haben sich ein paar Leute in Kalifornien ausgedacht. Es gibt ein Manifest dazu und das haben auch alle irgendwo gelesen, denke ich. Problem ist, Agilität funktioniert aus meiner Sicht nur, wenn man auch das Mindset innerhalb der Firma Alten und auch beim Kunden hat. Da muss jeder genau wissen, was er eigentlich darf, was er nicht darf, was er kann und was er tun oder lassen sollte. Und im Augenblick bin ich eher so unterwegs, dass ich sage, wir sind in einem agilen Festpreis unterwegs. Das heißt, man schreibt eine Agilität aus und hat ein Scrum Team, aber schon da fängt das ganze schon oft an, dass der Kunden einfach die Rollen nicht einhält, dass wir intern bei uns letztendlich noch nicht so weit sind, dass wir die Entwicklungsgeschwindigkeit oder das, was ein agiles Team eigentlich ausmacht, dass sie sich selber steuern und regeln und ihre Tasks bewerten und abarbeiten, da sind wir noch nicht so weit. Also ich jedenfalls in meinem Thema noch nicht und da fängt es ja schon an, weil in dem Augenblick wo ich dem Team sage du musst, das, das, das und das bis zum Wochenende schaffen, weil der Kunde will das so, bin ich nicht mehr agil unterwegs. Also, da ist unheimlich viel Gesprächsbedarf und natürlich auch das Verständnis, was bei der Sprintabnahme geliefert wird oder was passiert mit der Velocity oder den Themen die nicht geliefert wurden, das die eigentlich wieder in das Backlog hinten rein und neu bewertet werden müssen, dass man zum Beispiel eine gewisse Timeline nicht eingehalten hat, wobei auch da das Verständnis auf beiden Seiten klar sein muss, dass das bei einer agilen Entwicklung zu jedem Sprint eine lauffähige Version laufen muss. Auch das ist auf beiden Seiten noch nicht so ganz angekommen. Und dann natürlich, dass der Product Owner begreifen muss, dass wenn er einen Sprint anschiebt, dass er nicht von links und rechts einschieben kann. Also all das sind so Erfahrungswerte des letzten Jahres, wo ich sage, das hat uns unheimlich viel Zeit, Eskalation, Nerven gekostet, weil einfach das Mindset, wie ich es immer wieder sagen werde im Interview,

nicht da ist. Gerade bei so einem Automotiv, agil ist cool, agil ist hype, sie versuchen das zu steuern, aber spätestens, wenn sie das nicht machen, kommt der Einkauf und fragt, warum macht ihr Bug Fixing, wir haben doch eine lauffähige Software bestellt. Und da fängt schon wieder die Diskussion an. Agil löst im Augenblick überhaupt keine Probleme. Wir müssen umdenken, wir müssen uns anders organisieren. Aber es wird nur funktionieren, wenn auf beiden Seiten das Mindset gleich ist. Und das sehe ich gerade beim Automotive Bereich noch nicht.

I: Denkst du, wenn das Mindset vorhanden wäre, die Selbstorganisation funktionieren würde, der Kunde das agile Vorgehen versteht, der Product Owner, der Scrum Master, das machen, was sie machen sollten, die angesprochenen Probleme kompensieren könnte?

B5: Ich denke ja, weil es einfach sehr strukturiert ist und aus meiner Sicht ganz klar ist, was der Output eines Sprints ist, bzw. was man da machen möchte. Das heißt, viele Dinge sind einfach definiert und würden das eine oder andere weg kompensieren. Was wir aber nicht vergessen dürfen ist, wir sind sehr zeit- und Timeline getrieben. Das heißt, auch da wir uns agil nichts von der Platte putzen. Wenn wir SOP 2019 haben, muss das Ding 2019 fertig sein. Das heißt, auch da wird es diese Themen immer wieder geben. Zum Thema Geld, ist es einfacher, wenn ich im Angebot einen Auftrag haben, wo ich sage, im Sprint sind so viele Tasks fertig und es kostet so uns so viel. Also das heißt, ich habe die Diskussion nicht, mit wie vielen mach ich das, am Ende zählt die Sprintabnahme und es gibt keine Diskussion mehr. Die Kommunikation ist aus meiner Sicht eindeutiger.

I: Was sind aus deiner Sicht Grundvoraussetzungen für agiles Vorgehen?

B5: Die Grundvoraussetzungen sind aus meiner persönlichen Erfahrung, und ich bin jetzt 16 Jahre in diesem Automotive Umfeld sehr stark unterwegs, das Mindset. Also die Grundvoraussetzung ist, dass man sich eigentlich bevor man ein Projekt anfährt, an einen Tisch setzt und ganz klar definiert, was verstehen wir unter agiler Entwicklung und wie wollen wir die agile Entwicklung in diesem Projekt umsetzen. Das macht man ganz selten. Man schreibt ins Lastenheft, vorausgesetzt wird ein agiler Ansatz, bla, bla, bla und dann geht das Projekt los und spätestens nach 2 Wochen merkt man, dass man noch immer fremdgesteuert ist, dass der gute Kunde im Sprint noch 15 Tasks reinwirft, rausnimmt und nicht versteht, dass es ein Backlog gibt, der wieder rauf und runtergeht, der davon ausgeht, dass wenn er noch was dazu schmeißt, dass es im Sprint erledigt wird.

Das ist für mich wirklich rudimentär. Und wenn das nicht passt, braucht man sich mit Agilität keine großen Hoffnungen machen. Und das zweite Thema, das sehe ich auch bei uns auf Alten Seite, ist das Mindset, dass man letztendlich, wenn man agile Teams hat, diese so aufbauen und heranzuführen muss, dass sie selber agil arbeiten können. Im Augenblick sitzen wir mit unserem Projektleiter drinnen und der steuert die Leute. Der macht ein Standup und letztendlich gibt er vor, was zu tun ist, das sollte eigentlich anders herum sein. Im Augenblick machen wir es genau verkehrt.

I: Denkst du, ist das bei allen Standorten so?

B5: Ich glaube nicht. Ich denke es gibt Themen bei anderen, wo das schon besser läuft. Ich glaube, wir müssen uns selbst noch finden. Wir schulen, wir machen es glaube ich genau richtig, wir schaffen jetzt von oben, von der Hierarchie, erst einmal ein gemeinsames Verständnis. Scrum hat jeder gelesen oder gehört, aber keiner wusste so richtig was dahinter ist. Und das wird glaube ich auch noch ein bisschen dauern. Und dann muss man natürlich innerhalb der Firma dann auch noch loslassen können. Das heißt, wirklich sagen ok, ich bin der Scrum Master, ich steh bereit, ich guck mir das an, aber ihr macht jetzt. Und das wird tatsächlich die große Herausforderung, weil wie gesagt, aufgrund der Teamzusammensetzung, der Dynamik in diesen Teams, wie schnelle sie zusammengesetzt werden und wieder auseinandergerissen werden, sehe ich da die große Herausforderung.

