



Hochschule für Angewandte
Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelor-Thesis

Vor- und Zuname

Adrian Steffen Meier



Titel:

Ein Vergleich von Large-Scale Scrum [LeSS] und dem Scaled Agile Framework [SAFe] im Hinblick auf ihre Eignung für den Einsatz in Großprojekten unter Betrachtung ausgewählter Dimensionen

Erstprüfer: Prof. Dr. Rüdiger Weißbach

Zweitprüferin: Prof. Dr. Andrea Zirm

Eingereicht am: 23.08.2019

Hochschule für Angewandte
Wissenschaften Hamburg

Fakultät Wirtschaft und Soziales
Department Wirtschaft

Marketing/
Technische Betriebswirtschaftslehre

Inhaltsverzeichnis

Inhaltsverzeichnis	i
Abbildungsverzeichnis.....	iii
Abkürzungsverzeichnis.....	iv
Abstract.....	v
1. Einleitung.....	1
1.1 Problemstellung und Zielsetzung	2
1.2 Aufbau der Arbeit.....	2
2 Klassisches und Agiles Projektmanagement	4
2.1 Das klassischen Projektmanagement.....	4
2.2 Grenzen des klassischen Projektmanagements	5
2.3 Definition von „Agil“	7
2.4 Agiles Projektmanagement in der Praxis	8
2.4.1 Vor- und Nachteile	8
2.4.2 Anwendungsgebiete	9
2.5 Historie	10
2.6 Das Agile Manifest.....	12
2.7 Agile Prinzipien.....	13
2.8 Scrum.....	14
2.8.1 Grundlagen und Begrifflichkeiten von Scrum	14
2.8.2 Theoretischer Ablauf von Scrum.....	21
2.9 Kanban.....	23
3 Skalierbare agile Projektmanagementmethoden	25
3.1 Large Scale Scrum (LeSS)	25
3.1.1 Grundlagen und Begrifflichkeiten von LeSS	26
3.1.2 Theoretischer Ablauf von LeSS	30
3.1.3 Theoretischer Ablauf von LeSS Huger.....	31
3.2 Scaled Agile Framework (SAFe)	32
3.2.1 Grundlagen und Begrifflichkeiten von SAFe.....	33
3.2.2 Theoretischer Ablauf von SAFe.....	39

4	Auswirkung von LeSS und SAFe auf den Einsatz in Projekten unter Berücksichtigung ausgewählter Dimensionen	41
4.1	Auswahl der für die Analyse relevanten Dimensionen.....	41
4.2	Agilität der Unternehmenskultur	42
4.2.1	Theoretische Grundlage	42
4.2.2	Analyse der Frameworks hinsichtlich der Agilität der Unternehmenskultur	44
4.3	Größe der Entwicklungsorganisation.....	47
4.3.1	Theoretische Grundlage	47
4.3.2	Analyse der Frameworks hinsichtlich der Größe der Entwicklungsorganisation.....	48
4.4	Komplexitätsgrad eines Projektes.....	50
4.4.1	Theoretische Grundlage	50
4.4.2	Analyse der Frameworks hinsichtlich des Komplexitätsgrads des Projektes	51
4.5	Gesamtüberblick und Darstellung der Eignung der untersuchten Frameworks ..	55
5	Schlussbetrachtung	59
5.1	Zusammenfassung.....	59
5.2	Fazit.....	59
5.3	Ausblick.....	60
	Litertaturverzeichnis.....	61
	Eidesstattliche Erklärung.....	67

Abbildungsverzeichnis

Abb. 1: Darstellung der Phasen bei plangetriebenem und agilem Vorgehen	6
Abb. 2: Die Produkt Backlog Items werden in ein Sprint Backlog gezogen	20
Abb. 3: Ablauf von Scrum.....	21
Abb. 4: Ablauf von LeSS.....	30
Abb. 5: Ablauf von LeSS Hufe 31	31
Abb. 6: Ablauf von SAFe	39
Abb. 7: Darstellung der Frameworks anhand der Dimensionen Agilität der Unternehmenskultur und Größe der Entwicklungsorganisation.....	56
Abb. 8: Darstellung der Frameworks anhand der Dimensionen Komplexitätsgrad des Projektes und Größe der Entwicklungsorganisation	57
Abb. 9: Darstellung der Frameworks anhand der Dimensionen Komplexitätsgrad des Projektes und Agilität der Unternehmenskultur	58

Abkürzungsverzeichnis

SAFe	Scaled Agile Framework
LeSS	Large Scale Scrum
PMI®	Project Management Institute
PO	Product Owner
APO	Area Product Owner
PBR	Product Backlog Refinement
ART	Agile Release Train
PI	Programminkrement
KPI	key performance indicator

Abstract

Die vorliegende Bachelorarbeit vergleicht die beiden agilen Frameworks Scaled Agile Framework (SAFe) und Large Scale Scrum (LeSS) zu ihren Stärken und Schwächen in verschiedenen Projektkonstellationen. Dies geschieht mit Hilfe von relevanten Projekteigenschaften, die eine Unterscheidung und Analyse beider Frameworks ermöglichen. In dieser Arbeit wird eine umfassende theoretische Grundlage in die agilen Ansätze gegeben. Neben den agilen Methoden auf der Teamebene, werden die Frameworks weitreichend erläutert. Die theoretische Basis der Bachelorarbeit und die Einteilung und Analyse der Projekteigenschaften fand mit Hilfe von tiefgreifender Recherche der aktuellsten Fachliteratur statt.

Die Auswahl der für den Vergleich der Modelle relevanten Dimensionen wird in der Arbeit detailliert dargestellt. Folgende Dimensionen sind für die Frameworks von Bedeutung: die Agilität der Unternehmenskultur, der Komplexitätsgrad des Projektes und die Größe der Entwicklungsorganisation. Anhand dieser Dimensionen konnte eine Analyse beider Frameworks stattfinden. Im Vergleich wurde deutlich, dass sich SAFe eher für große, komplexe Projekte in Organisationen mit einer geringen agilen Kultur eignet, während LeSS seine Stärken bei kleinen bis mittlere Projekte bei einer agilen Unternehmenskultur besitzt. Das Ergebnis wurde zusammengefasst und für eine bessere Übersicht visualisiert. Der Autor erhofft sich damit einen wissenschaftlichen Erkenntnisgewinn im Themenbereich der beiden Frameworks Scaled Agile Framework und Large Scale Scrum.

1. Einleitung

Die Märkte, in denen Unternehmen agieren, werden immer komplexer. Dynamische Umweltbedingungen treffen auf sich schnell ändernde, individuelle Kundenforderungen. Weiterhin sind beschleunigte Produkt- und Technologiezyklen vermehrt notwendig, um sich gegen den Wettbewerb zu behaupten. Diese Komplexität hat zur Folge, dass der Weg zur Zielerfüllung in Projekten schwer vorab festgelegt werden kann.¹ In diesem Spannungsfeld ist die Softwareentwicklung besonders stark betroffen. Als Lösung werden die traditionellen Projektmanagementmethoden vermehrt von agilen Ansätzen abgelöst. Mittlerweile gehören agile Vorgehensmodelle wie Scrum, XP Programming oder Kanban in der Softwareentwicklung zum Standard.² Autonom arbeitende, cross-funktionale Teams entwickeln Schritt für Schritt in kurzen sich wiederholenden Zeitabständen Produktbeiträge bis die Zielerfüllung erreicht ist. Ständiges Feedback und eine intensive Stakeholderkommunikation gewährleisten zudem eine gute Qualität bei komplexen, unsicheren Kundenanforderungen in dynamischen Märkten.³ Die Prinzipien und Praktiken aus agilen Vorgehensmodellen lassen sich am Besten in kleinen Teams von maximal neun Personen umsetzen, um die nötige Kommunikation und Entscheidungsfindung zu erreichen.⁴ Größere Projekte lösen daher einen Widerspruch aus. Sie benötigen mehr Mitwirkende und somit mehr Teams, um die Fülle an anstehenden Aufgaben bearbeiten zu können. Eine Antwort darauf bieten agile Frameworks. Sie versprechen die Agilität auch auf organisationaler Ebene einzuführen und auf operativer Stufe ein agiles teamübergreifendes Arbeiten zu gewährleisten.⁵ Der Fokus dieser Arbeit liegt auf den beiden agilen Frameworks Large Scale Scrum (LeSS) und dem Scaled Agile Framework (SAFe). Beide Frameworks sind in der Praxis zurzeit am weitesten verbreitet.⁶ Beide Frameworks sind für Unternehmen besonders interessant, da sie als einzige genügend Use Cases, viele Informationen, eine aktive Community und weitreichende Trainingsangebote bereitstellen.⁷

¹ Vgl. Werro (2018), S. 1

² Vgl. Roock/Wolf (2016), S. 6

³ Vgl. Hanschke (2017), S. 1 ff

⁴ Vgl. Roock/Wolf (2016), S. 20

⁵ Vgl. Vgl. Mathis (2018), S. 1 f.

⁶ Vgl. Zillmann (2019), S. 21

⁷ Vgl. Uludağ et al. (2017), S. 125

1.1 Problemstellung und Zielsetzung

LeSS und SAFe sind von ihrer Struktur sehr unterschiedlich. Während LeSS auf Scrum basiert⁸, beinhaltet SAFe eine Vielzahl neuer Rollen, Prozesse und Ereignisse.⁹ Im Gegensatz zu den agilen Ansätzen auf der Teamebene, verändert die Einführung eines Frameworks eine Organisation nachhaltig und ist maßgeblich für den Erfolg oder Misserfolg einer großen Unternehmung verantwortlich.¹⁰ Sollte sich das Management dazu entschließen agil zu skalieren, ist die Auswahl des richtigen Frameworks dementsprechend von großer Bedeutung. Problematisch ist, dass beide Frameworks relativ jung sind.^{11,12} Ein detaillierter, akademischer Vergleich beider Methoden wurde bisher nicht vorgenommen. Bestehende Veröffentlichungen wurden in der Regel von Unternehmen vorgenommen, deren Sichtweise subjektiv sein kann. Diese Arbeit soll helfen, das Thema durch wissenschaftliche Erkenntnisse zu bereichern. Sie gibt eine Einführung in die Agilen Ansätze, erklärt die wichtigsten Methoden und beschreibt die beiden Frameworks. Weiterhin werden Dimensionen von Projekten definiert, anhand derer die beiden Frameworks erfasst werden. Somit können Entscheidungsträger ihre Projekte grob einer Einordnung unterziehen und das richtige Framework für nachfolgende intensivere Analysen auswählen.

1.2 Aufbau der Arbeit

Der Aufbau der Arbeit orientiert sich an der Zielsetzung und dem nötigen Wissensaufbau. In Kapitel zwei werden die theoretischen Grundlagen aufgezeigt. Im Zuge dessen werden zentrale Begriffe wie das klassische Projektmanagement und seine Grenzen erläutert. Anschließend wird sich dem agilen Projektmanagement zugewandt, indem der Begriff „agil“ definiert wird. Nachfolgend werden die Vor- und Nachteile, das Anwendungsgebiet, die Historie und die Werte und Prinzipien erklärt. Danach werden die agilen Methoden Scrum und Kanban beschrieben. Der Wissenstransfer ist nötig, um die beiden skalierten Frameworks LeSS und SAFe in Kapitel drei in den Focus der Betrachtung zu rücken und erläutern zu können. Als Nächstes wird die Auswahl, der für

⁸ Vgl. Larman/Vodde (2017), S. 8 ff.

⁹ Vgl. Mathis (2018), S. 9 ff.

¹⁰ Vgl. Maximini (2018), S. 122 ff.

¹¹ Vgl. Mathis (2018), S. 208

¹² Vgl. Uludağ et al. (2017), S. 126

die Betrachtung der beiden Frameworks relevanten Dimension in Kapitel 4 beschrieben. Danach erfolgt, nachdem jede Dimension ausführlich beleuchtet wurde, eine Einordnung der Frameworks. Enden wird die Arbeit mit einer Synthese und Visualisierung der Ergebnisse und einer Schlussbetrachtung in Kapitel 5.

2 Klassisches und Agiles Projektmanagement

2.1 Das klassischen Projektmanagement

In diesem Kapitel wird das klassische Projektmanagement beschrieben. Ein Projekt wird in der Literatur vom Project Management Institute (PMI®) als „[...] zeitlich begrenztes Vorhaben mit dem Ziel, ein einmaliges Produkt, eine einmalige Dienstleistung oder ein einmaliges Ergebnis zu schaffen“ definiert.¹³ Die Einmaligkeit und die zeitliche Begrenzung wird auch in anderen Literaturquellen erwähnt.^{14,15} Umsetzen lässt sich dieses zeitlich befristete Aktionsbündel mit dem klassischen oder auch traditionellen Projektmanagement. Diese Anwendungsdisziplin schafft eine konkrete Planung, Steuerung und Koordination, um vielseitig Projekte erfolgreich zum Ziel zu führen.

Zur Durchführung eignen sich verschiedene Vorgehensmodellen wie PMBOK, ICB und PRINCE2, die Aufgaben und Prozesse als Projektmanagement-Standards abbilden und damit eine Arbeitshilfe anbieten.¹⁶ Daneben gibt es die deutschen Normen DIN 69901 oder ISO 21500, die Begriffe, Grundlagen, Prozesse und Methoden beschreiben.¹⁷ In Deutschland wird das Projektmanagement in der DIN-Normreihe 69901 als „[...] Gesamtheit von Führungsaufgaben, -organisation, -techniken und -mitteln für die Initiierung, Definition, Planung, Steuerung und den Abschluss von Projekten [...]“ definiert.¹⁸ Daraus lässt sich herauslesen, dass das Projektmanagement nicht die Umsetzung, sondern das Management des Problemlösungsprozesses darstellt.¹⁹

Der grobe Ablauf des klassischen Projekt Managements ist in der Regel bei allen Vorgehensmodellen identisch. Erst müssen die gewünschten Ergebnisse ermittelt und darauf aufbauend der Projektverlauf festgelegt werden.²⁰ Ein planmäßiges Vorgehen ist also fundamental für das klassische Projektmanagement. Die Projektmanager haben einen großen Einfluss auf den Erfolg eines Projektes und müssen diesbezüglich umfang-

¹³ Project Management Institute (2008) S. 4

¹⁴ Vgl. Deutsches Institut für Normung (2009) S. 155

¹⁵ Vgl. Kusay-Merkle (2018), S. 10

¹⁶ Vgl. Lucht (2019), S. 40 ff

¹⁷ Vgl. Lucht (2019), S. 45

¹⁸ Vgl. Deutsches Institut für Normung (2009) S. 158

¹⁹ Vgl. Fiedler (2016), S. 6

²⁰ Vgl. Trepper (2012), S. 28

reiche Planungs- und Dokumentationsaufgaben erledigen.²¹ Dem Team stehen situationsabhängige Unterstützungsinstrumente für unterschiedliche Aufgabenfelder bereit. Dazu zählen Kreativitätstechniken wie Brainstorming oder weitere Instrumente wie der Projektstrukturplan, das Balkendiagramm und die Netzplantechnik.²²

Der Projektverlauf des klassischen Projektmanagements basiert auf einem abzuarbeitenden Phasenmodell. Die Art der Phaseneinteilung ist von dem genutzten Vorgehensmodell abhängig.²³ Die einfachste Form eines klassischen Vorgehensmodells ist das Wasserfallmodell. Dort werden verschiedene Phasen definiert (bspw. Analyse, Entwicklung, Design, Test, Lieferung) die nacheinander stattfinden und in denen bestimmte Aktivitäten erfolgen. Feedback wird nur zwischen den jeweils aufeinanderfolgenden Phasen ermöglicht.²⁴

2.2 Grenzen des klassischen Projektmanagements

Im Folgenden wird die Problematik dargestellt, warum das klassische Projektmanagement bei einigen Arten von Projekten an seine Grenzen stößt.

Alle Phasenmodelle unterliegen einem Lebenszyklus. Dieser ist nicht nur branchenspezifisch, sondern auch abhängig von der Dynamik der Anforderungen und der Umwelt. Das PMI[®] unterscheidet in ihrem „Agile Practice Guide“ vier Projektlebenszyklen, den prognostizierten, iterativen, inkrementellen und den agilen Lebenszyklus.²⁵ Bei einem vorhersehbaren, prognostizierten Projektlebenszyklus eignet sich das klassische Projektmanagement, da die stabilen Anforderungen eine Vorabplanung möglich machen. Bei einem agilen Lebenszyklus, der sowohl iterativ als auch inkrementell ist, stößt das klassische Projektmanagement jedoch an seine Grenzen. Die Projektlebenszyklen sind nun nicht mehr einmalig und zeitlich begrenzt, sondern können diese Phase repetitiv durchlaufen.²⁶ Bei dieser Art von Lebenszyklus, wird in der Literatur der Begriff Projekt durch den Begriff Produkt ersetzt. Im Vordergrund steht nicht mehr der Projektprozess, sondern die kontinuierliche Erweiterung und Verbesserung eines

²¹ Vgl. Kusay-Merkle (2018), S. 17

²² Vgl. Trepper (2012), S. 33

²³ Vgl. Fiedler (2016), S. 68+69

²⁴ Vgl. Trepper (2012), S. 30+31

²⁵ Vgl. Kusay-Merkle (2018) S. 23, zit. n. Project Management Institute (2017) S. 17

²⁶ Vgl. Kusay-Merkle (2018), S. 19

Produktes.²⁷ Dies ist nötig, um auf neue Anforderungen und Umweltirritationen in dynamischen und schnellen Märkten reagieren zu können. Lucht macht dies deutlich, indem er in seinem Sachbuch „Theorie und Management komplexer Projekte“ den Projektvorgang „[...] als Problem und die Projektphase als mehrere Probleme umfassendes „Meta-Problem“ [...]“ bezeichnet.²⁸ Mit den iterativen Vorgehensmodellen wie im agilen Projektmanagement angewendet, kann der Projektvorgang so oft wiederholt werden, bis eine akzeptable Lösung für das Problem erreicht wurde.²⁹

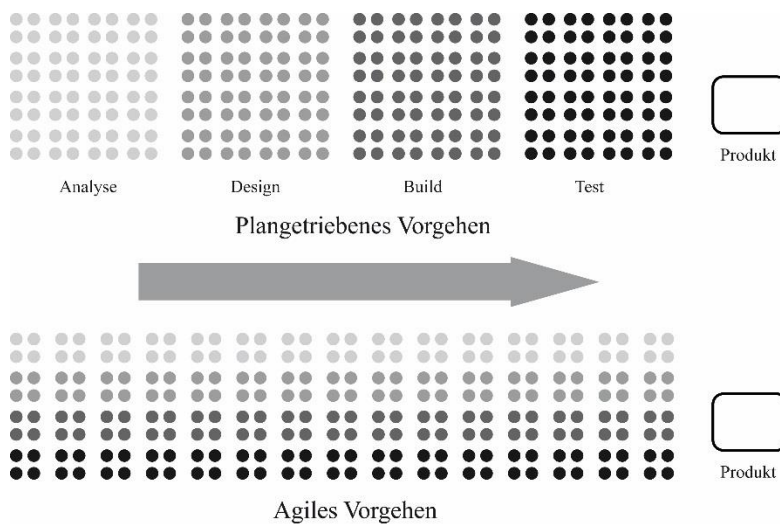


Abb. 1.: Darstellung der Phasen bei plangetriebenem und agilem Vorgehen³⁰

Ein weiterer großer Unterschied in den Projektmanagementansätze liegt neben dem unterschiedlichen Projektlebenszyklus, in der Zieldefinition. Der Erfolg eines traditionellen Projektes wird anhand des magischen Projektdreiecks bewertet. Es besteht aus drei sich gegenseitig beeinflussenden Zielen: dem Ziel der termintreue, dem Kostenziel, bestehend aus dem vorgegebenen Budget und dem Sachziel aus gewünschter Leistung und Qualität. Ein erfolgreiches Projekt erreicht alle drei Teilziele. Es ist daher für die Steuerung eines Projektes vorteilhaft von vornherein die genauen Zielpräferenzen zu kennen.³¹ Die Projektziele von Projekten mit agilen Projektlebenszyklen, lassen sich schwer anhand des magischen Projektdreiecks festlegen. Die Anforderungen müssen mit Hilfe der Iterationen erst entdeckt werden. Das Endergebnis, der Umfang des

²⁷ Vgl. Kusay-Merkle (2018), S. 28

²⁸ Lucht (2019), S. 239

²⁹ Vgl. Lucht (2019), S. 239

³⁰ Vgl. Kusay-Merkle (2018), S. 49 in Anlehnung an Griffiths (2015)

³¹ Vgl. Fiedler (2016), S. 6

Projektes, das Budget und der Zeitplan der Fertigstellung können nicht exakt definiert werden. Eine genaue Vorabplanung ist nicht möglich.³²

Zusammenfassend lässt sich sagen, dass Produkte mit einem agilen Projektlebenszyklus eine neue Art des Projektmanagement benötigen, da die Projekte nicht mehr zeitlich begrenzte, einmalige und planbare Vorhaben sind und sich die Projektziele vorab schwer definieren lassen.

2.3 Definition von „Agil“

Der deutsche Duden definiert agil als „von großer Beweglichkeit zeugend; regsam und wendig“. Als Synonyme werden beweglich und dynamisch vorgeschlagen.³³ Nachfolgend ist die Definition des Begriffs Agilität mit Hilfe von Literatur beschrieben.

Bleek und Wolk definieren den Begriff agil, als körperliche oder geistige Wendigkeit und Flinkheit. Auf die Softwareentwicklung angewandt, bedeutet dies, dass ein flexibler Prozess, schnell vorzeigbare Ergebnisse liefert.³⁴

Larman und Vodde übersetzen agil in ihrem Sachbuch als flexibel.³⁵ Sie bestimmen Agilität als „[...] the ability to move with quick easy grace, to be nimble and adaptable. To embrace change and become master of change – to compete through adaptability by being able to change faster than your competition can. This ability is supported by both lean and agile practices.“³⁶ Agile Methoden können zwar die Entwicklungszeit verkürzen und die Qualität verbessern, jedoch kann Agilität nicht von einem isolierten Entwicklungsteam erreicht werden. Larman und Vodde gehen einen Schritt weiter, für sie muss Agilität ein Teil der Unternehmenskultur werden und wird als Herausforderung an die Neugestaltung des organisationalen Systems verstanden.³⁷

So bedeutet „Agil“ in diesem Zusammenhang zügig auf veränderte Rahmenbedingungen zu reagieren und trotzdem schnelle Ergebnisse mit einer guten Qualität liefern zu können.

³² Vgl. Kusay-Merkle (2018), S. 49

³³ Bibliographisches Institut: Duden (online)

³⁴ Vgl. Bleek/Wolf (2008) S. 7

³⁵ Vgl. Larman/Vodde (2009), S. 140

³⁶ Larman/Vodde (2009), S. 140

³⁷ Vgl. Larman/Vodde (2009), S. 140

2.4 Agiles Projektmanagement in der Praxis

In der Praxis zeichnet sich die agile Projektumsetzung durch wiederholende, zeitgebundene Arbeitsschritte aus, die ein funktionierendes, nutzbares Produkt als Endergebnis anstrebt. Das Produkt beinhaltet naturgemäß für den Kunden wichtige und wertvolle Eigenschaften. Außerdem ist der Entwicklungsschwerpunkt auf eine qualitativ hochwertige Arbeit fokussiert.

Eine weitere grundsätzliche Eigenschaft von agilem Projektmanagement liegt in der Neugestaltung der Rollen von Teammitgliedern und Projektmanagern. Die Verantwortung und Aufgaben werden auf das Team verteilt.

Da es umfangreichen Feedbackschleifen und regelmäßigen Produktauslieferungen gibt, dürfen sich die Kundenwünsche und Anforderungen im Laufe des Projektes ändern. Als Resultat wird die Vorausplanung und Vorarbeit reduziert.³⁸

2.4.1 Vor- und Nachteile

Wolf und Meyer sehen die kurzen Iterationen als Vorteil gegenüber dem klassischen Projektmanagement.^{39,40} Gloger nennt außerdem, dass die Abhängigkeiten der Teams untereinander zum größten Teil aufgelöst werden.⁴¹ Beides führt zu kürzeren Entwicklungszeiten und zu einem schnelleren Time-to-Market, was die Wettbewerbsfähigkeit und Effektivität erhöht und woraus letztendlich niedrigere Kosten entstehen.⁴² Zusätzlich erzeugt schnelleres Feedback eine geringere Fehlerquote, die letztendlich zu einer besseren Produktqualität führt.⁴³ Cross-funktionale, selbstorganisierende Teams können in einer agilen, fehlerverzeihenden Unternehmenskultur die Selbstreflexion fördern. Dadurch wird ein organisationales Lernen angestoßen, welches das unternehmensinterne Know-how aufbaut und eine kontinuierliche Verbesserung zur Folge hat.⁴⁴

Ein möglicher Nachteil ist, dass eine optimale Besetzung der agilen Rollen nicht immer konfliktfrei funktioniert. So ist die Ausrichtung auf den Kunden problematisch, falls

³⁸ Vgl. Meyer (2014), S. 3 f.

³⁹ Vgl. Roock/Wolf (2016), S. 6 ff.

⁴⁰ Vgl. Meyer (2014), S. 154

⁴¹ Vgl. Gloger (2017), S. 44

⁴² Vgl. Roock/Wolf (2016), S. 7

⁴³ Vgl. Gloger (2017), S. S. 41

⁴⁴ Vgl. Mathis (2018), S. 27 ff.

diese Rolle nicht zur Verfügung steht oder nicht adäquat besetzt werden kann.⁴⁵ Auch können die agilen Methoden bei den Projektakteuren auf Disziplinlosigkeit, wenig Akzeptanz oder die Angst vor Veränderung und Verantwortung treffen und somit zu sozialen Konflikten führen.⁴⁶ Durch die Reduzierung der allgemeinen Planung besteht die Gefahr, die Materie ungenügend zu verstehen. Außerdem wird die Nennung eines Festpreises stark eingeschränkt, was die Vertragsgestaltung mit dem Kunden beeinflussen kann.⁴⁷ Fehlende Dokumentation und Projektüberwachung kann zu mangelhafter Durchführung von komplexen oder sicherheitsrelevanten Projekten führen.⁴⁸

2.4.2 Anwendungsgebiete

Die agilen Methoden haben ihren Ursprung in der Softwareentwicklung.⁴⁹ Der Scrum Guide definiert jedoch seinen Anwendungsbereich nicht nur auf die Entwicklung von Software, sondern bezieht sich auch auf „[...] die Verwaltung von Organisationen und der Entwicklung von fast allem, was wir in unserem täglichen Leben als Einzelpersonen und als Gesellschaften verwenden.“⁵⁰

Erfolgsversprechend ist das Agile Projektmanagement dann, wenn Produkte in einzelne Eigenschaften differenziert und separat bearbeitet werden können und es sicher ist, dass nach der Entwicklung dieser einzelnen Komponenten die Endergebnisse wieder zu einheitlichen Produkten zusammenzuführen sind. Lucht kritisiert, dass solche teilbaren Produkte außerhalb von IT-Projekten eher selten existieren.⁵¹

Eine Umfrage der Hochschule Koblenz aus dem Jahr 2015 ermittelte, dass agile Methoden, die auf der Teamebene stattfinden, zwar zum größten Teil in der Software und IT-nahen Entwicklung wie SAP eingesetzt werden, jedoch jedes vierte Unternehmen agile Methoden auch in Projekten ohne IT-Bezug einsetzt.⁵² Nichtsdestotrotz hat sich der Einsatz skalierter agiler Framework wie LeSS und SAFe bis jetzt nur bei Softwareprojekten durchgesetzt, was die Use-Cases bestätigen.^{53,54} Gloger und Larman zeigen

⁴⁵ Vgl. Bleek/Wolf (2008) S. 177 f.

⁴⁶ Vgl. Bleek/Wolf (2008) S. 180 f

⁴⁷ Vgl. Opelt et al. (2012), S. 21 f.

⁴⁸ Vgl. Trepper (2012), S. 107

⁴⁹ Vgl. Rook/Wolf (2016), S. 4 f.

⁵⁰ Vgl. Schwaber/Sutherland (2017), S. 4 (online)

⁵¹ Vgl. Lucht (2019), S. 239

⁵² Vgl. Komus/Kuberg (2015) S. 11

⁵³ Vgl. Scaled Agile, Inc: Case Studies von SAFe (online)

⁵⁴ Vgl. The LeSS Company B.V.: Case Studies von LeSS (online)

auf, dass die agilen Vorgehensweisen auch im Marketing Anwendung finden können. So lassen sich beispielsweise Stage-Gate-Prozesse in der Produktentwicklung stark verkürzen, wenn mit Hilfe von Iterationen das Feedback vom Kunden beschleunigt wird. Als Ergebnis wird die Zeit zur Herbeiführung einer Entscheidung über Marktfähigkeit und technischer Umsetzbarkeit reduziert.^{55 56} Es lassen sich außerdem agile und klassische Methode miteinander kombinieren oder ergänzen, was als hybrider Ansatz bezeichnet wird. Auch können bei Teilprojekten klassische und agile Ansätze nebeneinander zum Einsatz kommen. Beispielsweise werden vermehrt agile Methoden genutzt, wenn das Projekt viele Schnittstellen zum Endverbraucher hat.⁵⁷

2.5 Historie

Inspiziert wurden die agilen Projektmanagementprozesse und -methoden in den 90er-Jahren von dem Artikel "The New Product Development Game" aus dem Harvard Business Review im Jahr 1986 von Hirotaka Takeuchi and Ikujiro Nonak.⁵⁸⁵⁹ Der Artikel thematisiert besonders innovative und schnelle Produktentwicklungen: PKW's bei Honda, Spiegelreflexkameras bei Canon, Kopierer bei Fuji-Xerox und bei Canon sowie Personalcomputer von NEC.⁶⁰ Der Artikel zeigt auf, wie die verschiedenen Produktentwicklungsphasen wie Forschung und Entwicklung, Design, Produktion und Qualitätssicherung, die normalerweise erst beginnen, wenn die vorherige Phase vollständig abgeschlossen ist, sich nun überlappen können und synchron bearbeitet werden.⁶¹ Die sich überschneidenden Teams, werden Scrum Teams genannt. Den Namen haben Takeuchi and Nonak in Anlehnung an das Rugbyspiel gewählt, da ein Rugbyteam als crossfunktionales Team eng miteinander kooperiert und sich gemeinsam über das Spielfeld bewegt. Der Name Scrum ist im Rugby ein angeordnetes Gedränge und ist eine Methode, um das Spiel nach einer Unterbrechung neu zu starten.⁶²

⁵⁵ Vgl. Larman/Vodde (2009), S. 258 ff.

⁵⁶ Vgl. Gloger (2017), S. 22 ff.

⁵⁷ Vgl. Kusay-Merkle (2018), S. 50

⁵⁸ Vgl. Sutherland (2011) (online)

⁵⁹ Vgl. Larman/Vodde (2009), S. 45

⁶⁰ Vgl. Roock/Wolf (2016), S. S.1

⁶¹ Vgl. Roock/Wolf (2016), S. S.2

⁶² Vgl. Roock/Wolf (2016), S. 3

Auch wenn Takeuchi and Nonak dies nicht explizit erwähnt haben, basieren alle beschriebenen Unternehmen auf Lean.⁶³ Ziel dieses Wertesystems ist eine planbare Organisation mit einer kontinuierlichen Verbesserung und einem Minimum an Verschwendung. Übertragen auf die agile Softwareentwicklung bedeutet dies, in einem komplexen Umfeld schnell Anpassungsfähig zu sein und das organisationale Lernen als einen wesentlichen Erfolgsfaktor zu sehen. Im Vordergrund beider Wertesysteme steht der eigenverantwortliche Mensch und ein transparentes und vertrauensvolles Arbeitsumfeld.⁶⁴ In der Literatur ist jedoch umstritten, in wieweit agiles und Lean Management gleichzusetzen ist. Manche Autoren bemängeln, dass die Stärken agilen Managements in der Kreativität und Variabilität der Prozesse und Produkte liegt, währenddessen Lean Management dies aus seinen Produktionssystemen entfernen will, bis alle Prozesse weitestgehend identisch sind.⁶⁵ Trotzdem kann Lean als Inspiration der agilen Managementsysteme angesehen werden und beinhaltet mittlerweile viele Methoden, die sich mit den agilen Vorgehensweisen überschneiden.⁶⁶

Das erste Scrum Paper wurde von Sutherland und Schwaber 1995 mit dem Namen „SCRUM Development Process“ auf der Object-Oriented Programming, Systems, Languages & Applications '95 (OOPSLA '95) in Austin veröffentlichten. Es ist die erste öffentliche Beschreibung von Scrum für die Softwareentwicklung. Kent Beck veröffentlichte im Jahr 2000 in seinem Buch „Extreme Programming Explained – Embrace Change“ sein Konzept einer agilen Softwareentwicklung – die Geburtsstunde von Extreme Programming.⁶⁷ Trotz der anfangs breiten Ablehnung von Extreme Programming, setzten mit der Zeit immer mehr Teams XP erfolgreich ein, was auch Scrum zu einer erhöhten Bekanntheit verhalf und eine Inspiration für weitere Ansätze wie beispielsweise Crystal oder Feature Driven Development darstellte, die sich ebenso an die Lösung der Anforderungen orientierten, anstatt einem formalisierten Projektvorgehen.⁶⁸ Im Jahr 2001 trafen sich die IT-Spezialisten dieser Ansätze in Snowbird, Utha, um ein Agiles Manifest mit vier Werten und zwölf gemeinsamen Prinzipien⁶⁹ [Siehe 2.6 und 2.7] zu definieren, dessen Ziel es ist, die Zusammenarbeit zwischen Entwickler und Kunden zu

⁶³ Vgl. Sutherland (2011) (online)

⁶⁴ Vgl. Mathis (2018), S. 7

⁶⁵ Vgl. Gloger (2017), S. 8 f.

⁶⁶ Vgl. Roock/Wolf (2016), S. 1 ff.

⁶⁷ Vgl. Roock/Wolf (2016), S. 4

⁶⁸ Crystal und Feature Driven Sachbuch finden

⁶⁹ Vgl. Beedle et al. (2001) (online)

vertiefen und zu verbessern. In dem Zusammenhang wurden die Ansätze als agil bezeichnet.⁷⁰

2002 wurde von Ken Schwaber und Mike Beedle das erste Buch veröffentlicht, das Scrum zum Thema hatte. Auch David J. Anderson formulierte 2007 anhand dem Lean Manufacturing des Toyota Produktionssystems Kanban als Projektmanagement-Methode.⁷¹ 2010 wurde von Schwaber und Sutherland der erste offizielle Scrum Guide herausgegeben, der seitdem kontinuierlich weiterentwickelt wird.^{72 73}

2.6 Das Agile Manifest

Das agile Manifest kann auch als kleinster gemeinsamer Nenner aller agilen Vorgehensmodelle beschrieben werden.⁷⁴ Die dem Manifest zugrunde liegenden Werte sagen folgendes aus:

„Wir erschließen bessere Wege, Software zu entwickeln, indem wir es selbst tun und anderen dabei helfen. Durch diese Tätigkeit haben wir diese Werte zu schätzen gelernt:

Individuen und Interaktionen mehr als Prozesse und Werkzeuge

Funktionierende Software mehr als umfassende Dokumentation

Zusammenarbeit mit dem Kunden mehr als Vertragsverhandlung

Reagieren auf Veränderung mehr als Befolgen eines Plans

Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden, schätzen wir die Werte auf der linken Seite höher ein.“^{75 76}

Es wird deutlich, dass der Fokus auf die Entwicklung funktionierender Software gelegt wird, bei einer möglichst hohen Befriedigung der Kundenbedürfnisse. Die Werte der rechten Seite sind immer noch willkommen und sollten nach Möglichkeit eingehalten werden, jedoch werden die Werte auf der linken Seite als wichtiger angesehen.⁷⁷

⁷⁰ Vgl. Roock/Wolf (2016), S. 5

⁷¹ Vgl. Anderson/Carmichael (2016), S. 1, (online)

⁷² Vgl. Schwaber/Sutherland (2017) (online)

⁷³ Vgl. Roock/Wolf (2016), S. 5

⁷⁴ Vgl. Hruschka et al. (2009) S. 2 f.