I: Was sind deiner Meinung nach Knockout Kriterien für agiles Vorgehen.

B5: Ganz klar Mindset.

I: Wenn du an dein letztes Scrum Projekt denkst, was lief hier besonders gut und wo sieht du persönlich die Vorteile dieser Methodik?

B5: Ich muss ehrlich sagen, ich bin seit einem Jahr bei Alten und habe ein Projekt übernommen, dass nach Scrum gemacht wird und ich muss sagen, bis jetzt habe ich da nichts, wo ich sage, oh das war super. Man muss dazu sagen, es setzt auf einer vorhandenen Software auf, der Kunde hat seine Rolle als Product Owner überhaupt nicht verstanden, der Einkauf darüber hat und einen Chef und noch einen Chef, so wie das bei den Automobilisten ganz oft ist und dieses Ding hatte ganz schlechte Voraussetzungen. Das einzig Positive ist tatsächlich, dass das Team sich mit der Zeit gefunden hat und diese Dynamik, diese Aufgabenzuordnung und Verteilung angelaufen ist. Wir haben aber noch

heute Nachwehen von diesem Ding. Kann ich leider noch nicht sagen, dass ich besonders glücklich bin.

I: Wo traten da Nachteile auf?

B5: Die Nachteile ganz klar, wir haben letztendlich entwickelt, so wie es nach Scrum sein soll, wir haben unsere Themen priorisiert und festgestellt, dass der Kunde eine ganz andere Sicht hatte und spätestens als der Einkauf dazu kam und das benannte Bug Fixing, was dann irgendwann stattfindet, was auch besprochen wurde, dann plötzlich bemängelt wurde. Ihr macht ja Bug Fixing, dafür bezahlen wir euch nicht. Das Verständnis fehlt hier wieder. Hätten wir eine Software und wir sagen, ihr liefert uns ein Lastenheft, wir leiten ein Pflichtenheft mit 15 Funktionen ab, wäre es einfacher gewesen. Dann hätten wir das dem Einkauf auf den Tisch gelegt. So sind wir immer in einer permanenten Dynamik, wo wir erklären müssen, warum der Fehler da auftritt, weil den haben wir ja da programmiert und das ist mühsam. Das Verständnis ist einfach auf der anderen Seite nicht da, dass man da agil entwickeln kann. Bei dem Projekt hätte ich definitiv ein Lasten- Pflichtenheft gemacht. Weil der Kunde einfach nicht agil denkt. Also da hätte ich auf keinen Fall ein agiles Projekt daraus gemacht mit dem Wissen von heute.

I: Wie denkst du eignet sich IQP-3 für agiles Vorgehen?

B5: Das passt für mich, letztendlich ist es so, ich muss dem Kunden ja etwas anbieten und auch wenn man im Gewerk ist, man kalkuliert irgendwo eine Kapazität, das heißt ich kann sagen, ich habe ein IQP-3, hab 5 Leute im Einsatz, 160 h im Monat, das ist mein Budget und wie ich das dann regle im Gewerk ist meine Sache, aber damit kann ich das dann sehr gut abschätzen. Das heißt der Kunde bestellt nach IQP-3, 5 Sprints, die kosten ihm 50.000 € und weiß, er hat eine Mannschaft, die entwickelt. Für mich also kein Problem.

I: IQP-4?

B5: Also IQP-4 sind ja kleine IQP-5's, in Form von Arbeitspaketen. Arbeitspakete ist schon ein bisschen schwieriger, wobei auch das geht. Hatte ich noch nicht, dass wir nach Scrum Arbeitspakete abgeliefert haben. Ein Arbeitspaket macht man idealerweise für eine Beschreibung, was da drinnen passieren wird und dann ruft man diese Tickets ab. Also eigentlich eignet sich das mehr für das Ticketsystem, weniger für eine agile Entwicklung, weil wir da letztendlich aus meiner Sicht die Velocity nicht

berücksichtigen. Das heißt, wir haben ein Ticket, Arbeitspaket 1, beinhaltet das, das und das und ist eher geeignet für wiederkehrende Abläufe. Das ist für mich kein agiles Entwickeln. Für mich fehlt da so ein bisschen die Velocity und der Bedarf an Kapazitäten, die ich da reinschreibe. Ich verhandle letztendlich im IQP-4 immer ein Arbeitspaket. Das würde ich tatsächlich nehmen, wenn ich weiß, was ich zu tun habe. Dann weiß ich, was auf mich zukommt.

I: IQP-5?

B5: Das ist das Gesamtwerk. Das kann man sich ja dann wieder zusammenbauen wie man will. Also das ist ja das Ding, das wir am wenigsten machen. Würde ich nur machen, wenn ich ganz genau weiß, was ich machen muss, wenn ich verstehe, was ich machen muss, wenn ich den Kunden kenne. Also ich würde bei der agilen Entwicklung das IQP-5 Thema meiden. Weil da ist ja die Gesamtverantwortung. Letztendlich ist es ja so, wenn es dazu käme, ich krieg eine Software beauftragt, wo ich ganz genau weiß, was am Ende rauskommen soll und ich das intern agil entwickle, ist das meine Sache. Aber letztendlich ist das IQP-5 eine Bestellung und am Ende kommt was raus und das muss die im Lastenheft beschriebenen Funktionen haben. Das heißt agiler Festpreis.

I: Kann es sein, dass der Kunde nach agilem Festpreis beauftragt?

B5: Ja, aber er gibt dir ein Budget vor und am Ende muss was rauskommen. Das geht, ja, aber hatten wir de facto noch nicht so richtig. Im Augenblick würde ich bei agilen Projekten schon IQP-3 bevorzugen.

I: IQP-3 passt ja sehr gut zu agilem Vorgehen. Gibt es dennoch Punkte, die eine Verbesserungspotenzial haben?

B5: Was mir fehlt, ist die rechtliche Abgrenzung eines agilen Projektes innerhalb eines Angebotes. Der agile Festpreis ist ganz oft ein Stichwort das fällt, das greifbar zu machen. Das man einerseits sagt, wir entwickeln agil lieber Kunde, aber andererseits dem Kunden klar gesagt wird, wo sind die Grenzen. Das kriegen wir im Augenblick sehr schwer hin. Beim IQP-3 geht's ja, weil da werden wir tatsächlich nach Stunden beauftragt. Du hast ein Stundenvolumen und sagst, mit dem Stundenvolumen kannst du mit einem Team von 10 Mann, keine Ahnung 5 Sprints. Die bewertet man vorher und sagt ok, das ist die Velocity, so und so viel Tasks, alle 2 Wochen einen Sprint. Was aber wirklich ganz oft passiert, ist, dass wir einen Fachbereich haben, mit dem wir agil entwickeln und zum

Schluss aber der Einkauf kommt und sagt, Schlusstrich und eine Rechnung und jetzt müsste es doch eigentlich fertig sein. Und da fehlt mir so bisschen, wie macht man denn so einen Vertrag sauber fertig. Da bin ich noch nicht so ganz glücklich muss ich sagen. Da prozesstechnisch zu sagen, ja, wir haben eine Agilität, aber die Agilität hat auch ein Ende. Lieber Kunde, entweder du hast ein Lastenheft, wo wir die Funktionen doch am Ende beschreiben müssen, oder du hast ein Open End und das muss aber auch rechtlich in einem Angebot oder Vertragswerk gesichert sein. Es gibt sicher eine Möglichkeit. Das wäre für mich so der absolute Ansatz in Bezug auf alle IQP Modelle. Alle, weil das Problem ist halt, das haben wir überall, wir kriegen diese Abgrenzung halt nicht.