⁷⁵ Übersetzung aus Roock/Wolf (2016), S. 5

⁷⁶ Vgl. Beedle et al. (2001) (online) (Englische Originalfassung)

⁷⁷ Vgl. Trepper (2012), S. 67 ff.

2.7 Agile Prinzipien

Die neben dem agilen Manifest 2001 veröffentlichten zwölf offiziellen agilen Prinzipien eignen sich dazu ein Gefühl für die agile Denkweise zu bekommen:

1. Unsere höchste Priorität ist es, den Kunden durch frühe und kontinuierliche Auslieferung wertvoller Software zufrieden zu stellen.
2. Heiße Anforderungsänderungen selbst spät in der Entwicklung willkommen. Agile Prozesse nutzen Veränderungen zum Wettbewerbsvorteil des Kunden.
3. Liefere funktionierende Software regelmäßig innerhalb weniger Wochen oder Monate und bevorzuge dabei die kürzere Zeitspanne.
4. Fachexperten und Entwickler müssen während des Projektes täglich zusammenarbeiten.
5. Errichte Projekte rund um motivierte Individuen. Gib ihnen das Umfeld und die Unterstützung, die sie benötigen und vertraue darauf, dass sie die Aufgabe erledigen.
6. Die effizienteste und effektivste Methode, Informationen an und innerhalb eines Entwicklungsteams zu übermitteln, ist im Gespräch von Angesicht zu Angesicht.
7. Funktionierende Software ist das wichtigste Fortschrittsmaß.
8. Agile Prozesse fördern nachhaltige Entwicklung. Die Auftraggeber, Entwickler und Benutzer sollten ein gleichmäßiges Tempo auf unbegrenzte Zeit halten können.
9. Ständiges Augenmerk auf technische Exzellenz und gutes Design fördert Agilität.
10. Einfachheit – die Kunst, die Menge nicht getaner Arbeit zu maximieren – ist essentiell.
11. Die besten Architekturen, Anforderungen und Entwürfe entstehen durch selbstorganisierte Teams.
12. In regelmäßigen Abständen reflektiert das Team, wie es effektiver werden kann und passt sein Verhalten entsprechend an.⁷⁸

Laut Meyer handelt es sich bei den Prinzipien zum Teil um Praktiken und selbstverständliche Redewendung, die für ein tiefergehendes Verständnis der agilen Arbeits-

⁷⁸ Übersetzung aus Hanschke (2017), S. 8

weise wenig nützlich sind. Beispielsweise schlägt das fünfte Prinzip vor, Projekte, um motivierte Individuen aufzubauen. Dies sollte unzweifelhaft auch für das klassische Projektmanagement gelten.⁷⁹

2.8 Scrum

In diesem Kapitel wird die agile Methode Scrum vorgestellt. Sie ist für das tiefere Verständnis von LeSS und SAFe von großer Bedeutung. Viele Rollen, Ereignisse und Artefakte werden in gleicher oder etwas abgewandelter Form für die Frameworks weiterverwendet. Daneben haben Umfragen ergeben, dass um die 85 Prozent der Unternehmen eine durchgängige agile, hybride oder selektive Form von Scrum in der Praxis anwenden und Scrum damit die größte Verbreitung erfahren hat. Somit ist Scrum bei der Anwendung von einer agilen Arbeitsweise unerlässlich.⁸⁰⁸¹ Für ein besseres Verständnis werden dazu erst die Begrifflichkeiten erklärt und anschließend der Ablauf in der Praxis dargestellt.

2.8.1 Grundlagen und Begrifflichkeiten von Scrum

Rollen

Die Mitarbeiter während des Projektes müssen drei unterschiedliche Rollen einnehmen.

Der **Product Owner** tritt als Schnittstelle zu den Kunden und Anwendern auf und ist für den Productbacklog rechenschaftspflichtig.⁸² Seine Hauptaufgabe liegt laut dem Scrum Guide von Schwaber und Sutherland darin „[...] den Wert des Produktes zu maximieren, das aus der Arbeit des Entwicklungsteams entsteht.“⁸³ Dies geschieht, in dem er die einzige Person ist, die das Product-Backlog betreut und dort eine Priorisierung des Produktnutzen durchführt.⁸⁴ Er muss dafür über das Wissen und die Talente verfügen, die Produktausprägungen auszuwählen, die für die internen und externen Stakeholder die größte Wertschöpfungstiefe aufweisen.⁸⁵ Ein Verständnis über die Kun-

⁷⁹ Vgl. Meyer (2014), S. 50 f.

⁸⁰ Vgl. Komus/Kuberg (2017) S. 4

⁸¹ Vgl. Zillmann (2019), S. 21

⁸² Vgl. Werro (2018), S. 10

⁸³ Vgl. Schwaber/Sutherland (2017), S. 6, (online)

⁸⁴ Vgl. Roock/Wolf (2016), S. 43

⁸⁵ Vgl. Werro (2018), S. S. 11

denbedürfnisse, den Technologien und dem Businessmodell ist erforderlich.⁸⁶ Dies ist nötig, da laut Wolf und Roock „[...] der Wertbeitrag der Funktionen in Software extrem unterschiedlich ist“. Sie zeigen auf, dass bei klassisch entwickelter Software, 64 % der erstellten Funktionen selten oder nie genutzt werden.⁸⁷

Der **Scrum Master** hat die Aufgabe, Scrum im Team einzuführen, indem er laut dem Scrum Guide „[...] die Scrum-Theorie, Praktiken, Regeln und Werte [...]“⁸⁸ vermittelt. Anschließend hilft er bei der Umsetzung von Scrum. Er coacht und unterstützt das Team und den Product Owner.⁸⁹ Außerdem verbessert er die Teamentwicklung und die Organisationsstruktur.⁹⁰ Zu seinen Aufgaben zählt es, die Scrum Meetings zu moderieren, zu coachen und das Team vor äußeren Einflüssen zu schützen,⁹¹ oder Hindernisse zu beseitigen, welche die Produktivität des Teams beeinträchtigen. Das Verteilen von Arbeitsanweisungen gehört nicht zu seinen Aufgaben. Dafür ist das Team zuständig.⁹² Für eine detaillierte Auflistung aller Scrum Master Aufgaben, bietet sich die Scrum Master Checkliste von Michael James an.⁹³ Der Scrum Master ist ein dienender Führer,⁹⁴ trotz Informationsvorsprung muss er einen kollegialen Umgangstil zum Team pflegen.⁹⁵

Das **Entwicklungsteam** besteht normalerweise aus drei bis neun Mitgliedern.⁹⁶ Goll und Hommel beschreiben die optimale Teamgröße in ihrem Buch „Mit Scrum zum gewünschten System“ mit drei bis sechs Mitgliedern. Größere Teams würden die Effektivität verringern.⁹⁷ Das Entwicklungsteam arbeitet autonom und selbstorientierend. Bei der Auswahl, der vom Product Owner zusammengestellten und priorisierten Backlog-Items hat, das Entwicklungsteam ein Mitspracherecht.⁹⁸ Die täglichen Aufgaben werden allein vom Team aus dem Sprint-Backlog ausgewählt.⁹⁹ Nach Wolf et. al. verspricht man sich von der Selbstorganisation „[...] bessere Entscheidungen, die die Qualität des

⁸⁶ Vgl. Roock/Wolf (2016), S. 45

⁸⁷ Vgl. Roock/Wolf (2016), S. 9, zit. n. Johnson (2002)

⁸⁸ Vgl. Schwaber/Sutherland (2017), S. 8, (online)

⁸⁹ Vgl. Hanser (2010), S. 66

⁹⁰ Vgl. James (2012), S. 4 ff. (online)

⁹¹ Vgl. Schwaber/Sutherland (2017), S. 8, (online)

⁹² Vgl. Hanser (2010), S. 66

⁹³ Vgl. James (2012) (online)

⁹⁴ Vgl. Schwaber/Sutherland (2017), S. 8, (online)

⁹⁵ Vgl. Hanser (2010), S. 66

⁹⁶ Vgl. Roock/Wolf (2016), S. 20

⁹⁷ Vgl. Goll/Hommel (2015), S. 90

⁹⁸ Vgl. Roock/Wolf (2016), S. S. 20

⁹⁹ Vgl. Meyer (2014), S. 20

Produktes auf Dauer gewährleisten“.¹⁰⁰ Das Team soll cross-functional zusammengestellt werden, um die Funktionalität der Produktauslieferung nach einem Sprint zu gewährleisten. Hierarchien und Titel wie Programmierer oder Designer sind nicht erwünscht. Die Teammitglieder sollen jede Aufgabe bearbeitet und sich falls nötig das Wissen aneignen oder ihre Kollegen anderweitig unterstützen.¹⁰¹

Scrum Ereignisse

Um häufige Produktauslieferungen und damit ein einhergehendes schnelles Feedback zu erreichen, sind die Arbeitsphasen im agilen Management interaktiv aufgebaut.¹⁰² In Scrum wird die Iteration, in der die Teamarbeit stattfindet **Sprint** genannt. Mit dem Namen soll eine kurze Projektphase assoziiert werden.¹⁰³ Auch wenn die Länge des Sprints nicht vorgegeben ist, geht die Literatur übereinstimmend von einer Dauer von einer Woche bis maximal 30 Tagen aus.¹⁰⁴¹⁰⁵¹⁰⁶ Sie ist abhängig von der Vorhersehbarkeit und Planbarkeit des Produktlebenszyklus und der Volatilität des Projektumfeldes. Unklare Ergebnisanforderungen erfordern kürzere Iterationen, um die Geschwindigkeit zu erhöhen Feedback zu erhalten und dadurch umfangreicher zu Lernen.¹⁰⁷

In Scrum existieren vier Arten von Meetings, die im Folgenden betrachtet werden.

Das **Sprint Planning** wird am Anfang eines Sprints durchgeführt. Die Hauptthemen dieses Meetings bestehen aus dem Inhalt des kommenden Sprints, der planerischen Umsetzung,¹⁰⁸ sowie der Definition eines Ziels.¹⁰⁹ Den Inhalt des Sprints wählt das Team zusammen mit dem Product Owner aus den priorisierten Backlog Items aus. Das Entwicklerteam entscheidet über die Anzahl der zu bearbeiteten Backlog Items, um eine Überbelastung zu vermeiden und die Qualität des Produktes zu gewährleisten.¹¹⁰ Anschließend wird die Umsetzung der einzelnen Backlog-Items geplant. Meyer empfiehlt dafür eine Liste mit dem vom Team akzeptierten Arbeitskriterien für jeden Back-

¹⁰⁰ Roock/Wolf (2016), S. S. 20

¹⁰¹ Vgl. Meyer (2014), S. 81

¹⁰² Vgl. Meyer (2014), S. 89

¹⁰³ Vgl. Roock/Wolf (2016), S. 25

¹⁰⁴ Vgl. Roock/Wolf (2016), S. 25

¹⁰⁵ Vgl. Meyer (2014), S. 89

¹⁰⁶ Vgl. Maximini (2018), S. 197 f.

¹⁰⁷ Vgl. Maximini (2018), S. 198

¹⁰⁸ Vgl. Roock/Wolf (2016), S. 22

¹⁰⁹ Vgl. Meyer (2014), S. 98

¹¹⁰ Vgl. Roock/Wolf (2016), S. 22

log-Eintrag.¹¹¹ Zum Schluss des Meetings wird das Sprint-Ziel festgelegt. Es soll die Produkterweiterung und die wertschöpfende Arbeit, die im kommenden Sprint geschaffen wird, für die Anspruchsgruppen beschreiben, um ein Verständnis für die zukünftige Arbeit zu schaffen.¹¹² Um unnötige Diskussionen zu vermeiden, schlägt Wolf eine Meetinglänge von zwei Stunden pro Sprintwoche vor.¹¹³ Meyer benennt eine Länge von einem Arbeitstag, oder acht Stunden.¹¹⁴

Das **Daily Scrum** ist ein kurzes, tägliches Meeting des Scrum Teams, das durch seinen kurzweiligen Charakter im Stehen abgehalten wird.¹¹⁵ Es erfolgt immer zur gleichen Zeit, meist morgens.¹¹⁶ Die Meetinglänge beträgt ungefähr 15 Minuten.¹¹⁷ Bei größeren oder räumlich verteilten Teams kann der Daily Scrum auch mehr Zeit in Anspruch nehmen. Eine genaue Zeitvorgabe soll jedoch die Effektivität erhöhen und eine Inflation der Meetingzeit vermeiden.¹¹⁸ Gegenstand des Meetings sind folgende drei Fragestellungen, die sich auf das Scrum-Ziel beziehen und an jedem Teammitglied gestellt werde: Was wurde am letzten Arbeitstag getan? Was soll heute getan werden? Gibt es Hindernisse?¹¹⁹ Meyer gibt zu bedenken, dass der Focus nicht darauf liegt tiefergehende Probleme zu lösen oder in technische Diskussionen abzudriften. Wenn es nicht möglich ist die Hindernisse in der Meeting-Zeit zu lösen, sollen die Probleme lediglich identifiziert und die Verantwortung zur Lösung an Teammitglieder mit der nötigen Expertise verteilt werden.¹²⁰

Das **Sprint Review** wird nach einem Sprint abgehalten. Es nehmen das Scrum-Team, der Product Owner und die Stakeholder teil.¹²¹ Die Anwendung der drei Prinzipien Transparenz, Inspektion und Adaption, des empirischen Managements helfen bei der kontinuierlichen Verbesserung des Produktes. Notwendig ist ein lieferbares Produktinkrement. Laut Wolf wird ein Produktinkrement transparent, wenn die Anspruchsgruppe es testen kann¹²² und sichtbar wird „[...] was im Produktinkrement enthalten ist

¹¹¹ Vgl. Meyer (2014), S. 98

¹¹² Vgl. Meyer (2014), S. 98

¹¹³ Vgl. Roock/Wolf (2016), S. 23

¹¹⁴ Vgl. Meyer (2014), S. 98

¹¹⁵ Vgl. Roock/Wolf (2016), S. 23

¹¹⁶ Vgl. Roock/Wolf (2016), S. 106 f.

¹¹⁷ Vgl. Roock/Wolf (2016), S. 23

¹¹⁸ Vgl. Meyer (2014), S. 92

¹¹⁹ Vgl. Roock/Wolf (2016), S. 23

¹²⁰ Vgl. Meyer (2014), S. 91 f.

¹²¹ Vgl. Maximini (2018), S. 199

¹²² Vgl. Roock/Wolf (2016), S. 71 f.

und was noch fehlt.“¹²³ Die Inspektion geschieht durch das anschließende Feedback der Nutzer und Anspruchsgruppen.¹²⁴ Es soll der Änderungsbedarf festgestellt werden, ein Abnahmemeeeting soll verhindert werden.¹²⁵ Anschließend wird das Product Backlog vom Product Owner anhand des Feedbacks einer Adaption unterzogen. Die Änderungen werden priorisiert integriert oder verworfen.¹²⁶

Die **Sprint Retrospektive** wird als offizieller Abschluss eines Sprints gesehen.¹²⁷ Bei dem Meeting nimmt neben dem Team der Scrum Master teil.¹²⁸ Das Meeting hat seinen Ursprung im organisationalen Lernen und Wissensmanagement.¹²⁹ Es wird auf der Prozessebene von Scrum als selbstoptimierende Feedback-Schleife angesehen, bei der Verbesserungsmöglichkeiten identifiziert werden, um die Abläufen und den Inhalt des nächsten Sprint zu optimieren und außerdem die Zusammenarbeit und die Produktqualität zu verbessern.¹³⁰ Damit ist die Retrospektive wichtiger Teil eines kontinuierlichen Verbesserungsprozesses, in dem nicht nur der Prozess, sondern auch das Lernen selbst zum Gegenstand des Lernens werden kann.¹³¹

Artefakte

Übergeordnete Artefakte beschreiben die Vision und die Ziele des Projektes. Bergsmann beschreibt die Vision in seinem Sachbuch „Requirements Engineering – für die agile Softwareentwicklung“ als „[...] kreatives Wunschbild eine sehr allgemeine, oft noch unkonkrete Darstellung dessen, was erreicht werden soll.“¹³² Als Ziel definiert er einen eindeutig festgelegten, beabsichtigten Zustand, der für die Stakeholder verständlich und messbar ist.¹³³ Während die Vision noch unkonkret und nicht immer messbar sein muss, hilft es die Ziele SMART zu formulieren.¹³⁴ Die Absicht von Vision und Ziel

¹²³ Roock/Wolf (2016), S. S. 72

¹²⁴ Vgl. Roock/Wolf (2016), S. 72 f.

¹²⁵ Vgl. Maximini (2018), S. 199

¹²⁶ Vgl. Roock/Wolf (2016), S. 73 f.

¹²⁷ Vgl. Maximini (2018), S. 199

¹²⁸ Vgl. Meyer (2014), S. 99

¹²⁹ Vgl. Roock/Wolf (2016), S. 120 ff.