I: Du würdest also im Vertragskonstrukt oder Angebotsphase, Änderungen vornehmen?

B5: Ich würde ganz gern da eine Rechtssicherheit schaffen, weil wir schreiben uns da immer irgendwelche Klauseln rein, wo letztendlich ich mir ziemlich sicher bin, dass die so nicht greifen, wenn es hart auf hart kommt. Wir sind ja nicht die Einzigen, die das Problem haben und es muss ja irgendwie einen Prozess geben, Schranken, wie man diese Agilität in einem Vertragswerk zusammenfasst. A braucht der Kunde das, weil sonst können wir entwickeln und sagen, wir sind agil unterwegs und B müssen wir natürlich auch wissen, wo wir gegenlaufen. Prozessual ist es ja angeschoben. Die Leute werden geschult, es werden Scrum Master ausgebildet. Das war ja eine Voraussetzung, die wir schaffen mussten. Und wie gesagt, der Rest ist dann natürlich noch Firmenkultur, um dieses Mindset noch in diese Teams zu bringen und dann noch zusagen, ok, wir haben jetzt ein Team, die wir mal entwickeln lassen und die Führungskräfte sich mehr oder weniger so ein bisschen beiseite stellen. Das muss wachsen. Das kommt erst, wenn man 1 oder 2 Projekte gemacht hat.

I: Sieht du sonst noch Änderungsbedarf für die IQP Modelle?

B5: Ich wüsste nicht, was ich bei IQP-3 ändern müsste. Das ist eine Vorgabe aus Frankreich, die sich letztendlich bewährt hat. Es ist alles gewerkfähig. IQP-4 ist das klassische Ticketsystem. Das passt für mich. Und IQP-5 ist ein Thema, womit wir uns im Augenblick nicht so viel beschäftigen. Weil das ist Kompletverantwortung für ein Produkt letztendlich oder irgendwie ein Ergebnis, von A bis Z. Das heißt, ich bewege mich viel in IQP-3 und die Möglichkeiten agil zu entwickeln sind aus meiner Sicht gegeben.

I: Wenn du an dein letztes agiles Projekt denkst, hast du das laut dem in der Literatur beschriebenen Prozess durchgeführt und welche Abweichungen gab es in Bezug auf Meetings, Rollen und Artefakte?

B5: Wo fange ich an und wo höre ich auf. Das Einzige, was Scrum konform ist, sind die Standup Meetings, die Sprint Plannings und Reviews. Das funktioniert wirklich nach Scrum. Die Rolle des Scrum Masters, des Product Owners ist aus meiner Sicht nicht die Richtige. Also wir machen es nicht richtig. Es sind die Rollen, aber wir interpretieren sie einfach falsch. Das liegt in der Natur der Sache, weil ich kann jetzt nicht einem Team, was ich aus 5 Nationen eine Woche vorher zusammengewürfelt habe, sagen, so ihr macht das und ich komm dann wieder und ihr sagt mir, wie weit ihr seid. Dann hast du noch den Kunden, der letztendlich zwar Product Owner ist, aber die Rolle nicht begriffen hat und sie auch nicht so ausführt. Der permanent die Kommunikation stört, der Entscheidungen nicht trifft, die er treffen muss, der fremdgesteuert ist von seinen Übergeordneten, wie das so oft ist im Automotive Bereich. Also da ist echt noch eine Menge zu tun, da müssen wir noch umdenken.

I: Gibt es Abweichungen bei den Artefakten?

B5: Das haben wir im Griff, glaube ich. Das funktioniert tatsächlich. Wir haben die Infrastruktur dafür, das mit dem Backlog funktioniert auch gut. Wir haben eine Übersicht, das was drinnen ist, priorisieren wir mit dem Kunden. Was nicht funktioniert ist, was ich schon Eingangs sagte, wir priorisieren und besprechen einen Sprint von 2 Wochen und bekommen dann plötzlich neue Anforderungen, die in den Sprint reingeplant werden, was ja ein absolutes No-Go ist. Unser Verständnis für ein Backlog ist schon da, aber auf Kundenseite definitiv nicht.

I: Werden die Product Backlog Items mit Story Points geschätzt?

B5: Ja, die sind mit Story Points geschätzt und auch mit dem Kunden kommuniziert, wie viele Story Points das Team schaffen kann. Das ist alles konform würde ich sagen.

I: Wie werden die Storys bzw. Epics beschrieben? Aus Sicht des Kunden oder über ein Lastenheft und Ihr wandelt diese dann Scrum konform um?

B5: Wir haben ja tatsächlich kein Lastenheft. Wir entwickeln wirklich agil. Das heißt, wir haben eine Software, die bestimmte Grundfunktionen hat und wir haben ein Backlog und eine Velocity im Team, die wir ansetzen und innerhalb der Velocity sagen wir, ok, in

einem Sprint können wir 100 Story Points abarbeiten. Priorisier mal, was willst du umgesetzt haben bis zum nächsten Sprint. Was fehlt ist dann natürlich tatsächlich dann nach jedem eine stabile lauffähige Version. Das kriegen wir noch nicht so hin. Das funktioniert aus verschiedenen Gründen noch nicht. A, zu viel Einflüsse von außen und B, manchmal auch die Testqualität bei uns selber. Dieser ganze Prozess von Implementieren, Testen und Ausrollen ist noch nicht so ganz. Aber viel hängt auch davon ab, dass wir die 2 Wochen, so wie es gedacht wäre, keine Ruhe haben. Das wir permanent beschossen werden mit Infos und Änderungswünschen. Also überhaupt keine agile Entwicklung. Deswegen ist das Projekt nicht das beste Beispiel für eine erfolgreiche Scrum Implementierung. Der Kunde versteht halt nicht, dass er mit uns die Tasks definiert und dann weg ist. Wenn man sich dann wieder trifft und bespricht, was haben wir umgesetzt, was haben wir nicht umgesetzt, was tun wir im Backlog, was priorisieren wir für den nächsten Sprint. Das versteht er nicht. Der Kunde priorisiert und dann ist er am Dienstag wieder da mit neuen Ideen und am Mittwoch überwirft er die und am Freitag wundert er sich warum nur die Hälfte umgesetzt wurde.

I: Wenn du das Projekt neu starten könntest und du es wieder agil machen willst, welche Aktionen würdest du vornehmen in Bezug auf den Kunden?