¹³⁰ Vgl. Schwaber/Sutherland (2017), S. 14, (online)

¹³¹ Vgl. Meyer (2014), S. 99

¹³² Bergsmann (2015), S. 71

¹³³ Vgl. Bergsmann (2015), S. 71

¹³⁴ Vgl. Bergsmann (2015), S. 73 f.

besteht darin, „[...] die strategische Ausrichtung und die groben Leitlinien des zu entwickelnden Produkts [...] zu beschreiben.“¹³⁵

Die im Folgenden beschriebenen Scrum-Artefakte sind Dokumente, die den Teammitgliedern eine Möglichkeit zur Überprüfung, Anpassung und Dokumentation bei der Teamarbeit geben.¹³⁶

In dem **Produktbacklog** sind die Anforderungen und Funktionalitäten an das Produkt beschrieben. Diese werden vom Product Owner erfasst, dokumentiert und der Umsetzungsaufwand geschätzt.¹³⁷ Das Team kann den Product Owner dabei unterstützen,¹³⁸ indem alle gemeinsam ein Product Backlog Refinement Meeting abhalten und gemeinsam die Anforderungen priorisieren und verfeinern.¹³⁹ Die Anforderungen werden **Product Backlog Items** genannt. Um die Anforderungen zu erfassen, nutzen die agilen Methoden wie Scrum oder eXtreme Programming **User Stories** oder **Epics**. Dort wird die Nutzersicht in kurzer Geschichtsform wiedergegeben.¹⁴⁰ Epics erfassen die Anforderungen nur weitläufig und wenig spezifiziert,¹⁴¹ besitzen jedoch einen Geschäftswert und sind in größere Projekten bedeutsam, wenn mit dem Stakeholdern die Anforderungen besprochen werden. Sind die Epics verfeinert und die Anforderungen detaillierter erfasst, gliedern diese sich in User Storys auf, die in dem Backlog-Item erfasst und in einem Sprint umgesetzt werden können.¹⁴² Die User Storys und Epics wandeln sich im Verlauf des Projektes, aufgrund von veränderten Anforderungen oder Umweltbedingungen und werden anhand neuem Feedback ständig überarbeitet. Das Product Backlog muss infolgedessen im Zeitverlauf immer weiterentwickelt werden.¹⁴³ Bergsmann kritisiert in seinem Sachbuch, dass Epics und User Stories nur einige von vielen Spezifikationsarten sind und mit anderen Methoden, wie Personas¹⁴⁴ oder Use Cases¹⁴⁵, kombiniert werden

¹³⁵ Bergsmann (2015), S. 71

¹³⁶ Vgl. Schwaber/Sutherland (2017), S. 14, (online)

¹³⁷ Vgl. Hanser (2010), S. S. 73

¹³⁸ Vgl. Goll/Hommel (2015), S. 93

¹³⁹ Vgl. Roock/Wolf (2016), S. 75 ff.

¹⁴⁰ Vgl. Roock/Wolf (2016), S. 58

¹⁴¹ Vgl. Bergsmann (2015), S. 75

¹⁴² Vgl. Roock/Wolf (2016), S. 63

¹⁴³ Vgl. Schwaber/Sutherland (2017), S. 15, (online)

¹⁴⁴ Vgl. Bergsmann (2015), S. 86

¹⁴⁵ Vgl. Bergsmann (2015), S. 97

sollten, um konkrete Erkenntnisse aus den Kundenwünschen zu ziehen.¹⁴⁶ Im Product Backlog sind höher priorisierte Backlog Items detaillierter beschrieben.¹⁴⁷ Wenn ein Backlog Item soweit verfeinert wurde, dass er für den Sprint vom Entwicklerteam ausgewählt werden kann, wird dieser als bereit markiert. Er kann nun im Sprint Planning berücksichtigt und dem nächsten Sprint Backlog hinzugefügt werden.¹⁴⁸

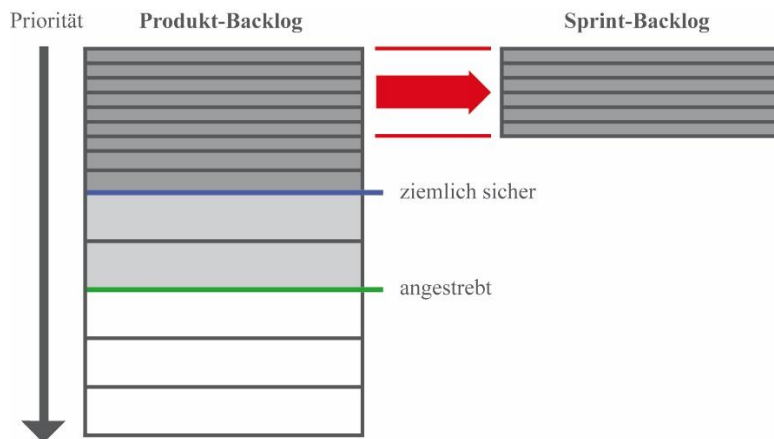


Abb. 2: Die Produkt Backlog Items werden in ein Sprint Backlog gezogen¹⁴⁹

Das **Sprint Backlog** beinhaltet die Produkt Backlog Items und den Aktivitätsplan, um die Items zu erschaffen und die Sprint-Ziele zu erreichen.¹⁵⁰ Nach Hanser ist es „[...] das Resultat der Sprint-Planungssitzung [Zusatz d. Verf.: Sprint Planning].“¹⁵¹ Es dokumentiert die in Zeitstunden geschätzten Aufgaben des Sprints¹⁵² und soll jederzeit für das Team eine transparente Übersicht über die verbleibende Arbeit geben. Das Sprint Backlog muss daher regelmäßig aktualisiert werden, um den aktuellen Status des Sprints sichtbar zu machen. Das Sprint Backlog wird nur von dem Entwicklerteam bearbeitet.¹⁵³

Das Ergebnis eines Sprints ist ein **Produktinkrement**. Es besteht aus den für den Sprint ausgewählten Produkt Backlog Items.¹⁵⁴ Das Produktinkrement soll auslieferbar sein und einen sichtbaren und getesteten Nutzen für den Anwender haben und eine

¹⁴⁶ Vgl. Bergsmann (2015), S. 19

¹⁴⁷ Vgl. Goll/Hommel (2015), S. 92

¹⁴⁸ Vgl. Schwaber/Sutherland (2017), S. 15 f., (online)

¹⁴⁹ Vgl. Hanschke (2017), S. 84

¹⁵⁰ Vgl. Goll/Hommel (2015), S. 94

¹⁵¹ Hanser (2010), S. 75

¹⁵² Vgl. Hanser (2010), S. 75

¹⁵³ Vgl. Schwaber/Sutherland (2017), 16 f., (online)

¹⁵⁴ Vgl. Goll/Hommel (2015), S. 96

Verbesserung zu den vorherigen Produktinkrement bieten, um den Wert des Produktes anzuheben. Frühzeitig Feedback kann mit einem lauffähigen Produktinkrement erhalten werden.¹⁵⁵ Das Produktinkrement muss den festgelegten Qualitätsanforderungen, den **Definition of Done** genügen. Dies sind zu Beginn eines Projektes, vom Team verfasste und beschlossene Qualitätsmerkmale, die ein Product Backlog Item als fertig definieren (Done).¹⁵⁶

2.8.2 Theoretischer Ablauf von Scrum

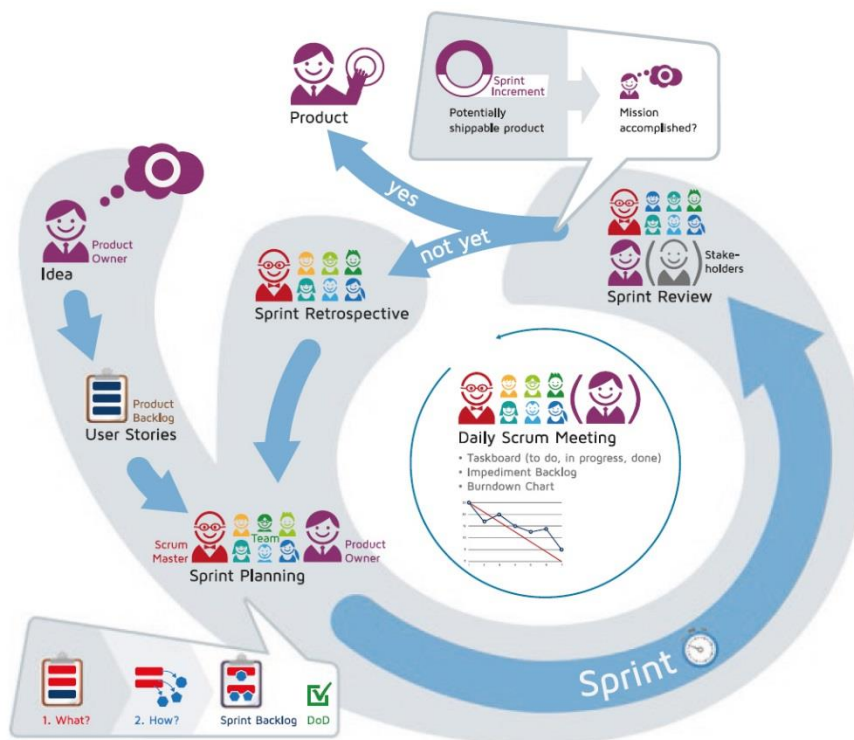


Abb. 3: Ablauf von Scrum¹⁵⁷

Der Grundstein für eine erfolgreiche Umsetzung in Scrum wird schon in der Projektvorbereitung gelegt. Auch wenn der Eindruck von Scrum nicht den Anschein macht, ist eine gute Vorplanung unausweichlich.¹⁵⁸ Haschke gibt den Rat „[...] Ihre agile Planung an der Erwartungshaltung Ihrer Stakeholder ausrichten.“¹⁵⁹ Es ist daher dringend notwendig, schon im Vorfeld die Stakeholder zu ermitteln und richtig einzuschätzen. Auch

¹⁵⁵ Vgl. Roock/Wolf (2016), S. 26 f.

¹⁵⁶ Vgl. Goll/Hommel (2015), S. 96

¹⁵⁷ mm1 Consulting & Management PartG (2016)

¹⁵⁸ Vgl. Hanschke (2017), S. 82

¹⁵⁹ Hanschke (2017), S. 82

wenn die Budget- und Terminplanung schwer kalkulierbar ist, verlangen viele zumindest ein grobes Budget und ein ungefährer Zeitpunkt des Projektendes.¹⁶⁰ Neben einer Risikoanalyse, sollte außerdem die Projektvision und die Projektziele definiert, sowie die multidisziplinären Mitarbeiter ausgewählt und ihre Rollen im Scrumteam verteilt werden.¹⁶¹ Die Scrum Artefakte müssen mit Informationen gefüllt werden. Das Team sollte eine übereinstimmende Auffassung von Scrum, der Vision und den Definition of Done festlegen. Nachfolgend müssen die Produkthanforderungen erfasst werden. Mittels Epics, User Storys oder anderen Spezifikationsarten wird das Product Backlog gefüllt.¹⁶² Um die Produkthanforderungen genauer zu erfassen, werden Stakeholder interviewt oder ein Gedankenaustausch initiiert, was zwangsläufig zu einem Anforderungsdokument führt. Auch wenn die agile Denkweise Dokumentation¹⁶³ und UpFront Requirements¹⁶⁴ kritisiert, ist dies im geeigneten Maß für die erfolgreiche Projektumsetzung unablässig. Meyer macht darauf aufmerksam, dass auch das Erfassen und Studieren von wahrscheinlich nicht umsetzbaren Anforderungen dabei hilft, Diskussionen anzustoßen und die richtigen Entscheidungen zu treffen und sie nicht im Sinne von „lean“ als „waste“ angesehen werden sollten.¹⁶⁵ Die vom Product Owner im Product Backlog erfassten Anforderungen werden priorisiert.¹⁶⁶ Dies geschieht anhand des Geschäftswertes des Produkt Backlog Items. Es gibt unterschiedliche Priorisierungsmöglichkeiten. Beispielsweise kann das Item mit dem höchsten Wert und dem niedrigsten Kosten zuerst erstellt werden. Auch der Risiko-Wert oder die Verzögerungskosten, die Kosten für die Zeit der Nicht-Verfügbarkeit eines Features, kann als Dimension hinzugezogen werden.¹⁶⁷ Nach der Priorisierung kann das komplette Scrum Team mit der Sprint Planung beginnen. Der Scrum Master moderiert das Meeting. Um den Umfang der einzelnen Produkt Items zu ermitteln und damit einzuschätzen wie viele Items in die Scrum Backlog übernommen werden können, eignen sich verschiedenen Planungstechniken, wie der Planning Poker oder Story Points. Hier schätzt jedes Teammitglied die Komplexität der Produkt Backlog Items.¹⁶⁸ Danach wird das Item

¹⁶⁰ Vgl. Hanschke (2017), S. 82

¹⁶¹ Vgl. Goll/Hommel (2015), S. 103 ff.

¹⁶² Vgl. Goll/Hommel (2015), S. 104 ff.

¹⁶³ Vgl. Bergsmann (2015), S. 45 ff.

¹⁶⁴ Vgl. Bergsmann (2015), S. 18 ff.

¹⁶⁵ Vgl. Meyer (2014), S. 32 ff.

¹⁶⁶ Vgl. Roock/Wolf (2016), S. 18

¹⁶⁷ Vgl. Roock/Wolf (2016), S. 52 ff.

¹⁶⁸ Vgl. Hanschke (2017), S. 37

schrittweise in den nächsten Sprint gezogen.¹⁶⁹ Nach dem Sprint Planning kann das Entwicklerteam die ausgewählten Items bearbeiten. Es soll sich dabei selbst organisierend und autonom arbeiten.¹⁷⁰ Solange der Sprint läuft, dürfen die ausgewählten Items nicht verändert werden. Meyer spricht hier von der „closed-window“ Rule. Erst im nächsten Sprint Planning können Änderungen vorgenommen werden.¹⁷¹ Der Scrum Master hat währenddessen die Aufgabe, eine funktionstüchtige Arbeitsumgebung zu schaffen, Hemmnisse aufzulösen und die Teammitglieder methodisch zu unterstützen.¹⁷² Das Daily Scrum findet täglich statt, ein ungefähr 15 minütiges Meeting, welches den Fortschritt und mögliche auftretende Probleme behandelt.¹⁷³ Am Ende des Sprints wird ein Produktinkrement ausgeliefert, das die Definition of Done abdeckt¹⁷⁴ und vom Product Owner einer umfangreichen Prüfung unterzogen wurde.¹⁷⁵ Im Sprint Review wird den Stakeholder das Inkrement mit seinen neuen Eigenschaften präsentiert, um an Hand des Feedbacks die Produkthanforderungen zu überarbeiten. Dies hat zur Folge, dass das Produktbacklog aktualisiert wird, aus dem die Produkt Items für den neuen Sprint gezogen werden.¹⁷⁶ Am Ende findet die Sprint Retrospektive statt. Ein Meeting, bei dem das Scrum Team versucht, den Managementprozess, die agilen Methoden und die Abläufe zu reflektieren und zu verbessern, damit die Zusammenarbeit im nächsten Sprint noch reibungsloser verläuft.¹⁷⁷ Der nächste Sprint wird ohne Pause angeschoben und fängt wieder mit einem Sprint Planning an.¹⁷⁸

2.9 Kanban

Kanban bedeutet übersetzt etwa visualisierte (KAN) Karte oder Tafel (BAN). Im japanischen Lean Produktionsansatz wird tatsächlich mit einer eingeschweißten Informationskarte gearbeitet, die an einem Teilebehälter angebracht ist und entweder die Entnahme oder die Produktion der benötigten Teile von Arbeitsstationen regelt. Die Karte wird bei der Nutzung der Teile an die vorgelagerten Arbeitsstationen weiter-

¹⁶⁹ Vgl. Roock/Wolf (2016), S. 95

¹⁷⁰ Vgl. Roock/Wolf (2016), S. 19

¹⁷¹ Vgl. Meyer (2014), S. 71 ff.

¹⁷² Vgl. Hanser (2010), S. 65 ff.

¹⁷³ Vgl. Hanser (2010), S. 71 ff.

¹⁷⁴ Vgl. Roock/Wolf (2016), S. 89 ff.

¹⁷⁵ Vgl. Hanser (2010), S. 69

¹⁷⁶ Vgl. Roock/Wolf (2016), S. 19

¹⁷⁷ Vgl. Hanser (2010), S. 140

¹⁷⁸ Vgl. Roock/Wolf (2016), S. 19

gegeben, damit die verbrauchten Teile wieder aufgefüllt werden. Somit gehen die Arbeitsaufträge zurück bis zum ersten Arbeitsschritt, in die Rohmaterialien bearbeitet oder bereitgestellt werden. Bei Kanban wird nach dem Pull-Prinzip mit Just-in-time-Verbindung gearbeitet. Durch die ausgelösten Arbeitsaufträge werden in Echtzeit nur die benötigten, angeforderten Teile erstellt. Eine Überproduktion und damit verbundene größere Lagerhaltung findet nicht statt. Bei den agilen Projektmethoden wird der Ansatz dazu genutzt, die Prozesse in ihren hintereinander ablaufenden Handlungen an einem Kanban-Board, mit Hilfe von Kanban Karten abzubilden und den Fortschritt der Arbeitsschritte darzustellen. Dabei durchlaufen die Arbeitsschritte die Prozess- oder Wertschöpfungskettenphasen meist von links nach rechts bis zur Spalte „DONE“.¹⁷⁹ Scrum und Kanban werden in der Praxis häufig zusammen genutzt, weil es sich hervorragend kombinieren lässt. Beispielsweise kann der Sprint Backlog mit der Kanban Methode verwaltet werden, um jederzeit einen transparenten Überblick über den Arbeitsfortschritt zu gewährleisten.¹⁸⁰

¹⁷⁹ Vgl. Hanschke (2017), S. 22 f.

¹⁸⁰ Vgl. Kusay-Merkle (2018), S. 232 ff.