B5: Also ich würde auf jeden Fall einen Workshop machen und wie ich schon gesagt habe, ganz klar mit allen Beteiligten und zwar mit den Einkäufern, den Chefs und den Product Owner, also unseren direkten Ansprechpartnern. Einen Workshop machen, ganz klar zu dieser Agilität, wo sich alle Seiten darauf committen, wie das abzulaufen hat. Das wäre aus meiner Sicht das, allererste was ich machen würde. Weil bevor ich nicht ein gemeinsames Mindset habe, brauchen wir das Ding nicht anfangen. Und zwar gezielt den Einkauf, damit er auch versteht, was er da eigentlich beauftragt und was er zu erwarten hat.

I: Würdest du auch die Business Manger miteinbeziehen?

B5: Nein

I: Denkst du, könnte man das auch für nächste Projekte implementieren und wäre das aus deiner Sicht ein kluger Schritt für andere agile Projekte und den Kunden?

B5: Ja klar, ich finde das ist das A und O. Ohne gemeinsames Mindset funktioniert das nicht. Es wird immer Probleme geben, dass man im guten Wissen und Gewissen sein

Projekt fährt und der Kunde sich nicht abgeholt fühlt und er das Thema anders sieht. Und deshalb muss man sich an den Tisch setzen und sagen, so gehen wir vor, das sind unsere Sprints, so wird das geplant, das sind die Tasks, da findet das Planning statt, da findet das Review statt, da findet die Retrospektive statt. Und in den zwei Wochen bist du einfach mal nicht da. Der Weg, wie wir in einem agilen Projekt zusammenarbeiten muss klar sein. Und wenn man dem Einkauf dann auch noch sagt, du darfst dich nicht wundern, dass wir zwei Sprints Bug Fixing und Testing heißen, weil das nun Mal dazugehört, ohne dass ich damit jetzt diskutieren muss, dass du mir erzählst, ich habe eine fehlerhafte Software ausgerollt, weil es ja nicht so ist. Dann hat man schon eine ganze Menge gewonnen.

I: Gibt es sonst noch Wünsche oder Ideen von deiner Seite?

B5: Was mir wichtig wäre, wäre, dass wir einmal einen Prozess draus machen. Gerade wenn man ein agiles Thema startet, dass es dann auch zwingend ist, dass man nicht nur so einen Kickoff, sondern auch einen Workshop macht, glaube ich. Und dass man nach und nach in der Lessons Learned auch diese Themen in der Runde der TD verarbeitet. Ich habe das Gefühl, wir machen ein Projekt und machen Lessons Learned und dann ist es weg. Und dann natürlich, wenn wir mehr Scrum Themen haben, dass wir auch bei uns in der Firma schauen. Wir müssen das bei uns ja auch implementieren, dass der Scrum Master auch der Scrum Master ist. Wir müssen einfach ein gemeinsames Bild auf das Thema haben. Dann funktioniert das überall.

Interview 6

I: Was sind deiner Meinung nach negative Einflussfaktoren für den Projekterfolg bei klassisch gemanagten Projekten?

B6: Änderungen der Anforderungen. Wenn sich während dem Projekt sich Anforderungen ändern, dann ist das immer ein Problem, was mach ich jetzt damit. Bei klassischen Projekten im Sinne, ich habe einen klassischen Werkvertrag, IQP-5 mäßig, als fest definiert und dann ändern sich die Anforderungen. Beim IQP-3 ist das weniger ein Problem, da geht es ja mehr auf Ressourcen, weil da kann man ja drauf reagieren, aber trotzdem bleibt da immer die Frage, was sind denn die Meilensteine jetzt plötzlich, ändern die sich dadurch, oder muss man dann Change Requests in der Regel dann machen. Man muss jetzt echt unterscheiden, ist es ein IQP-3, 4 oder 5 und bei 5 ist es zum Beispiel stärker das Problem, auch wenn die Spezifikation, also die Ausschreibung

des Projektes, die Beschreibung was zu tun ist, nicht vollständig ist, zu schlecht ist oder zu oberflächlich ist. Wobei ich der Meinung bin, es gibt eigentlich keine Spezifikation, die wasserdicht und vollumfänglich komplett ist. Es gibt immer noch Raum, wo man klären, interpretieren muss, usw. Je schlechter die Beschreibung ist, umso problematischer wird es nachher. Und das ist halt oft ein Problem, dass diese Spezifikation nicht so gut ist. Wenn wir von Spezifikationen des Tier1 sprechen, dann hat dieser als Grundlage die Anforderungen von seinem Kunden, dem OEM und die sind dann High Level und damit nicht vollständig oder nicht in dem Detailgrad damit man sofort damit arbeiten kann. Das heißt, wenn die Übernahme des Projektes ist, dann ergeben sich erst die Details und die genauen Beschreibungen, wie man es macht zum Beispiel. Und dann kann man sich vorstellen, alles drum herum, Abschätzung der Umfänge, Timing, reicht die Mannschaft oder nicht, alles hängt ja da dran. Das kann gut gehen, aber oft geht es eben nicht gut. Da stellt man dann fest, das klappt ja so gar nicht. Anderes klassisches Problem ist auch, dass man sagt, ok während des Projektes arbeiten wir basiert auf irgendwelchen Dokumenten, aber die Dokumente sind noch nicht vollständig bei Projektstart. Was am Anfang noch nicht so ein Problem ist, wenn man genug hat, um arbeiten zu können. Aber oft verzögert sich das dann. Das auch so ein bisschen Richtung, was ich am Anfang sagte, verändernde Anforderungen. Ist halt oft auch so, dass die Anforderungen gar nicht richtig vorhanden sind, oder vollständig sind. Und dann werden die im Laufe des Projektes Stück für Stück weiter detailliert und auch das verzögert sich häufig. Und da hat man auch immer das Problem, was mach ich jetzt. Desto mehr es Richtung IQP-5 geht, desto problematischer ist das. Es ist sehr starr, festes Timing, festes Team. Wenn sowas passiert, muss man sofort mit Change Requests arbeiten, neu planen, neue Angebote schreiben, usw.

I: In vielen Unternehmen gibt es einen Trend in Richtung agile Abwicklung. Siehst du den eher beim Kunden oder in der Alten Organisation?

B6: Also wir versuchen das proaktiv an den Kunden zu bringen und zu empfehlen, weil diese agile Methode in Kombination mit IQP-3 für den Kunden ein guter Einstieg ist, wenn er noch nicht so viel mit Gewerken gemacht hat, oder wenn es eine sehr dynamische Umgebung ist. Wo der Kunde jetzt sagt, er möchte das und dies machen und dessen Kunde, also zum Beispiel der OEM wieder mit neuen Sachen kommt oder er sagt, ach wir können das gar nicht so genau definieren. Dann sagen wir, ja kein Problem, braucht ihr auch nicht, wir machen das agil mit IQP-3 und dann habt ihr volle Flexibilität und

Transparenz. Da ist es eher von uns getrieben, aber es gibt inzwischen auch immer mehr Kunden, die das selber machen und vielleicht muss man noch unterscheiden zwischen agil im eigenen Team arbeiten, da muss der Kunde noch nicht mal selber agil sein, wäre natürlich viel besser, muss aber nicht. Und das andere ist, auch der Kunde arbeitet agil und wir machen dann auch agil. Und das Dritte, wo ich meistens davon rede, ist, dass das Zusammenspiel zwischen Kunden und uns auf einem agilen Prozess basiert. Zum Beispiel der Product Owner ist der Kunde und wir haben vielleicht auch Product Owner, Scrum Master. Product Owner auf Kunden Seite und unserer Seite, die sind dafür verantwortlich um zu definieren, was das Team machen soll. Und der Scrum Master ist mehr so ein Prozessmensch. Das ist dann teilweise neu bei vielen Kunden, auch wenn die selber schon agil arbeiten. Aber der Trend ist klar, dass immer mehr agil gemacht wird.

I: Wo denkst du, sieht der Kunde Vorteile bei der agilen Vorgehensweise?

B6: Das sind die klassischen Gründe. Scrum kommt ja ursprünglich aus der SW, aber immer mehr stellen fest, dass es nicht nur auf das beschränkt ist und man es überall einsetzen kann. Und der Grund warum die Kunden das machen, ist der gleiche warum es in der SW entstanden ist. Damit man in vernünftigen Schritten entwickelt, eine realistische Planung und Umsetzung der Planung zu haben. Man plant etwas und dann funktioniert das so nicht, weil Requirements unklar sind, oder sich ändern, technische Gegebenheiten sich ändern oder ich stell fest, das Ding funktioniert so nicht, es ist doch komplexer. Und diese Quereinflüsse führen dazu, dass die ursprüngliche Planung so nicht funktioniert. Und man dann einen Recovery-Plan braucht, um das wieder aufzuholen, usw. Oder der Wettbewerber kommt mit einem neuen Auto raus und man stellt fest, oh, wir müssen ja das und das Feature auch machen, sonst sind wir mit unserem Wagen nicht mehr wettbewerbsfähig. Und das zieht sich bis zum letzten Lieferanten durch. Und das passiert halt, wenn eine Entwicklung 1-2 Jahre läuft, in einem Feld, was recht dynamisch ist, dann passiert das. Und dann, wenn man agil arbeitet, kann man denk ich auch besser darauf reagieren, weil man in kleineren Scheiben plant und dort schneller feststellt, wenn man aus dem Plan rausfällt. Also eine Gesamtplanung braucht man immer, aber ich mach das in kleinen Scheiben und dann stelle ich vielleicht nach 2 Wochen schon fest, ah habe ich nicht geschafft, das war doch komplexer, aufwändiger. Bei der agilen Vorgehensweise ist man immer gezwungen in kleinen Scheiben zu planen und zu schauen, ob man diese auch schafft.

I: Denkst du, gibt es überhaupt gewisse Grundvoraussetzungen, die es benötigt, dass eine agile Projektabwicklung Sinn macht?

B6: Agil macht Sinn in kleinen Teams zu machen. Es macht keinen Sinn ein Team mit 100 Leuten zu haben. Wenn man nun aber Projekte hat, die komplexer sind und die man nicht mit 7 Leuten machen kann, sondern vielleicht 100 Leuten, dann muss man sich jetzt überlegen, wie koordiniere ich zum Beispiel 15, 20 Teams miteinander. Ich muss den Teams ja sagen was sie machen sollen, damit das Hand in Hand geht. Die Ergebnisse der einzelnen Teams müssen auch zusammenspielen. Wenn ich da an SW, HW oder andere Disziplinen, das macht eigentlich nur Sinn, wenn ich einen Prozess habe, der diese Teams synchronisiert. Und da sind wir bei diesen SAFE Thema. Aber was ich auf jeden Fall denke, ist, dass sich Scrum nicht nur auf SW Entwicklung beschränken lässt. Ich denke, dass geht genauso gut bei HW Entwicklung, Mechanik Themen, Konstruktion Themen, usw., wo man mit agilen Teams arbeiten kann.

I: Siehst du gewisse Grundvoraussetzungen beim Mindset des Kunden?

B6: In meinem Umfeld arbeiten wir viel mit dem V-Modell. Es ist immer die Frage, was man unter einer agilen Entwicklung versteht und manche haben den Eindruck, dass, wenn ich agil arbeite, löse ich alles auf, Prozesse. Viele sagen, V-Modell und agil das sind 2 Welten. Aber ich behaupte, das ist Unsinn. Es ist teilweise richtig, aber nicht wirklich. Agil ist ja in der SW, dass ich diese nicht nur definiere und entwickle, sondern auch teste und das alles in einem kleinen Team. Und dieser Mindset beim Kunden, der muss sich auch selber so organisieren. Wenn der Kunde heute eine Testabteilung hat, reine Spezifikationsabteilung, eine reine SW Entwicklungsabteilung und man will ein agiles Team wirklich stringent machen, dass alle drei Disziplinen in einem Team sind, dann löse ich die Abteilungen auf. Das ist dann auch ein notorisches Problem und da muss man sich auch überlegen, wie weit man das treiben will und das ist dann sicherlich auch eine Frage des Mindsets, wie ich denke. Habe ich meine starren Strukturen, mit meinen starren Plänen, mag ja auch sein, aber ich kann da drunter trotzdem agil arbeiten. Und viele sagen, ja, wenn ich agil arbeite, dann läuft ja alles aus dem Ruder, kann ich nichts mehr kontrollieren. Also das ist schon eine Mindset Frage.

I: Denkst du, gibt es Knockout Kriterien, die ein agiles Vorgehen nicht zulassen?

B6: Wenn ein Kunde ein reines IQP-5 Projekt haben will und wirklich die Komplettentwicklung rausgeben will, dann macht das vielleicht Sinn, wenn wir intern

agil arbeiten, aber das Zusammenspiel mit dem Kunden vereinbaren wir gemeinsam, zum Beispiel den Backlog für den nächsten Sprint und wenn der Sprint durch ist, haben wir die Stories abgearbeitet, die gehen dann auf Approved vom Kunden und dann kommt der nächste Sprint, so dass der Kunde immer Teil des Prozesses ist. Wenn man das machen will, dann wird das mit dem IQP-5 eigentlich unsinnig, das ist entgegen der IQP-5 Idee. Beim IQP-5 sagt der Kunde, hier ist die Spec., jetzt mach, ich möchte das im Detail auch gar nicht wissen und steuern, sondern das ist dann deine Verantwortung. Und klar machen wir dann immer ein Review, einmal im Monat oder alle 2 Monate, keine Ahnung. Da geht es aber nur um, was ist das Ergebnis, sind wir soweit mit dem was wir liefern müssen. Das ist aber nicht wirklich agil, das ist das klassische Entwicklungskonzept. Also ich würde bei IQP-5 und dem Zusammenspiel mit dem Kunden sagen, das ist nicht so geeignet dafür. Natürlich können wir intern immer agil und der Kunde kann für seine Sachen agil arbeiten, aber so das Zusammenspiel macht wenig Sinn.