3 Skalierbare agile Projektmanagementmethoden

Scrum und andere agile Methoden sind auf ein Team ausgelegt und auf kleinere Produktentwicklungen beschränkt.¹⁸¹ Um jedoch größere Entwicklungsvorhaben agil zu realisieren, müssen die agilen Teams parallel miteinander funktionieren. In der Fachliteratur wird von einem „skalierten“ Arbeitsumfeld, einem sogenannten Framework gesprochen, das benötigt wird, um mit neuen Prozessen und Methoden den aufkommenden Herausforderungen umgehen zu können.¹⁸²

Der deutsche Duden definiert „skalieren“ mit „die Größe eines Bildes o. Ä. unter Einhaltung der Proportionen ändern“.¹⁸³ Czudek schreibt in seinem Artikel „Ist SAFe das neue Scrum?“: „Agilität zu skalieren heisst [sic!], Praktiken und Methoden zu etablieren, die mehreren (agilen) Teams und den involvierten Stellen eine Zusammenarbeit ermöglicht, ohne die agilen Werte und Prinzipien zu kompromittieren.“¹⁸⁴ In der Literatur lassen sich mit dem Zitat übereinstimmende Meinungen finden. Wolf sieht einerseits die Praktiken und Methoden für die Strukturierung von mehreren Teams als Aufgabe der Skalierung an, andererseits sind auch die Verankerung der agilen Werte und Prinzipien, also die Verankerung der Unternehmenskultur als grundlegende Mission, bei der Skalierung von agilen Methoden zu sehen.¹⁸⁵

3.1 Large Scale Scrum LeSS

Large Scale Scrum (LeSS) wurde von Larman und Vodde 2008 publiziert.¹⁸⁶ Es ist kein Framework, dass Scrum, im Gegensatz zu SAFe, nur auf der Teamebene enthält, sondern mit vielen Teams arbeitet.¹⁸⁷ In anderen Worten ist LeSS keine eigenständige Methode, sondern besteht aus Scrum in einem größeren organisationalen Kontext.¹⁸⁸ Large-Scale Scrum unterscheidet zwei Frameworks, das LeSS in dem zwei bis acht

¹⁸¹ Vgl. Roock/Wolf (2016), S. 20 ff.

¹⁸² Vgl. Maximini (2018), S. 121

¹⁸³ Bibliographisches Institut GmbH: Duden (online)

¹⁸⁴ Vgl. Czudek (2015) S. 41

¹⁸⁵ Vgl. Roock/Wolf (2016), S. 164 f.

¹⁸⁶ Vgl. Mathis (2018), S. 208

¹⁸⁷ Vgl. Larman/Vodde (2017), S. 8

¹⁸⁸ Vgl. Larman/Vodde (2017), S. 13

Teams arbeiten können und LeSS Huge für mehr als acht Teams.¹⁸⁹ In LeSS Huge werden die Teams in Requirement Areas zu je vier bis acht Teams zusammengefasst, in denen wie in der kleineren LeSS-Implementierung gearbeitet wird, sich jedoch ein übergeordneter Product Owner auf die Umsetzung des Gesamtproduktes konzentriert.¹⁹⁰

3.1.1 Grundlagen und Begrifflichkeiten von LeSS

LeSS basiert auf den agilen Werten aus Kapitel 2.6, orientiert sich außerdem an den Prinzipien von Scrum und erweitert diese um eigene. Hervorzuheben wäre hier, dass LeSS ausdrücklich betont Scrum zu sein und daher so einfach wie möglich angewandt werden soll. Dementsprechend betont das Prinzip „More with less“, dass bewusst auf eine große Anzahl an neuen Prozessen, Artefakten und Rollen verzichtet wurde, damit die Teams weiterhin von den agilen Vorteilen profitieren.¹⁹¹ Um jedoch die Skalierung von Scrum sicherzustellen, müssen sich Elemente von Scrum in ihrer Anwendung ändern, sowie ein paar neue Rollen, Ereignisse und Artefakte hinzukommen. Diese werden im nachfolgenden Kapitel erläutert.

Rollen

Auch bei LeSS gibt es nur einen **Product Owner (PO)**, der das Product Backlog betreut und die Items priorisiert. Er ist dabei für bis zu zehn Teams zuständig.¹⁹² Natürlich werden die Backlog-Einträge und der Kontakt mit den Stakeholdern umfangreicher, er darf daher den ganzheitlichen Produktfokus nicht verlieren. Um eine Überlastung zu vermeiden, sollte der PO mehr Aufgaben an die Teams delegieren.¹⁹³ Beispielsweise kann auch das Verfeinern der Backlogeinträge von den Teams übernommen werden.¹⁹⁴ Außerdem muss der PO nur an dem ersten Sprint Planning, dem Sprint-Review, sowie der Gesamtretrospektive teilnehmen. Die zweiten Sprint Plannings halten die Teams untereinander ab.¹⁹⁵

¹⁸⁹ Vgl. Larman/Vodde (2017), S. 15

¹⁹⁰ Vgl. Larman/Vodde (2017), S. 41 ff.

¹⁹¹ Vgl. Larman/Vodde (2017), S. 12 ff.

¹⁹² Vgl. Mathis (2018), S. 209

¹⁹³ Vgl. Larman/Vodde (2017), S. 188

¹⁹⁴ Vgl. Larman/Vodde (2017), S. 196

¹⁹⁵ Vgl. Larman/Vodde (2017), S. 210

In Less Huge, bei mehr als acht¹⁹⁶ andere Quellen sprechen von zehn Teams,¹⁹⁷ wird die Rolle des Product Owner aufgeteilt. Es existieren nun ein LeSS Huge Product Owner (PO) und ein Area Product Owner (APO). Der APO ist wie der PO im kleineren LeSS für den Product Backlog seines Anforderungsbereiches (Area) zuständig und nimmt an den Scrum-Meetings seiner Teams teil. Die Aufgaben des LeSS Huge Product Owners ändern sich, er ist nur noch für bereichsübergreifende Anforderungen und organisatorische Aufgaben zuständig. Sein Aufgabenbereich umfasst darüber hinaus das Untersuchen von neuen Trends und Themen, das Schaffen oder Liquidieren neuer Areas, das Zuweisen von Entwicklungsteams und Rollen auf die Areas, sowie das Unterstützen der APOs. Er muss außerdem eng mit dem höheren Management zusammenarbeiten. Zeit an den LeSS-Meetings teilzunehmen, hat er nicht mehr.¹⁹⁸ Die APOs und der PO formen zusammen das Product Owner Team. Dort werden produktweite Priorisierungsentscheidungen getroffen und auf die Areas verteilt.¹⁹⁹ Der PO hat eine größere Autoritätsgewalt als der APO. Damit soll verhindert werden, dass jeder Area Product Owner seine eigenen Ziele verfolgt. Er soll die APOs dabei unterstützen, die gleiche Basis der Zielerreichung auszuhandeln. Notfalls muss er jedoch eingreifen können, um die Ausrichtung vorzugeben.²⁰⁰

Der **Scrum Master** ist in LeSS für ein bis drei Teams zuständig. In Folge dessen beinhaltet ein größeres Projekt mit mehr als drei Teams mehrere Scrum Master. Im Gegensatz zu Scrum, hat sich der Scrum Master in LeSS zu einer Vollzeitrolle entwickelt. Sein Aufgabenbereich ist größtenteils deckungsgleich mit dem von Scrum, jedoch verschiebt sich sein Aufgabenschwerpunkt im Laufe der Zeit. Während am Anfang eines Projektes noch das Schulen, Betreuen und Beraten des Product Owners und des Entwicklungsteams im Vordergrund steht, wechselt sein Schwerpunkt mit der Zeit auf das Verbessern der Teamentwicklung und der organisationalen Strukturen.²⁰¹ Die Reorganisation der Organisation bei vielen Teams ist sinnvoll, um die Organisationsstruktur soweit zu verändern, dass für die Scrum-Teams optimale Arbeitsbedingungen

¹⁹⁶ The LeSS Company B.V.: LeSS Huge (online)

¹⁹⁷ Vgl. Mathis (2018), S. 209

¹⁹⁸ Vgl. Larman/Vodde (2017), S. 212 ff.

¹⁹⁹ Vgl. Larman/Vodde (2009), S. 220 f.

²⁰⁰ Vgl. Gloger (2017), S. 114 ff.

²⁰¹ Vgl. Larman/Vodde (2017), S. 148

herrschen. Um die Organisation darauf auszurichten, werden die Scrum Master aus den verschiedenen Teams gemeinsam als „Change Agents“ aktiv.²⁰²

Ereignisse

Vor dem ersten Sprint Planning sollte ein Backlog Refinement Meeting mit allen Teammitgliedern stattfinden, um die grundsätzlichen Produkthanforderungen zu erklären und eine gemeinsame Wissensbasis aufzubauen.²⁰³

Das Sprint Planning besteht aus zwei aufeinander aufbauenden Meetings. Beim übergeordneten Sprint-Planning 1 bespricht der Product Owner oder in LeSS Huge, der Area Product Owner, mit den ernannten oder gewählten Vertretern der einzelnen Teams die Gesamtanforderungen des Produktes.²⁰⁴ Dabei wird entschieden, welche Backlogitems die einzelnen Teams umsetzen. Anschließend diskutieren im Sprint Planning 2 die Teams untereinander, wie sie die ausgewählten Items umsetzen und erstellen ihren Sprint Backlog.²⁰⁵ Sollten sich Backlog Einträge sehr ähneln, oder ist ein Team auf ein anderes angewiesen, können in diesem Fall mehrere Teams ein Multiteam-Sprint-Planning 2 durchführen. Es erhöht die Transparenz, verbessert die Zusammenarbeit und soll Abhängigkeiten auflösen.²⁰⁶

Die Sprint Plannings werden in jeder Requirement Area unabhängig durchgeführt. Damit jedoch der übergeordnete Produktfokus nicht verloren geht, nehmen die Area Product Owner an einem Product-Owner-Team-Meeting teil. Hier wird laut Vodde und Larman die gemeinsame Ausrichtung diskutiert, außerdem tauscht man sich über die spezielle Situation und die anstehenden Ziele aus.²⁰⁷

Artefakte

Das Product Backlog Refinement (PBR) wird normalerweise in der Mitte eines Sprints abgehalten.²⁰⁸ Es werden vier unterschiedliche PBR-Workshops unterschieden. Im Initiale PBR erstellen die Teams in Zusammenarbeit das erste Product Backlog. Beim Gesamt-PBR wird die Grundsätzliche Ausrichtung im Hinblick auf die gesamte

²⁰² Vgl. Roock/Wolf (2016), S. 111 f.

²⁰³ Vgl. Larman/Vodde (2017), S. 19

²⁰⁴ Vgl. Larman/Vodde (2017), S. 297 ff.

²⁰⁵ Vgl. Larman/Vodde (2017), S. 297 ff.

²⁰⁶ Vgl. Larman/Vodde (2017), S. 302

²⁰⁷ Vgl. Larman/Vodde (2017), S. 305

²⁰⁸ Vgl. The LeSS Company B.V.: Product Backlog Refinement (online)

Produktanforderung diskutiert und besprochen welches Team welche Anforderungen weiter verfeinert. Neben diesen beiden größeren teamübergreifenden PBRs gibt es in LeSS auch kleinere PBRs.

Bei ihnen werden in LeSS in der Regel nur die Teams in dem Workshop involviert sein, die den Backlog Eintrag wahrscheinlich bearbeiten. So wird das Einzelteam-PBR dann durchgeführt, wenn das entsprechende Team einen Bezug zu dem Eintrag hat. Es können jedoch auch mehrere Teams Einträge bearbeiten und verfeinern. Dieses Multiteam-PBR profitiert vom Know-How anderer Gruppen.²⁰⁹

Das Review und die Retrospektive erfolgt wie bei Scrum nach einem Sprint und unterscheidet sich nur anhand der Anzahl der involvierten Personen. Alle Teams nehmen mit den Stakeholdern nach einem Sprint an dem Produkt-Sprint-Review teil.

Die Sprint Retrospektive wird wie bei Scrum nur von den Teams allein durchgeführt, trotzdem werden anschließend umfassendere Probleme und Angelegenheiten in der Gesamtretrospektive behandelt. Dort nehmen neben dem Product Owner, Scrum Master und den Managern, die Stellvertreter der Teams teil, um organisationsstrukturelle Lösungen zu entwickeln.²¹⁰

Larman und Vodde geben explizit keine Vorgaben oder Regeln bezüglich des Reviews oder der Retrospektive vor. Die einzelnen Area Requirements können zwar die Scrum Vorgaben anwenden, sie kritisieren jedoch, dass größere Sprint-Reviews schwer durchführbar sind, wenn die Area Requirements zu unterschiedlich und komplex sind. Des Weiteren machen die verschiedenen Standorte der Areas ein gemeinsames Meeting problematisch.²¹¹

Um die Informationstransparenz über das Produkt, seiner Anforderungen und der Entwicklungsarbeit zu gewährleisten, wird in LeSS, wie in Scrum auch, nur ein Product Backlog verwendet.²¹² In LeSS Hufe gibt es ein übergeordnetes Product Backlog, jedoch wird darin nur jeweils ein Eintrag für ein Requirement verwaltet. Die Area Product Backlogs beinhalten für diesen Eintrag alle verfeinerten Anforderungen, welche die Teams benötigen, um damit zu arbeiten.²¹³

²⁰⁹ Vgl. Larman/Vodde (2017), S. 268 ff.

²¹⁰ Vgl. Larman/Vodde (2017), S. 339 ff.

²¹¹ Vgl. Larman/Vodde (2017), S. 351 f.

²¹² Vgl. Larman/Vodde (2017), S. S. 218

²¹³ Vgl. Larman/Vodde (2017), S. 236 ff.

3.1.2 Theoretischer Ablauf von LeSS

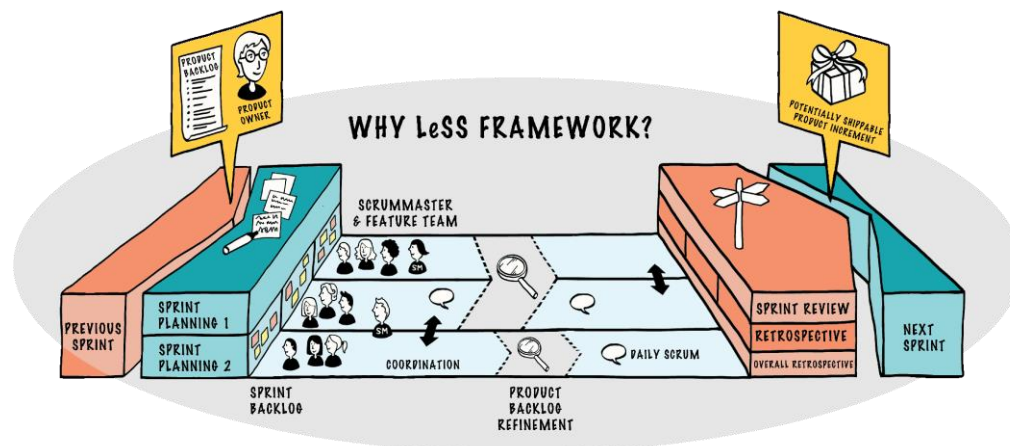


Abb. 4: Ablauf von LeSS²¹⁴

Im ersten gemeinsamen Sprint-Planning One wird das Produkt und die Anforderungen mit allen Teammitgliedern besprochen, um die Wissensbasis in allen Teams gleichmäßig zu gewährleisten. Bei mehr als zwei Teams²¹⁵ nehmen an den Sprint Plannings der nächsten Projektwochen nur noch die Stellvertreter teil. Diese können einmalig ernannt werden oder wöchentlich wechseln.²¹⁶ Im Sprint Planning One werden die vom Product Owner priorisierten Backlog Items von den Teams für den nächsten Sprint ausgewählt. Anschließend treffen die Teams sich zum Sprint Planning Two. Dort besprechen und planen sie abhängig von den Zusammenhängen der ausgewählten Backlog Items, entweder allein oder mit mehreren Teams zusammen, die Umsetzung im nächsten Sprint.²¹⁷ Der Sprint wird mit allen Teams gemeinsam durchgeführt.²¹⁸ Während der Sprints ist der Scrum Master für die reibungslose Kommunikation zwischen den Teams zuständig. Außerdem coacht, hilft und berät er die Teams und ist für die richtige Durchführung der Meetings von bis zu drei Teams zuständig.²¹⁹ In der Mitte des Sprints wird ein Product Backlog Refinement abgehalten. In diesem Workshop werden zusammen mit den Stakeholdern die Backlog-Einträge verfeinert und für das nächste Sprint-Planning vorbereitet. Zuerst wird mit den Stellvertretern der Gruppen ein

²¹⁴ Larman/Vodde (2017), S. 17

²¹⁵ Vgl. The LeSS Company B.V.: Sprint Planning One (online)

²¹⁶ Vgl. Larman/Vodde (2017), S. 19

²¹⁷ Vgl. Larman/Vodde (2017), S. 302

²¹⁸ Vgl. Larman/Vodde (2017), S. 17

²¹⁹ Vgl. The LeSS Company B.V.: Scrum Master (online)

Gesamt-Product Backlog Refinement abgehalten. Dort werden die Backlog-Einträge an die Gruppen verteilt. Abhängig ist dies davon, welche Gruppe den Eintrag im nächsten Sprint wahrscheinlich bearbeiten wird. Die Product Backlog Refinements können auch wieder gemeinsam mit anderen Gruppen durchgeführt werden, falls es für den Backlog Eintrag vorteilhaft ist.²²⁰ Nach einem Sprint wird wie in Scrum auch, mit allen Gruppen ein Sprint-Review mit den Stakeholdern abgehalten, das Gesamtkrement präsentiert und Feedback eingeholt. Dieses Feedback wird im Product Backlog festgehalten. Ein Teil des Sprint Reviews, kann wie eine Wissenschaftsmesse (Review-Basar) aufgebaut werden. Die einzelnen Teams besetzen Bereiche, in dem sie mit den Stakeholdern über die umgesetzten Backlog Einträge diskutieren.²²¹ Auch die Sprint Retrospektive wird nach dem Sprint durchgeführt. Erst tauscht sich jedes Team unter sich aus. Anschließend diskutieren alle Stellvertreter der Teams, der Product Owner, sowie übergeordnete Stellen in der Gesamtretrospektive die organisatorische Umsetzbarkeit über die Teamgrenzen hinweg. Der Wissenstransfer und die kontinuierliche Verbesserung soll damit auch auf organisationaler Ebene sichergestellt werden.²²² Anschließend wird der nächste Sprint mit einem Sprint-Planning 1 initiiert.