I: Wenn der Kunde aber ein IQP-5 Modell fordert und sagt, ok, ich bin damit einverstanden, dass wir zusammen agil entwickeln. Er also einen Product Owner auf seiner Seite stellt, wie siehst du das dann?

B6: Da ist halt die Frage, was daran IQP-5 ist und wie groß diese Sprints sind. Das widerspricht so ein bisschen der Idee der agilen Entwicklung. Das agile Entwickeln ist ja nicht nur, dass ich in kleinen Scheiben arbeite, sondern da ist auch die Fähigkeit des Teams, Teil davon und diese Retrospektive. Ich sag, ich habe 5 Leute. Dann schau ich, was können die 5 Leute denn machen und dann mache ich Stories, die diese 5 Leute auch schaffen. Beim IQP-5 sagt der Kunde, mir ist egal wieviel Leute ihr habt. Man könnte es so machen, die Stories vorgeben, die in jedem Sprint zu tun sind oder die Ergebnisse, wie auch immer man das nennt, aber es interessiert mich nicht, wie groß das Team ist. Das entspricht eigentlich nicht der Idee des agilen Arbeitens. Wenn wir die Ressourcen Komponente rausnehmen, dann macht das nicht mehr viel Sinn, das agil zu machen. Agil lebt ja davon, das frühzeitig zu erkennen, was das Team hinkriegt und was nicht und darauf reagieren zu können und den nächsten Sprint neu zu planen oder so zu planen, dass es wieder funktioniert. Also das ist ja der Bedarf eines permanenten Anpassens, warum man agil überhaupt macht, weil man früher genau das Problem hatte, dass man Sachen geplant hat, die nicht funktionieren, weil sich Anforderungen ändern oder Sachen doch komplexer sind als man vorher abgeschätzt hat. Und IQP-5 lebt ja davon, dass ich vorher alles definiere und wir müssen dann abliefern. Ja, wir können es machen, halte es aber

für fraglich, wie weit das Sinn macht. Man kann es machen, aber es geht dann ein wichtiger Hebel, Vorteil, des agilen Entwickelns verloren, wenn man es konsequent nach IQP-5 macht.

I: Soweit ich weiß, hat Alten schon ein paar agile Projekte mit Scrum durchgeführt. Gab es hier auch Nachteile dieser Methode?

B6: Es gibt verschiedene Projekte, die mehr oder weniger agil sind. Mehr oder weniger deshalb, weil ich nicht Scrum von A bis Z durchführe, aber von der Idee, also das Zusammenspiel mit uns und Kunde aber agil ist. Der Kunde ist der Product Owner und wir haben auch einen Product Owner und wir definieren zusammen die Stories und Backlogs. Da gibt es ein Projekt, da werden die Aufwendungen mit Story Points dargestellt und das ist nachher auch die Abrechnungsgrundlage für den Kunden. Also der Leistungsnachweis. Das kennst du ja, die Stories haben Story Points und die reflektieren ja die Arbeitsumfänge. Nicht nur Komplexität, sondern wir sagen, Komplexität ist gleich Arbeitsumfang. Und das führt aber dazu, dass diese Story Points alle getrackt werden müssen und immer passen müssen. Die Story Points die wir leisten, müssen auch mit unserem Stundenkontingent und Aufwendungen passen. Und wenn das nicht zusammenpasst, haben wir vielleicht weniger, oder verdienen sogar mehr. Das heißt, von der Idee finde ich es nicht schlecht, aber in der Umsetzung ist es relativ aufwändig. Ich habe andere Projekte, wo wir das aber nicht mit den Story Points machen. Klar wir haben ein Backlog und Aufgaben, die besprechen wir auch mit dem Kunden. Und am Ende der Sprints besprechen wir, wie weit wir gekommen sind und gut ist. Und wenn wir dann das erreicht haben, was wir erreichen wollten, dann passt das. Wenn der Kunde auch sieht, dass wir vernünftig gearbeitet haben, dann kriegen wir die Arbeit auch bezahlt. Das ist nicht festgemacht an den Story Points und das erleichtert natürlich das PM Management und Controlling immens. Da gibt es viele Knackpunkte mit den Story Points, Angebot zum Beispiel. Im Großen und Ganzen funktioniert Scrum sehr gut. Ich bin ein Fan davon. Wenn man es sehr transparent macht, sieht der Kunde sehr genau was wir machen, ob jemand krank ist oder nicht. Das kann schon dazu führen, dass wir sehr eng vom Kunden überwacht werden. Aber das muss nicht so sein. Viele weitere Nachteile wüsste ich jetzt nicht. Außer wenn man es mit einem IQP-5 macht, da haben wir ja schon kurz darüber gesprochen, da sehe ich das eher schwierig.

I: Wie gut eignet sich aus deiner Sicht das IQP-3 Modell für agiles Vorgehen a la Scrum und warum?

B6: Das eignet sich eher als IQP-5. Also ich rede jetzt nicht wie wir intern arbeiten, sondern in Bezug auf das Zusammenspiel mit dem Kunden. Also IQP-3 eignet sich am besten, auch wenn es kein reines IQP-3 ist. Ich bin der Meinung immer, Scrum ist ein IQP-3.5. Das ist zwischen 3 und 4, weil 4 ist ja laut Definition, wiederkehrende Aufgaben mit festen Umfängen, so ein Abrufsystem, ein Ticketsystem. Ich habe verschiedene Tickets mit verschiedene Komplexitäten und dahinter sind verschiedene Preise, aber die sind fest. Das Ticket mit der Nummer 14, mit dem Typ und der Komplexität, kostet immer so und so viel Geld, weil wir abgeschätzt haben, so ein Ticket hat so viele Aufwendungen. Und so zieht sich das durch. Das ist im Grund ein IQP-5 in Raten, in kleinen Schritten, in vielen kleinen IQP-5. Aber Scrum ist halt nicht so, wie ich eben schon erzählt hab. Scrum ist auch kein reines Ressourcenmodell, wo man mit IQP-3 dann sagt, ok ich mach was du brauchst und eigentlich mach ichs nach Stunden, klar ich mach dann ein Ergebnis, im Abnahmeprotokoll steht drin, was der gemacht hat, aber ansonsten werden die Stunden abgerechnet, wie wir auch gearbeitet haben. Das ist eigentlich auch kein Scrum. Scrum ist da was dazwischen. Auf der einen Seite ist das klar so Ressourcenmodell getrieben und das ist transparent beim Kunden. Das heißt, ich habe 4 Leute im Scrum Team und die 4 Leute können auch nur bestimmte Mengen machen und wenn, dann mehr gebraucht wird, dann braucht man mehr Leute oder müssen das im nächsten Sprint machen und das auch transparent. Also er weiß, es sind 4 Leute im Team und die können nur so und so viel schaffen in dem Sprint. Aber auf der anderen Seite werden auch Aufgaben definiert und die werden am Ende des Sprints auch bewertet, ob diese fertig sind. Und die sollten ja dann auch fertig sein, das ist ja auch das Ziel. Und nur in Ausnahmen mit der Retrospektive sagt er ok, das ist nicht ganz fertig, das wird dann nochmal gemacht. Das Abliefern ist schon ein wichtiger Faktor dabei, das harmonische Zusammenspiel von Ressourcen, die da sind und gelieferten Ergebnissen. Und daher ist das aus meiner Sicht so ein 3.5, also das zwischen 3 und 4 ist. Oder man kann auch sagen, es ist ein IQP-3 mit stärkerem Fokus auf Abliefern. Weil man ja vorher genau definiert, was zu tun ist mit den Stories, aber mit dem Kunden vereinbart, dass die Aufgaben, Stories, die zu tun sind in einem Sprint, das würde man ja bei einem reinen IQP-3 gar nicht so machen. Natürlich wird der Kunde uns sagen, das und das brauchen wir und dann aber im Endeffekt rechnet ihr das ab und fertig. Bei dem agilen ist der Druck denk ich mal etwas höher, dass man bestimmte Sachen auch abgeliefert.

I: Siehst du beim IQP-Modell die Notwendigkeit, Änderungen oder Anpassungen vorzunehmen, wenn man agil entwickeln will?

B6: Angebotsgestaltung definitiv, da sind wir ja auch dran. Da haben wir auch schon mehrere Anläufe gemacht, wo ich auch involviert bin oder war. Wenn ich agile Angebote mache, dann ist das auch definitiv im Angebot definiert. Und das weicht an den meisten Stellen schon ein Stück weit ab, von dem Standardtemplate, das wir haben. Bei manchen Sachen muss man es auch tatsächlich überlegen, wie man es auch dort formuliert und integriert. Solche Sachen, wie eben mit den Story Points. Nehme ich das auf als Maß der Leistung, oder nicht. Die Angebote sehen da wirklich anders aus. Da schreibe ich zum Beispiel High Level, was zu tun ist, damit man weiß, welche Leute mit welchen Skills brauch ich überhaupt, so der große Plan. Und dann kann man vielleicht noch Beispiele nennen, wie so eine Story, Backlog oder Scrum Planung aussieht. Und dass man dann exemplarisch typische Aufgaben, Stories definiert, wie die dann aussehen. Wenn man dann mit Story Points arbeitet, dann muss man das dann natürlich auch genau beschreiben, in so einem Angebot wie man damit umgeht. Zum Beispiel 4 Punkte, wieder indem man Beispiele von Stories macht. Das ist Komplexität 1 also 1 Story Point. Diese Sachen muss man dann schon in einem Angebot definieren und sowas ist in dem Template gar nicht vorgesehen zurzeit. Also Angebotsvorlagen so wie sie jetzt sind, müssen wir ein Stück weit dafür anpassen. Das macht, glaub ich, zur Zeit jeder selber. Ansonsten ist es eher so, dass die Leute wissen wie das funktioniert, dass sie die richtige Ausbildung haben, auch, dass wir das dem Kunden erklären, dass er das versteht. Und dass man diese Abläufe auch dem Kunden gegenüber entweder in dem Angebot beschreibt oder wo auch immer.

I: Wie siehst du jetzt diese gesamte Thematik bei einem IQP-4 Modell? Wie eignet sich das Modell mit Begründung und siehst du hier auch die Notwendigkeit für Änderungen?

B6: Laut meinem Verständnis sind IQP-4 kleine IQP-5 Projekte. Wenn man das jetzt so versteht als Ticketsystem oder Abrufsystem von Arbeitspaketen, kleinen Arbeitspaketen, das ist jedes Paket für sich ein kleinen IQP-5. Und damit haben wir schon das Problem, dass es eigentlich nicht mehr agil ist, weil die Ressourcengröße im Auftrag nicht definiert ist, oder dem Kunden egal ist. Das einzige agile, was du jetzt hast, ist nur welche Tickets mach ich denn jetzt. Man kann sagen, jede Story ist ein Ticket und dann sag ich, welche Stories müssen gemacht werden. Für Story 1 brauch ich einen Mann, für Story 2 brauch

ich einen halben und Story 3 brauch ich 0,2 und alles zusammen macht irgendwie, ich habe 4 Leute, das ist auch klar und dann kann ich das auch planen. Das ist wie beim Agilen, aber das Problem ist hinten raus, wenn das nicht so funktioniert und die Retrospektive stattfindet, dann habe ich ein Problem, weil ja diese Tickets fest definiert sind vom Preis. Ich meine, man könnte das ohne IQP-4 machen, oder man kann das Ganze im Grunde genommen auch das agile IQP-4 nennen, wenn ich keine Festpreise hinter die Tickets schreibe. Sagen wir mal so, wenn ich es erlaube eine Retrospektive zu machen und gegebenenfalls diese Preise zu verändern, wenn ich feststelle, dass es doch komplexer war und ich das auch vernünftig begründen kann. Wenn ich diese Tür aufmache, dann würde auch IQP-4 funktionieren.

I: Also das müsste man dann im Angebot festlegen.

B6: Ja genau. Also dann wäre im Grunde was ich beschrieben habe. Diese 3.5 wäre dann 4 und dann ist das 4. Aber das 4, so wie wir das verstehen und auch dem Kunden gegenüber schmackhaft machen, nach dem Motto hier hast du feste Preise und Ticketsystem. Also ich sage, ich will Ticket 1, Nr. 14, Nr. 18 und 256 und ich weiß jetzt schon genau, was es kostet und wie lange es braucht, dann ist das nicht mehr agil. Dann habe ich keine Retrospektive mehr. Also da ist jetzt die Frage wie man IQP-4 definiert. Dann habe ich nur ein halbes agiles IQP-4. Agil ist schon, dass man die Stories agil definiert, die dann meine Tickets darstellen und dann auch mein Backlog. Das passt ja alles, aber was nicht passt, ist der flexible Umgang mit den Umfängen der Stories und das ist genau der Kern des agilen Arbeitens. Agil ist ja nicht nur, dass ich das in kleine Scheiben schneide, sondern, dass ich das immer wieder agil anpasse diese Planung durch die Retrospektive. Wenn ich die nicht mehr habe, weil der Kunde sagt das ist mir egal, dann können wir zwar das intern agil machen, klar das geht, aber gegenüber dem Kunden und wie wir den Kunden abrechnen, das ist nicht mehr agil. Man kann sagen, diese Tickets im IQP-4 sind im Grund meine Stories und alle 2 Wochen definiere ich mit dem Kunden zusammen die Tickets und das Scrum nimmt das als Backlog, dass ist ok. Aber das Problem ist, wenn ich bei einer Retrospektive feststelle, das ich für ein Thema doch länger brauche, dann würde der Kunde sagen, ja gut, aber mehr Geld kriegst du trotzdem nicht.

I: Siehst du hier noch weitere mögliche Anpassungen, oder Änderungen die man machen müsste?