3.1.3 Theoretischer Ablauf von LeSS Huge

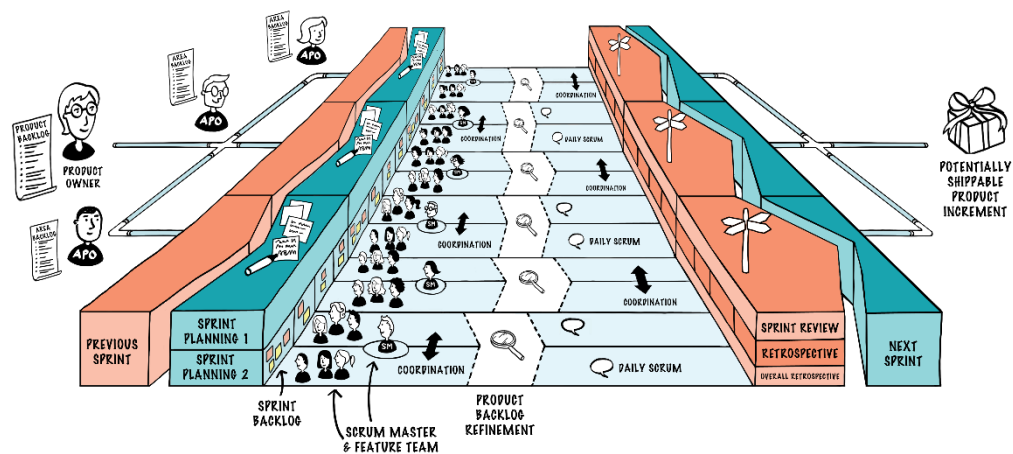


Abb. 5: Ablauf von LeSS Huge²²³

²²⁰ Vgl. Larman/Vodde (2017), S. 269 ff.

²²¹ Vgl. Larman/Vodde (2017), S. 342 f.

²²² Vgl. Larman/Vodde (2017), S. 344 ff.

²²³ Larman/Vodde (2017), S. 43

LeSS Huge wurde als Framework für mehr als acht Teams,²²⁴ beziehungsweise 10 Teams,²²⁵ entwickelt. Die größte Änderung zum kleineren Less-Framework ist das Schaffen von Requirement Areas. Diese sind nach wichtigen Kundenbedürfnissen ausgerichtet und decken damit einen größeren Produkthanforderungsbereich ab. Der Requirement Area besteht aus vier bis acht Teams. Da die Requirement Areas aus Produkthanforderungen bestehen, kann sich die Größe und Bedeutung der Bereiche, ebenso wie die Anforderungen auch, im Laufe eines Projektes ändern.²²⁶ In jeder Area wird der oben genannte LeSS-Ablauf vollzogen. Ein Unterschied besteht jedoch in der Rollenverteilung des Product Owners. Für ein Area ist jetzt der Area Product Owner zuständig, der für das Area Product Backlog verantwortlich ist. Für die gesamte Produktsicht ist ein übergeordneter Product Owner zuständig, der die Autoritätsgewalt über die Areas, sowie die Gesamtverantwortung innehat. Er verwaltet ein übergeordnetes Product Backlog. In dem nur die Produkthanforderung der Areas angegeben ist. Larman und Vodde geben als Areaanforderungen beispielsweise die Geschäftsabwicklung, die Markteinführung oder den Anlagenservice an.²²⁷ Die genaueren Produkthanforderungen der Areas für die Teams sind dann im Area Backlog beschrieben und werden, wie im kleineren LeSS auch, im Sprint Backlog verfeinert dargestellt und in den Sprints der jeweiligen Areas umgesetzt. Die von allen Areas umgesetzten Produkt-Einträge werden nach einem Sprint und den Stakeholder-Reviews zu einem gesamten Produktinkrement zusammengefügt.²²⁸

3.2 Scaled Agile Framework (SAFe)

Das zweite Modell neben LeSS ist das Scaled Agile Framework (SAFe). Es ist ebenso wie LeSS ein Framework, um agile Methoden zu skalieren und wurde von Dean Leffingwell 2011 publiziert.²²⁹ Es hat die Form einer Aufbauorganisation und ähnelt damit der Struktur vieler Unternehmen. Um die Skalierung auf verschiedenen organisationalen Ebenen zu vollziehen, beinhaltet SAFe einen Werkzeugkasten mit Methoden, Prozessen, Rollen, Artefakten und Meetings. SAFe basiert auf dem Agilen- und Lean-Wertesystem, das laut Mathis als Ziel hat „[...] eine effektive, planbare Orga-

²²⁴ The LeSS Company B.V.: LeSS Huge (online)

²²⁵ Vgl. Mathis (2018), S. 209

²²⁶ Vgl. Larman/Vodde (2017), S. 39 ff.

²²⁷ Vgl. Larman/Vodde (2017), S. 42

²²⁸ Vgl. Larman/Vodde (2017), S. 43 ff.

²²⁹ Vgl. Uludağ et al. (2017), S. 126

nisation mit kontinuierlicher Verbesserung und einem Minimum an Verschwendung [...]“²³⁰ in einer komplexen Umgebung, in der die Zukunft unsicher ist und das Lernen und Anpassen ein wesentlicher Erfolgsfaktor ist, zu errichten.²³¹

3.2.1 Grundlagen und Begrifflichkeiten von SAFe

In diesem Kapitel werden nur die wichtigsten Begrifflichkeiten von SAFe erklärt, da eine Beschreibung aller neuen Ideen, Ausdrücke und Begriffe den Rahmen der Arbeit sprengen würde.

SAFe basiert auf vier Ebenen, die immer dann realisiert werden, wenn ein Projekt über eine gewisse Größe hinauswächst.²³²

Die Teamebene

Auf der **Teamebene** findet die Entwicklungsarbeit mit bewährten agilen Vorgehensmodellen. Dies ist in der Regel Scrum, kann jedoch auch XP-Programmierung, Kanban oder ein ähnliches agiles Modell sein.²³³ Die Teams sind so groß wie in Scrum, cross-funktional und selbstorganisierend aufgebaut und beinhalten die in Scrum besetzten Rollen, wie Scrum Master und Product Owner.²³⁴ Die Teams arbeiten mit den typischen Vorgehensweisen, Meetings und Artefakten wie in Scrum definiert.²³⁵ Ziel der Teams ist das Entwickeln von User Storys, Features und Enabler. In SAFe sind Enabler auf allen Ebenen zu finden. Sie beinhalten die Anforderungen an die Systemarchitektur,²³⁶ die unausweichlich sind, wenn Features in das System implementiert werden, da neue Schnittstellen, Funktionen oder Technologien benötigt werden.²³⁷

Die Programmebene

Erst in der **Programmebene** unterscheidet sich die Struktur von SAFe grundlegend im Aufbau von LeSS und Scrum. Die agilen Teams der Teamebene sind auf der

²³⁰ Vgl. Mathis (2018), S. xiii

²³¹ Vgl. Mathis (2018), S. 7

²³² Vgl. Mathis (2018), S. 10 ff.

²³³ Vgl. Mathis (2018), S. 36

²³⁴ Vgl. Scaled Agile, Inc: Dev Team (online)

²³⁵ Vgl. Mathis (2018), S. 36

²³⁶ Vgl. Larman/Vodde (2017), S. 56

²³⁷ Vgl. Larman/Vodde (2017), S. 68 ff.

Programmebene in einem übergeordneten Team, dem sogenannten „Agile Release Train“ zusammengefasst.²³⁸ Mathis spricht in seinem Fachbuch von einem „Team der Teams“.²³⁹ Auf der Programmebene sind maximal 150 Personen involviert²⁴⁰, in der Regel 50 bis 125.²⁴¹ SAFe orientiert sich dabei an der „Dunbar-Zahl“, die von dem Kommunikationsexperten Robin Dunbar ermittelt wurde und aussagt, dass eine persönliche Kommunikation ab einer Größe von 150 Personen schwer aufrechtzuerhalten ist.²⁴² Die Programmebene beinhaltet neue Rollen und Artefakte, die im Folgenden weiter erläutert werden.²⁴³

Als **Agile Release Train** (ART) wird ein Team genannt, das aus den Teams der Teamebene besteht und durch gemeinsame Ziele eine übergeordnete Leistung schaffen soll.²⁴⁴ Diese Leistung äußert sich in einem fertigen Produktinkrement. Es ähnelt dem Requirement Area von LeSS, ist in seiner Struktur jedoch detaillierter aufgebaut, da eine Vielzahl neuer Rollen dazukommen.²⁴⁵

Im Agile Release Train werden folgende Rollen und Gremien besetzt:

Der **Release Train Engineer** nimmt die Rolle des Scrum Masters ein.²⁴⁶ Er unterstützt bei der Durchführung der Meetings, löst Hindernisse auf, hilft beim kontinuierlichen Verbesserungsprozess und coacht die Teams im Umgang mit den SAFe-Strukturen.²⁴⁷

Das **Produktmanagement**, bestehend aus einer Person oder einer Gruppe, erstellt und priorisiert das Backlog des Agilen Release Trains mit Hilfe der Stakeholder und gegebenenfalls der Teammitglieder und Product Owner aus der Team Ebene. Außerdem muss er die Vision und Ziele definieren. Auch wenn der Name anderes vermuten lässt, ist die Arbeit mit dem eines Product Owners von Scrum vergleichbar.²⁴⁸

²³⁸ Vgl. Scaled Agile, Inc: Agile Release Train (online)

²³⁹ Vgl. Mathis (2018), S. 10

²⁴⁰ Vgl. Mathis (2018), S. 36

²⁴¹ Vgl. Scaled Agile, Inc: Agile Release Train (online)

²⁴² Vgl. Mathis (2018), S. 3

²⁴³ Vgl. Mathis (2018), S. 42

²⁴⁴ Vgl. Mathis (2018), S. 41

²⁴⁵ Vgl. Larman/Vodde (2017), S. 236 ff.

²⁴⁶ Vgl. Mathis (2018), S. 42

²⁴⁷ Vgl. Mathis (2018), S. 83 f.

²⁴⁸ Vgl. Mathis (2018), S. 81

Die Systemarchitektur kann im Projektverlauf Probleme bereiten. Jedes Entwicklerteam erstellt in der Softwareentwicklung in der Regel sein eigenes Teilprogramm, das erst im späteren Projektverlauf mit anderer Applikation verbunden wird. Wenn die Entwicklerteams unkoordiniert ihre eigene Systemarchitektur schaffen, kann es Probleme bei der Verbindung oder den Abhängigkeiten geben.²⁴⁹ Um dem entgegenzuwirken, ist eine Rolle als **Systemarchitekten** besetzt. Er hilft den Entwicklerteams bezüglich der Systemarchitektur die richtigen Entscheidungen zu treffen, indem er im nötigen Maße vorausplant und als Coach und Berater zwischen den Teams vermittelt.²⁵⁰

Das **Releasemanagement** ist in der Regel eine Gruppe, der laut Mathis „der Release Train Engineer, der Systemarchitekt, das Produktmanagement und einige Stakeholder bzw. Entscheidungsträger, die nicht voll am Release Train mitarbeiten, wie Business Owner, Führungskräfte, Marketing, Qualitätssicherung und Vertrieb“, angehören.²⁵¹ Ihre Aufgabe besteht darin mit der nötigen Autorität, die Termine und den Umfang des Releases zu verabschieden und die gewünschte Qualitätsanforderungen in der Unternehmenspolitik zu verankern und zu verbreiten. Sie sind dafür zuständig die Auslieferung der Releases an die Stakeholder zu organisieren und stehen mit ihnen im ständigen Austausch bezüglich neuer Anforderungen oder Änderungen. Ihre cross-funktionale Zusammensetzung und die nötige Entscheidungskompetenz führen dazu, dass Probleme und Hindernisse schnell auf organisationaler Ebene beseitigt werden können, um einen reibungslosen Ablauf des Release Trains sicher zu stellen.²⁵²

Damit der Agile Release Train einen kontinuierlichen Wertestrom liefert, wird in SAFe eine **Lieferpipeline** beschrieben. Sie setzt sich aus vier Elementen zusammen, die jeweils Aktivitäten beschreiben.²⁵³ Die kontinuierliche Exploration hat zur Folge, dass die Markt und Benutzerbedürfnisse ständig überwacht werden, um die Vision und die Roadmap besser ableiten zu können und bei der Entdeckung neuer Anforderungen die Features sofort anpassen zu können.²⁵⁴ Das zweite Element besteht aus der kontinuierlichen Integration. Es beschreibt das ständige Prüfen und Testen der fertigen Software unter realistischen Systembedingungen. Somit können Fehler und Probleme schnellst-

²⁴⁹ Vgl. Gloger (2017), S. 38

²⁵⁰ Vgl. Mathis (2018), S. 85

²⁵¹ Mathis (2018), S. 83

²⁵² Vgl. Mathis (2018), S. 83

²⁵³ Vgl. Mathis (2018), S. 125 f.

²⁵⁴ Vgl. Mathis (2018), S. 126 ff.

möglich erkannt und darauf reagiert werden. Dies ist die Voraussetzung für das dritte Element – die kontinuierliche Auslieferung. Die kontrollierten Features können jederzeit erfolgreich in das Endsystem implementiert und danach in Echtzeit vom Endbenutzer in Betrieb genommen werden, um das letzte Element und das Ziel der Pipeline zu bedienen – die Produktlieferung nach Kundenbedarf.²⁵⁵

Das **Programminkrement** (PI) ist eine auf der Programmebene acht bis zwölf Wochen lang ablaufende Iteration. In dieser Iteration laufen wiederum die Sprints der Teams des Agile Release Trains, in denen die operative Arbeit vollzogen wird. Ziel ist das Entwickeln des Program Increment Objectives. Das PI Objective ist releasefähige Software, die sich aus den fertigen Features der umgesetzten Team-Sprints zusammensetzt.²⁵⁶

Der PI Iteration geht eine **PI Planung** mit allen beteiligten Mitarbeitern und Stakeholdern voraus. In der Programmebene ist es das wichtigste, aber auch umfangreichste Meeting. Über zwei Tage stehen viele unterschiedliche Themen auf der Agenda.²⁵⁷ Der Ablauf dieses Meetings ist in SAFe vorgegeben und im Internet einsehbar.²⁵⁸ Grundsätzlich dient das Meeting dazu, dass die zehn obersten ART-Backlog Features erklärt und anschließend von den Teams geplant, verfeinert und in Ziele abgeleitet werden. Die Ziele werden finalisiert und vom Management abgesegnet. Es werden außerdem die Risiken und Abhängigkeiten, die im Planungsprozess ersichtlich wurden, identifiziert, geprüft und Lösungen erarbeitet.

Etwas Besonderes ist die Vertrauensabstimmung am Ende des Meetings. Dort erfolgt eine Abstimmung, wie sicher die Teams die Zielerreichung einschätzen. Somit können schon im Vorfeld bei einer pessimistischen Einschätzung direkt in der Planungsphase neue Änderungen umgesetzt werden.²⁵⁹ Der große Vorteil dieses Meetings liegt in der Anwesenheit aller relevanten Stakeholder und ranghoher Führungspersonen. So können alle Mitarbeiter auf den gleichen Wissensstand gesetzt werden, wenn die Produkt- und die Architekturvision sowie der geschäftliche Kontext persönlich von den verantwortlichen Managern erläutert wird. Auch können so Probleme und Hindernisse direkt mit

²⁵⁵ Vgl. Mathis (2018), S. 126 ff.

²⁵⁶ Vgl. Scaled Agile, Inc: Program Increment (online)

²⁵⁷ Vgl. Mathis (2018), S. 60

²⁵⁸ Vgl. Scaled Agile, Inc: PI-Planning (online)

²⁵⁹ Vgl. Mathis (2018), S. 60 ff.

den zuständigen Stakeholdern abgesprochen und gelöst werden. Außerdem hat die PI Planung eine Vielzahl akzeptierter und abgestimmter SMART-Ziele, Release Termine und Meilensteine als Ausgangspunkt, auf die die Teams ihre Sprint-Ziele ausrichten können.²⁶⁰

Die Large Solution Ebene

Die **Large Solution Ebene** wird verwendet, wenn mehrere Agile Release Trains gleichzeitig an einem Produkt arbeiten. Das Produkt aus mehreren Trains kann sehr groß ausfallen, dementsprechend hat es einen größeren Wert für den Kunden, da es aus abgeschlossenen Programmen, Produkte, oder Servicetätigkeiten besteht, die in SAFe Solution genannt werden.²⁶¹

Die Meetings der Produktebene funktionieren anhand der von Dunbar definierten Kommunikations-Limits ab einer gewissen Personenanzahl nur eingeschränkt. Als Lösung werden die Agilen Release Trains in einem übergeordneten Team zusammengefasst, dem Solution Train, dieser gibt eine gemeinsame Vision und Mission vor und macht die Solution Ebene variabel für komplexe, sehr große Organisationen mit bis zu tausenden von beteiligten Organisationmitgliedern.

Ganz wie die Produkt- und Teamebene besitzt auch die Solution Ebene einen eigenen Solution Backlog und eine Iteration, die wiederum die beteiligten ARTs terminlich ausrichtet.²⁶²

Auch auf der Solution Ebene wird iterativ entwickelt und ausgeliefert.²⁶³ In dem Solution Backlog sind die Features aus der Produktebene jedoch zu Capabilities zusammengefasst. Mathis spricht hier vereinfacht von einer höheren Abstraktionsstufe, um die Eigenschaft des Stakeholderbedürfnisses zu beschreiben.²⁶⁴

Es gibt auch weitere Analogien zu den anderen Ebenen. So gibt es einen Solution Train Engineer, einen Solution Manager und einen Solution Architekten. Die Rollen haben die gleichen Aufgaben wie in den unteren Ebenen.²⁶⁵

Eine neue Rolle existiert im Shared Services. Dies sind Mitarbeiter, die über Spezialwissen verfügen und in bestimmten Agile Trains eingesetzt werden können.