B6: Das habe ich ja eben schon gesagt, das gilt ja grundsätzlich. Also das kommt auch immer darauf an, wie die Aufgaben gestrickt sind, also wie vollumfänglich die Aufgaben sind. Oft ist es ja so, dass wir zum Beispiel eine Aufgabe haben, SW zu entwickeln, oder wir haben die Aufgabe zu testen. Eine Idee bei dem Agilen ist ja auch, dass dieses Scrum Team selber, mehrere Disziplinen in dem Team haben, also Tester, eine Person für Spezifikation, einen Entwickler und so weiter. Wenn man solche Aufträge hat vom Kunden, der sagt, mach alles, also die SW Entwicklung mit Test und Spezifikationsentwicklung, dann kann das gegebenenfalls bei uns bedeuten, dass wir uns so natürlich auch organisieren müssen. Dass wir dann Sprint Teams aufsetzen, wo dann alle Disziplinen in einem Team sind, aber soweit ich das so sehe, ist das nicht so häufig ein Problem, weil wir in der Regel sowieso mehrere Teams haben, wenn es größer ist und wir uns so organisieren können. Also ich würde mal sagen, für uns ist das kein so großes Thema. Dass wir uns organisatorisch umstrukturieren müssen, würde ich eher nein sagen, weil wir uns eh immer projektspezifisch organisieren. Es gibt keine starre Organisation. Früher hatten wir mal ein Center of Competence, da war das noch anders. Wir haben ja nicht wie beim Kunden eine Testabteilung, eine SW Entwicklungsabteilung, eine Abteilung für Requirements, eine Mechanik Abteilung. Wenn wir Projekte haben, dann schauen wir, welche Leute brauchen wir und bauen das Team in dem Projekt so zusammen, wie wir es benötigen. Ich weiß, es gibt Dienstleister, die sind stärker nach Disziplinen organisiert, da kann es ein Problem werden und natürlich auch bei unseren Kunden. Deswegen ist es auch eine Herausforderung bei den Tier1, den großen Konzernen oder auch OEMs, die halte nach diesen Fachdisziplinen organisiert sind. Wenn man jetzt ein Sprintteam macht, wo Leute aus verschiedenen Fachdisziplinen zusammenkommen müssen, dann ist die Frage, wie werden die organisiert unter welcher Organisation laufen die und wer ist dafür verantwortlich. Wer ist der disziplinarische Vorgesetzte, wer ist der fachliche Vorgesetzte. Aber dieses Problem haben wir bei Alten nicht.

I: Wie gut eignet sich das IQP-5 Modell aus deiner Sicht mit Begründung und wo siehst du bei diesem Modell die größten Probleme?

B6: Wie gesagt, im Zusammenspiel mit uns und dem Kunden, agil über IQP-5 halte ich für sehr schwierig oder eigentlich fast unmöglich, weil da fehlt der Aspekt der Retrospektive. Man versucht ja mit dem Agilen in kleineren Stücken zu planen und damit zahlungssicherer zu werden und wenn dann Requirements-Änderungen reinkommen, auf

die reagieren zu können. Das kann man ja auch machen, aber das heißt bei IQP-5, dass man sofort einen Change Request machen muss. Und wenn ich dann jedes Mal einen Change Request machen muss, dann ist die Frage, bin ich denn noch bei IQP-5, oder ist das nicht eigentlich wieder 4 oder was. So ein Change Request ist schon möglich, dieser sollte aber eher die Ausnahme sein und nicht Bestandteil des Prozesses und alle zwei Wochen gucke, habe ich eine Änderung und dann einen Change Request mache. Sondern eher, jetzt kommt eine Requirements-Änderung vom Kunden, dann wird das analysiert und dann macht man dann einen Change Request, was ja heißt, ich mache eine Angebotsanpassung. Das heißt, wieviel Leute brauch ich dafür, was kostet das, wann bin ich damit fertig. So eine Art Planungsgeschichte, aber das auf Vertragsform zu bringen, dass würde ich denke ich mal nicht im 2-Wochen Sprint Rhythmus machen. Jedes Mal, weil eine Story drei Tage länger gebraucht hat, eine Vertragsänderung mache. Bei IQP-5 ist das schwierig und da stellt sich bei mir die Frage, ist das überhaupt IQP-5. Man muss hier aufpassen, wenn man das nicht so reflektiert, was ich da gerade gesagt habe, also nicht darüber nachdenkt, besteht hier sicherlich die Gefahr, dass der Kunde sagt, ja wir machen Festpreis, IQP-5, aber agil, weil wir arbeiten alle agil. Und dann machen wir unsere Sprints, Backlogs, bewerten das nachher in den Sprint Meetings. Ja, das können wir schon so machen, aber im Endeffekt ist das ein Festpreis nachher. Also dann ist die Frage, was ist denn das. Ist das jetzt wirklich agil Entwickeln, oder ist das nur ein anderes Tracking der Fortschritte eines Projektes. Was ich damit sagen will, man muss halt aufpassen, dass wir hier nicht, ja wir machen agil, agile Prozesse, aber wir haben Festpreis IQP-5 und dann denken wir, ja alles gut und da können wir ja dann reagieren und wenn es dann aber so weit kommt, sagt der Kunden, ja können wir schon machen, aber mehr Geld gibt es da nicht, das ist ja ein Festpreis. Das ist die Gefahr, das nicht klar ist auf beiden Seiten, was man jetzt mit agil wirklich meint.

I: Siehst du allgemein Änderungsbedarf in Bezug auf die agile Entwicklung?

B6: Also erst mal denke ich, der Trend ist da, zweitens müssen wir uns noch besser darauf vorbereiten, was das Vertragswerk angeht. Die Templates, die jetzt da sind, geben das nicht so optimal her. Und auch im Vertrieb glaube ich, dass viele das noch nicht so richtig verstanden haben, was das wirklich heißt, wie weit das geht, wie weit das Sinn macht und wie man das dem Kunden schmackhaft machen kann. Also der Kunde hat hohe Flexibilität und Transparenz und vor allem was viele Kunden bezweifeln, dass es immer noch ein sauberes Gewerk ist. Und da ist es auch wichtig, dem Vertrieb klar zu machen,

dass das auch geht. Dass das sowohl rechtlich sauber ist, wenn man es richtig macht, als auch, dass der Kunde auch Vorteile hat und es kein reines IQP-3 ist, wo die gesamte Verantwortung beim Kunden liegt, sondern das hier weiterhin Leistung geliefert wird. Und der Kunde auch sagen kann, eure Leistung stimmt nicht. Bei einem reinen IQP-3 ist das schon etwas schwieriger. Die Balance zwischen ist es jetzt leistungsbezogen aber auch trotzdem immer noch ressourcengetrieben, da den richtigen Weg zu finden ist nicht so einfach und auch nachher im Projekt zu erklären, wenn es dann nochmal Probleme gibt. Das ist glaube ich eine Herausforderung für alle, sowohl beim Kunden als auch bei uns.