²⁶⁰ Vgl. Mathis (2018), S. 64

²⁶¹ Vgl. Scaled Agile, Inc: Solution (online)

²⁶² Vgl. Mathis (2018), S. 91

²⁶³ Vgl. Mathis (2018), S. 93

²⁶⁴ Vgl. Mathis (2018), S. 92

²⁶⁵ Vgl. Mathis (2018), S. 93

Des Weiteren spielen Systemteams eine Rolle. Diese Teams werden bei train-übergreifenden Integrationsaufgaben gebildet, um die Zusammenarbeit zu erleichtern.²⁶⁶

Die Portfolio Ebene

Die Portfolio Ebene ist die höchste der SAFe Ebenen und bildet die Portfolios in einem Unternehmen ab. Die Portfolios beinhalten einen Wertstrom. Mathis definiert diesen als „[...] alle wertschöpfenden Aktivitäten, die notwendig sind, um ein Produkt herzustellen.“²⁶⁷ Im Wertstrom werden in einem kontinuierlichen Zeitablauf alle Werte für den Kunden geschaffen. Diese Werte werden aus unternehmerischen Impulsen gewonnen, die aus den strategischen Themen resultieren, die das Portfolio mit der Unternehmensstrategie verknüpfen.

Um die Werte zu definieren, werden diese als Epics beschrieben. Epics sind relativ große Vorhaben, können mehrere Solutions oder Release Trains betreffen und werden im Portfolio Backlog gespeichert und priorisiert, sowie in einem Portfolio Kanban visualisiert und gesteuert.²⁶⁸ Epics werden in der Regel genutzt, um neue Werte zu identifizieren und nach grundlegender Betrachtung und Wirtschaftlichkeitsprüfung auf die Capability und Features herunterzubrechen, die anschließend bei der Realisierung von den unteren Ebenen einer weiteren Analyse und Planung unterzogen werden.²⁶⁹

Betreut werden die Epics vom Epic Owner, der auch bei der Umsetzung in den Solution und Release Trains, die Verantwortung behält. Die Realisierung und Entwicklung der Epics obliegt dem Lean Portfolio Management. Es überwacht die Ausrichtung mit Hilfe der Investitionsentscheidungen und der Vergabe des Budgets für die Solution oder AR Trains, welches anschließend den Solution- oder den Produktmanagern zu Verfügung steht.²⁷⁰

²⁶⁶ Vgl. Mathis (2018), S. 95

²⁶⁷ Vgl. Mathis (2018), S. 115

²⁶⁸ Vgl. Mathis (2018), S. 107

²⁶⁹ Vgl. Scaled Agile, Inc: Features and Capabilities (online)

²⁷⁰ Vgl. Safe Buch Seite. 101

3.2.2 Theoretischer Ablauf von SAFe

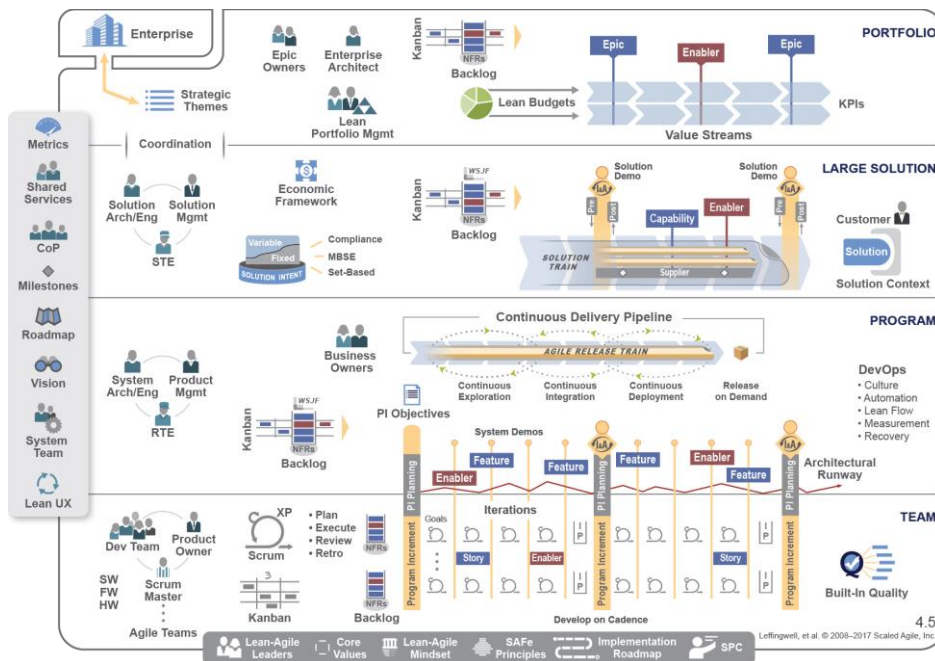


Abb. 6: Ablauf von SAFe

Der Ablauf von SAFe ist abhängig von der Größe des Projektes. Bei sehr umfangreichen Projekten können alle Ebenen in den Prozess mit einbezogen werden. Bei kleineren Projekten reicht in der Regel nur die Programm- und die Teamebene.²⁷¹ In diesem Kapitel wird SAFe, ausgehend von einem großen Projekt mit mehr als 150 Mitarbeitern, beschrieben.

Zuerst muss ein Unternehmen in ihren Portfolios die Wertströme identifizieren. Die Wertströme produzieren meist ein Produkt oder Service, den SAFe als Solution betitelt.²⁷² Anhand des Wertstroms können die Stakeholder identifiziert werden, für die eine Solution bestimmt ist oder die damit in Verbindung stehen. Jedem Portfolio wird vom Lean Portfolio Management eine übergeordnete Strategie und Vision hinzugefügt, dabei werden sie durch das Projekt- oder Programmbüro unterstützt.²⁷³ In Zusammenarbeit mit den Stakeholdern werden die ersten Epics und Epic-Enabler geschaffen und vom Epic Owner ausgearbeitet. Ein Epic Owner bearbeitet ein bis zwei Epics.²⁷⁴ Diese werden vom Lean Portfolio Management auf ihre Umsetzbarkeit und Wirtschaftlichkeit überprüft.²⁷⁵ Bei einer Genehmigung wird das Epic anschließend vom Epic Owner prio-

²⁷¹ Vgl. Mathis (2018), S. 40

²⁷² Vgl. Mathis (2018), S. 101

²⁷³ Vgl. Mathis (2018), S. 119

²⁷⁴ Vgl. Mathis (2018), S. 121

²⁷⁵ Vgl. Mathis (2018), S. 101

risiert und das Lean Portfolio Management weist ein Budget zu.²⁷⁶ Der Epic Owner bleibt auch nach Freigabe und Bearbeitung für das Epic verantwortlich und muss alle für die unteren Ebenen nötigen Informationen bereitstellen und mögliche Hindernisse beseitigen. Er arbeitet stark mit dem Produkt- oder Solutionmanagern zusammen, indem er diese unterstützt.²⁷⁷ Der Enterprise Architekt gibt die Richtlinien und Vorgaben der gemeinsamen Architektur vor.²⁷⁸

Umgesetzt werden die Epics in den Release Trains, sie bestehen aus Teams, die aus Teams bestehen. Bei einem Produkt, das mehr als 150 Personen einbindet, wird die Large Solution Ebene mit einbezogen. Die Epics werden in Capabilities heruntergebrochen und in den Solution Trains entwickelt. Die Solution Trains bestehen aus mehreren ARTs und besitzen eine eigene Iteration. Der zeitliche Umfang der Iteration kann je nach Solution mehrere PI Iterationen der ARTs umfassen.

Die ARTs sind auf der Programmebene vorhanden und bestehen aus den agilen Entwicklungsteams der Team Ebene. ARTs umfassen in der Regel 80 bis 125 Personen. In den ARTs übernehmen die Produktmanager und Product Owner der Programmebene die Verwaltung und Bearbeitung der Capabilities, die nun in Features verfeinert werden. Auch klassische Projektmanagementtechniken wie der Meilensteinplan und eine Roadmap werden angewendet.²⁷⁹ In den ARTs findet ebenso eine Iteration statt, die in der Regel acht bis zwölf Wochen dauert.²⁸⁰ Die sogenannte PI-Iteration hat das Ziel, Productinkremente zu schaffen. Diese werden durch Features definiert und auf der Teamebene von den einzelnen Teams in Iterationen umgesetzt. Anschließend findet auf den höheren Ebenen eine Zusammensetzung zu Product Inkrementen und später zu Solutions statt.

Auf jeder der vier Ebenen werden die Anforderungen in Backlogs definiert und verfeinert: Epics auf der Portfolio Ebene, Capabilities auf der Solution Ebene, Features im Backlog der Programm Ebene und zum Schluss in User Stories in den Sprint Backlogs der Teamebene. Das Ziel, ist das Identifizieren und Ausliefern immer größer werdender Produkthanforderungen, die bei ansteigender Größe den Wert für den Kunden erhöhen.

²⁷⁶ Vgl. Mathis (2018), S. 103

²⁷⁷ Vgl. Mathis (2018), S. S. 121

²⁷⁸ Vgl. Mathis (2018), S. S. 13

²⁷⁹ Vgl. Mathis (2018), S. 79

²⁸⁰ Vgl. Scaled Agile, Inc: Agile Release Train (online)

4 Auswirkung von LeSS und SAFe auf den Einsatz in Projekten unter Berücksichtigung ausgewählter Dimensionen

4.1 Auswahl der für die Analyse relevanten Dimensionen

Nachdem Scrum im Detail erklärt und die Frameworks dargestellt wurden, müssen für einen Vergleich der Skalierungsmodelle Dimensionen von Projekten bestimmt werden. Anhand der Dimensionen findet eine Betrachtung der Stärken und Schwächen der beiden Rahmenwerke statt. Die Auswahl, der für die Einordnung der Frameworks entscheidenden Dimensionen findet unter Zuhilfenahme von Fachliteratur statt. Es wurden drei relevante Dimensionen identifiziert, dessen Auswahl nachfolgend erläutert wird.

Madauss hat in seinem Sachbuch „Projektmanagement – Theorie und Praxis aus einer Hand“ zwölf Definitionsquellen zusammengetragen, die Projektmerkmale benennen. Das Ziel von Madauss war es, anhand der meisten Nennungen von Projektmerkmalen aus Definitionsquellen, die wesentlichen Projektmerkmale zu bestimmen. Drei Merkmale hatten mit Abstand die meisten Erwähnungen: „die Einmaligkeit, die zeitliche Begrenzung und die eindeutige Aufgabenstellung mit Verantwortung und Zielsetzung“ eines Projektes.²⁸¹ Bei einem agilen Projekt handelt es sich hauptsächlich um eine Produktentwicklung, bei der die zeitliche Begrenzung und die eindeutige Aufgabenstellung mit Verantwortung und Zielsetzung nicht gegeben sind. Daher konnte nur die Einmaligkeit als Projektmerkmal in Betracht gezogen werden. Die beiden anderen Merkmale kommen für agile Projekte nicht in Betracht und wurden nicht weiter berücksichtigt.

Bei den einmaligen Projekten lohnt sich jedoch eine genauere Betrachtung. Sie sind meist neuartig, innovativ und interpretationsbedürftig. Die Ursache-Wirkungs-Prinzipien sind schwer identifizierbar. Das Projekt wird als Komplex angesehen.²⁸² Daher ist es wichtig, dass ein agiles Framework mit steigender Komplexität umgehen kann. Auch der Fachartikel „agile@scale: Do more with LeSS or be SAFe? Ansätze zur

²⁸¹ Vgl. Madauss (2017), S. 52

²⁸² Vgl. Grimm (2009), S. 29

Skalierung - ein Überblick“ von Daut kommt zu der Erkenntnis, dass die Komplexität eines Projektes Einfluss auf die Auswahl des skalierenden Frameworks hat.²⁸³

Daut nennt außerdem die Größe der Entwicklungsorganisation als weiteres relevantes Projektmerkmal.²⁸⁴ Dies wird von dem Artikel „Agilen Erfolg skalieren - die Frameworks LeSS, Nexus und SAFe im Vergleich“ von Hildebrand bestätigt.²⁸⁵

Außerdem lässt Mathis erkennen, dass LeSS einen größeren Einfluss auf die agile Unternehmenskultur und Denkweise der Organisationsmitglieder hat, als andere Frameworks.²⁸⁶

Demzufolge wurden drei relevante Dimensionen zur genaueren Betrachtung herangezogen: Der Komplexitätsgrad, die Größe der Entwicklungsorganisation, sowie die Agilität der Unternehmenskultur. Die Dimensionen werden zuerst anhand von Fachliteratur definiert und anschließend einer Einordnung unterzogen.

4.2 Agilität der Unternehmenskultur

4.2.1 Theoretische Grundlage

Die Unternehmenskultur ist laut Herget der „[...] Lebensraum einer Organisation. Lebensraum umfasst dabei den Mikrokosmos, wie er von den Organisationsangehörigen erlebt und gelebt wird.“²⁸⁷ In diesem Lebensraum sind alle Organisationsmitglieder im aktiven Austausch untereinander und mit der Umwelt, dadurch bilden sich individuelle oder kollektive Denkweisen, die sich auf die Interpretations- und Verhaltensweisen der Organisationsmitglieder auswirken. Diese bestimmen das soziale Miteinander und haben einen großen Einfluss auf die Leistung der Organisationsmitglieder,²⁸⁸ indem nach Herget „[...] Ideen entstehen oder unterdrückt werden, Initiative befördert oder erstickt, Motivation gefördert oder vernichtet, Engagement belohnt oder entmutigt, Zusammenarbeit unterstützt oder behindert wird“.²⁸⁹

Es lohnt sich eine genauere Betrachtung, was die Unternehmenskultur ausmacht. Ein Modell, das die Unternehmenskultur beschreibt, wurde vom Organisationspsychologen

²⁸³ Vgl. Daut (2015) S. 164

²⁸⁴ Vgl. Daut (2015) S. 165

²⁸⁵ Vgl. Hildebrand, S. 2 f. (online)

²⁸⁶ Vgl. Mathis (2018), S. 209 f.

²⁸⁷ Herget (2018), S. 5

²⁸⁸ Vgl. Herget (2018), S. 5

²⁸⁹ Herget (2018), S. 5

Edgar Schein publiziert. Die Unternehmenskultur manifestiert sich laut Schein auf drei verschiedenen Ebenen: die Artefakt-Ebene, die Werte und Normen, sowie die Grundannahmen. Nur die oberste Ebene, die Artefakte, sind sichtbar. Bei ihr handelt es sich um Verhaltensweisen und Rituale, wie das Kommunikationsverhalten, die Kleidung im Unternehmen oder die Art und Weise wie sanktioniert oder belohnt wird.²⁹⁰

Werte und Normen bilden die zweite Ebene der Unternehmenskultur. Sie sind nicht sichtbar und haben sich im Laufe der Zeit durch positive Reaktionsmuster gebildet.²⁹¹ Es handelt sich um innere Muster, die der Person das Gefühl geben, das Richtige zu tun. In einer Organisation sind die Werte und Normen kollektiv verankert, und werden von vielen Organisationsmitgliedern gleichermaßen gelebt. Organisationsmitglieder, die davon abweichen, können sozial sanktioniert werden. Es beginnt eine Auslese, bei denen meist nur die Mitglieder mit den gleichen Werten und Normen in der Organisation verbleiben.²⁹²

Die Grundannahmen bilden die dritte, unterste Ebene ab. Sie sind individuell in jedem Menschen fest verankert und bestimmen seinen Blick auf die Umwelt und wie er damit interagiert. Sie können schwer verändert werden, weil sie unterbewusst wahrgenommen werden.²⁹³

Eine agile Unternehmenskultur äußert sich in einer ausgeprägten Fehlerkultur, Selbstreflektion, Verantwortungsbewusstsein, offene Kommunikation, eine transparente Entscheidungsfindung und natürlich indem die agilen Prinzipien und Werte gelebt werden. Außerdem ist die Motivation und Kompetenz bei den Organisationsmitgliedern vorhanden, die agilen Methoden anzuwenden und kontinuierlich zu verbessern.²⁹⁴ Für die Führungskräfte bedeutet dies, Hierarchien weitestgehend abzuflachen und die Verantwortung auf die Organisationsmitglieder zu verteilen, sowie die agilen Prinzipien und Werte vorzuleben.²⁹⁵ Führung verschiebt sich vom Leader-Follower-Verhältnis zum Diener oder Lehrer des Teams, indem Führungskräfte coachen und versuchen Probleme und Hindernisse zu lösen.²⁹⁶

²⁹⁰ Vgl. Schein (2018), S. 14

²⁹¹ Vgl. Sauter et al. (2018), S. 95 f.

²⁹² Vgl. Schein (2018), S. 15 ff.

²⁹³ Vgl. Schein (2018), S. 17 ff.

²⁹⁴ Vgl. Hanschke (2017), S. 102 ff.

²⁹⁵ Vgl. Maximini (2018), S. 20 f.

²⁹⁶ Vgl. Maximini (2018), S. 18

Für die Organisation bedeutet eine agile Kultur, das Lernen auf allen organisationalen Ebenen zu etablieren und zu verankern.²⁹⁷ Außerdem muss die Organisationsstruktur geändert werden. Die hierarchische Pyramidenform wird auf operativer Ebene meist durch eine zellulare Organisation abgelöst.²⁹⁸

In der Praxis äußert sich die von Schein publizierte sichtbare Artefakt-Ebene der Unternehmenskultur darin, dass beispielsweise vermehrt in Teamräumen gearbeitet wird, Krawattenzwang oder Uniformen wegfallen und transparent mit Informationen umgegangen wird, indem diese an den Wänden hängen. Die Teammeetings sind effektiv und die Mitarbeiter duzen sich untereinander.²⁹⁹

Die agile Kultur auf den unteren Ebenen ist auf den ersten Blick nicht immer sichtbar. Die agilen Werte und Normen könne jedoch erkannt werden. Beispielsweise wird Hierarchie und Autorität abgelehnt. Das klassische Projektmanagement wird kritisch betrachtet. Das kann soweit führen, dass Organisationsmitglieder, welche die agilen Regeln noch nicht kennen, abgelehnt werden. Belohnt wird in einer agilen Kultur Teamarbeit, Unterstützung und individuelle Entwicklung. Sanktioniert wird Egoismus, Wettbewerbsdenken und unsoziales Verhalten, welches die Gruppe schwächt.³⁰⁰

Die Dritte der Kultur-Ebenen, die Grundannahmen, ergeben sich in einer agilen Kultur daraus, dass jeder Mensch arbeiten möchte und eine intrinsische Motivation vorliegt. Dementsprechend sind eine transparente Strategie und eine eindeutige Vision, meist nützlicher als eine extrinsische Motivation. Die Umwelt wird als komplex und ein Plan meist als wenig nützlich angesehen. Unbekanntes wird mit Neugierde und Wissensdurst begegnet und mit Hilfe von Experimenten entdeckt.³⁰¹

4.2.2 Analyse der Frameworks hinsichtlich der Agilität der Unternehmenskultur

Nachdem die Unternehmenskultur definiert und seine agilen Ausprägungen beschrieben wurde, kann eine Einordnung der beiden Frameworks anhand der Unternehmenskultur vorgenommen werden.

²⁹⁷ Vgl. Hanschke (2017), S. 103 ff.

²⁹⁸ Vgl. Maximini (2018), S. 16

²⁹⁹ Vgl. Maximini (2018), S. 14 f.

³⁰⁰ Vgl. Maximini (2018), S. 16 f.

³⁰¹ Vgl. Maximini (2018), S. 18

Das Management hat einen großen Einfluss auf die oberste von Scheins Ebenen. Artefakte, wie die Kleidung, die Büroausstattung oder das Belohnungssystem lassen sich vom Management steuern. Die zweite Ebene, die Werte und Normen können nicht so leicht geändert werden. Zwar können nach Sauter et al. Hochglanzbroschüren mit Leitlinien, Visionen oder Verhaltenskodexe (Code of Conduct) im Unternehmen durchaus publiziert werden, trotzdem wird ein Wertewandel meist nicht vollzogen. Werte können sich nur im Zusammenhang mit der Umwelt auf allen organisationalen Ebenen bilden.³⁰² Doch wieso sind Werte und Normen für eine agile Arbeitsweise so wichtig? Neben Fach- und Sachwissen³⁰³ sind Werte und Normen Voraussetzung für den Kompetenzaufbau in einem Unternehmen. Sauter et al. schreibt dazu: „Erst Werte ermöglichen ein Handeln unter Unsicherheit, sie überbrücken oder ersetzen fehlendes Wissen, schließen die Lücke zwischen Wissen einerseits und dem Handeln andererseits.“³⁰⁴ Werte beeinflussen bewusst oder unbewusst aus vorherigen Bewertungsprozessen das richtige Handeln.³⁰⁵ Werte und Normen sind also neben Wissen und Qualifikationen die obligatorischen Bedingungen für Kompetenzen, die ein selbstorganisiertes, kreatives Handeln unter einem ungewissen Umfeld ermöglichen.³⁰⁶ Es wundert daher nicht, dass verschiedene Scrum Autoren wie Wolf,³⁰⁷ Maximini³⁰⁸ oder selbst die Gründer von LeSS, Larman und Vodde, davon abraten, agil zu skalieren, wenn Fähigkeiten und Werte nicht auf der kompletten zu skalierenden Organisationsebene vorhanden sind. Trotzdem gibt es bezüglich SAFe und LeSS Unterschiede. LeSS ist Scrum in einer großen Umgebung und wurde bewusst schlicht beschrieben, wie schon das LeSS-Prinzip „More with less“ (mehr durch weniger) deutlich macht. Larman und Vodde befürchten, mehr Rollen verringern die Verantwortung der Teams und mehr Artefakte entfernen die Teams von dem Kunden.³⁰⁹ Weniger Regeln bedeutet jedoch auch, dass Unsicherheiten auf skaliertem Umfeld entstehen können, wenn das genaue Fachwissen fehlt damit umzugehen. Um trotz Unberechenbarkeit richtig agil zu handeln, benötigt es tief verwurzelte agile Werte und Normen bei den Organisationsmitgliedern und die nötige Kompetenz, eigenständig neue agile Rollen, Meetings und Artefakte einzuführen und an

³⁰² Vgl. Sauter et al. (2018), S. 134 f.

³⁰³ Vgl. Sauter et al. (2018), S. 109

³⁰⁴ Sauter et al. (2018), S. 95

³⁰⁵ Vgl. Sauter et al. (2018), S. 96

³⁰⁶ Vgl. Sauter et al. (2018), S. 108

³⁰⁷ Vgl. Roock/Wolf (2016), S. 164

³⁰⁸ Vgl. Maximini (2018), S. 121

³⁰⁹ Vgl. Larman/Vodde (2017), S. 13 ff.

die Gegebenheiten anzupassen. Bei der Einführung von LeSS werden viele Entscheidungen offen gelassen oder auf die Optionalität der Wegweiser und Experimente verwiesen.³¹⁰ Die Teams sollen sich an die wenigen Regeln von LeSS halten und diese im Laufe der Zeit verbessern und an die eigenen Bedürfnisse anpassen.³¹¹ Dazu benötigen die Teams jedoch das Fachwissen von allen Skalierungsmodellen und agilen Techniken – was wiederum nur bei sehr agilen Unternehmen der Fall ist. Maximini gibt außerdem zu bedenken, dass schlecht umgesetztes Scrum, bei einer Skalierung mit vergrößert wird und dadurch zu vielen Problemen führen kann. Bevor skaliert wird, müssen die agilen Werte und Normen in allen Teams verinnerlicht sein und die agilen Methoden verstanden und gelebt werden.³¹² Die Einführung von LeSS kann als Bottom-up-Ansatz angesehen werden, da auf die mit Scrum arbeitende agile Teamebene eine schrittweise Vergrößerung mit immer mehr Teams erfolgt. Dementsprechend wird eine agile Organisationskultur schon vor der Skalierung benötigt.³¹³ Das Hilft auch um die Akzeptanz des Frameworks zu gewährleisten, da bei einer Einführung von LeSS große Teile der traditionellen Organisationsstruktur weichen müssen.³¹⁴ Eine agile Kultur ist daher für die Einführung von LeSS unausweichlich.

SAFe dagegen enthält viele Rollen und Artefakte aus dem klassischen Projektmanagement und setzt die agilen Methoden hauptsächlich auf der Teamebene um. Manche Autoren wie Maximini unterstellen der Methode sogar nicht agil zu sein, wenn sie komplett umgesetzt wird.³¹⁵ Hildebrand beschreibt SAFe etwas diplomatischer und schreibt, dass „[...] das Wesen agiler Softwareentwicklung, wie es das Manifest vorsieht, hier in den Hintergrund gerückt wird.“³¹⁶ Dementsprechend muss die Unternehmenskultur nicht vollständig agil sein, um SAFe umsetzen zu können. Mathis spricht sogar davon, dass größere, etablierte Unternehmen meist keine Erfahrung mit den agilen Techniken haben und die agile Unternehmenskultur noch nicht weit fortgeschritten ist. SAFe soll durch klare Regeln, Rollen und Artefakten bei der Umsetzung helfen und unerfahrene Unternehmen Anhaltspunkte bei der Umsetzung bieten, ohne zu viel

³¹⁰ Vgl. Mathis (2018), S. 208

³¹¹ Vgl. Larman/Vodde (2017), S. 10 ff.

³¹² Vgl. Maximini (2018), S. 121

³¹³ Vgl. Zillmann (2019), S. 21

³¹⁴ Vgl. Mathis (2018), S. 218 f.

³¹⁵ Vgl. Maximini (2018), S. 121 f.

³¹⁶ Hildebrand, S. 8 (online)

experimentieren zu müssen.³¹⁷ Mathis spricht bei SAFe von einem „[...] definierten Startpunkt für eine agile oder Lean-Transformation.“³¹⁸ Die Case Studies der Livestock Improvement Corporation,³¹⁹ EdgeVerve Systems Ltd³²⁰ und der Standard Bank³²¹ bestätigen Mathis. Viele Unternehmen versprechen sich eine Agilität der Unternehmenskultur bei der Einführung von LeSS. Somit eignet sich SAFe in Unternehmen mit einer wenig ausgeprägten agilen Unternehmenskultur. In sehr agilen Kulturen kann SAFe stattdessen auf Ablehnung stoßen. Viele Rollen werden in SAFe mit einer erhöhten Autorität³²² ausgestattet und die unterschiedlichen Ebenen sind einer hierarchischen Pyramidenform nicht unähnlich. Insofern kann es bei den Organisationsmitgliedern sehr agiler Unternehmen bei der Einführung von SAFe zu sozialen Spannungen oder Ablehnung kommen.

4.3 Größe der Entwicklungsorganisation

4.3.1 Theoretische Grundlage

Die Größe der Entwicklungsorganisation ergibt sich aus der Anzahl der beteiligten Teams. Da Scrum, XP Programming und andere agile Prozesse nur auf der Teamebene stattfinden, muss bereits bei zwei Teams eine Form der Skalierung stattfinden. Ganz nach dem angestrebten Produkt, können Projekte aus tausenden von Organisationsmitgliedern bestehen. Große Projekte umfassen eine Fülle von Arbeitspaketen, die gezwungenermaßen Abhängigkeiten unter den Teams verursachen. Das führt zu Schnittstellen der Teams untereinander, zu arbeitsteiligen Prozessen oder Verbindungen zum Projektumfeld, die einen erhöhten Abstimmungsbedarf zur Folge haben und dadurch mehr Koordinations- und Steuerungsfunktionen benötigen.³²³

³¹⁷ Vgl. Mathis (2018), S. 8

³¹⁸ Mathis (2018), S. 8

³¹⁹ Vgl. Scaled Agile, Inc: SAFe Case Study: Livestock Improvement Corporation (online)

³²⁰ Vgl. Scaled Agile, Inc: SAFe Case Study: Edgeverve Systems (online)

³²¹ Vgl. Scaled Agile, Inc: SAFe Case Study: Standard Bank (online)

³²² Vgl. Mathis (2018), S. 101 + 88 + 81

³²³ Vgl. Meyer/Reher (2016), S. 29 ff.

4.3.2 Analyse der Frameworks hinsichtlich der Größe der Entwicklungsorganisation

LeSS ist Scrum mit erweiterten Prinzipien und Regeln. Es ist einfach gestaltet und gerade nur für die wenige Scrum-Teams geeignet, die eine agile Kommunikationsfähigkeit besitzen und das agile Arbeiten gewohnt sind. Die Teams können sich in den Sprint Plannings absprechen, Abhängigkeiten koordinieren und aufkommende Hindernisse wie Architektur- und Systemschnittstellen lösen. Bei einer größeren Anzahl von synchron arbeitenden Teams kann der Koordinationsbedarf andererseits stark ansteigen. Larman und Vodde geben die Obergrenze bei acht Teams an,³²⁴ Mathis bei zehn Teams,³²⁵ die sie mittels empirischer Beobachtung als geeignet einstufen, von einem Product Owner betreut werden zu können. Bei mehr als acht Teams kann der Product Owner den Überblick über das Produkt verlieren und die Arbeit mit dem Product Backlog wird zu umfangreich für eine Person.³²⁶ LeSS Huge ist für mehr als acht Teams geeignet. Die Product Owner behalten jeweils ihre acht Teams und bilden damit eine Area. Die Areas werden von einem übergeordneten Product Owner verwaltet.³²⁷ Durch diese Struktur können mehrere Areas nebeneinander existieren und LeSS kann die Anzahl von acht Teams übersteigen.

Es lassen sich bei der Struktur von SAFe und LeSS Übereinstimmungen finden. Auch dort werden die Teams gemeinsam koordiniert. Die gemeinsamen Teams heißen Area Release Train in SAFe und Requirement Area bei LeSS. Während die Skalierung von LeSS zu LeSS Huge bei acht Teams stattfindet, orientiert sich SAFe an der von Dunbar ermittelte Obergrenze von 150 Personen.³²⁸ Ab dem Zeitpunkt werden mehrere ART-Trains in einem Solution Train zusammengefasst, um weiterhin Face-to-Face-Meetings, eine gemeinsame Koordination und Releases zu gewährleisten. Identisch ist bei SAFe und LeSS zudem das übergeordnete Product Backlog, das mit der steigenden Anzahl an Anforderungen umgehen soll, ohne den gemeinsamen Produktfokus zu verlieren.^{329,330}

³²⁴ Vgl. Larman/Vodde (2017), S. 15

³²⁵ Vgl. Mathis (2018), S. 209

³²⁶ Vgl. Larman/Vodde (2017), S. 15 f.

³²⁷ Vgl. Larman/Vodde (2017), S. 212 ff.

³²⁸ Vgl. Mathis (2018), S. 3

³²⁹ Vgl. Mathis (2018), S. 12

³³⁰ Vgl. Larman/Vodde (2017), S. 236 ff.

Bei LeSS sollen die Verantwortlichkeiten auf der Teamebene bestehen bleiben.³³¹ Dazu ist eine persönliche, dezentrale Selbstkoordination der Teams gewünscht.³³² Eine hierarchische Struktur, eine Ablauforganisation oder neue Rollen die eine zentralisierte Koordination ermöglichen, um mit der wachsenden Anzahl an Stakeholdern und Teams kommunizieren zu können, ist nicht vorhanden.³³³ Der Kommunikations-, Steuerungs- und Koordinationsbedarf wird jedoch weiter ansteigen, was von den Teams, Scrum Mastern und Product Owner allein nicht mehr bewerkstelligt werden kann.³³⁴

SAFe hingegen beinhaltet Ebenen. Diese ähneln einer Aufbauorganisation und umfassen eine Vielzahl unterschiedlicher Rollen mit definierten Verantwortungsbereichen. SAFe schafft es eine Vielzahl von internen und externen Stakeholder sowie die klassischen Unternehmensbereiche wie beispielsweise Marketing, Qualitätssicherung und Vertrieb mit einzubinden und so den Koordinations- und Steuerungsaufwand der Teams zu reduzieren.³³⁵

Zusammengefasst lässt sich sagen, dass die Struktur der Teams sich ähneln. Beide Frameworks fassen ihre Teams zusammen, um alle Anforderungen zu bündeln und eine gemeinsame Produktauslieferung zu gewährleisten. SAFe umfasst die Teams in einem Release Train und LeSS in einem Requirement Area. Danach findet eine weitere Erweiterung der Teams bei LeSS Hufe ab acht Teams, also maximal 80 Personen statt. Bei SAFe wird die Solution Ebene bei 150 Personen eingeführt.

Ab einer gewissen Personenobergrenze, können LeSS und LeSS Hufe durch ihre einfache Struktur den steigenden Anforderungen nicht standhaltend. Die wachsende Anzahl an Teams und Stakeholdern können den Kommunikations- und Koordinationsbedarf nicht mehr allein bewältigen. Hier fehlen die klaren Strukturen von SAFe, um Vorgaben und Erwartungen über viele Teams hinweg zu kommunizieren.

SAFe hingegen kann bei kleineren Projekten zu mächtig und schwerfällig erscheinen.³³⁶ In den SAFe Case Studien gibt es kein Projekt mit weniger als 100 involvierten Projektakteuren.³³⁷

³³¹ Vgl. Mathis (2018), S. 13 f.

³³² Vgl. Larman/Vodde (2017), S. 309 f.

³³³ Vgl. Mathis (2018), S. 13 f.

³³⁴ Vgl. Daut (2015) S. 8

³³⁵ Vgl. Mathis (2018), S. 83

³³⁶ Vgl. Daut (2015), S. 8

³³⁷ Vgl. Scaled Agile, Inc: Case Studies von SAFe (online)

4.4 Komplexitätsgrad eines Projektes

4.4.1 Theoretische Grundlage

Im folgenden Kapitel wird die Komplexität eines Projektes anhand von Fachliteratur beschrieben und Kriterien definiert, die mehrheitlich ein komplexes Projekt ausmachen. Malik beschreibt in seinem Sachbuch, dass bei komplexen Systemen die Funktionsweise unbekannt oder schwer zu verstehen ist und die Interpretation mehrere Lösungsvarianten zulässt, weil die Ursache-Wirkungs-Zusammenhänge nicht eindeutig sind.³³⁸

Nach Röpke führt „[...] Komplexität [...] dazu, dass die Akteure ein Projekt nicht vollständig übersehen und deshalb nicht in der Lage sind, die für die Zielerfüllung notwendigen Aktivitäten detailliert im Voraus zu bestimmen.“³³⁹

Als praxisbezogenes Beispiel kann die Softwareentwicklung herangezogen werden. Selbst Experten können die Fülle an unterschiedlichen Systemen nicht mehr vollends verstehen. Verschiedene Programmmodule und Codesstrukturen treffen auf Programmierfehler und Schnittstellenproblematiken, die wiederum auf eine Vielzahl an beteiligter Akteure treffen.³⁴⁰

In der Literatur sind viele Kriterien zu finden, die komplexe Projekte ausmachen. Diese sind sich größtenteils sehr ähnlich.³⁴¹ In dieser Arbeit übernehme ich die Definition von Kerzner / Belack und dem Project Management Institute (PMI).

Sie geben übereinstimmend eine große Anzahl schwer zu befriedigender Projektstakeholder an. So müssen viele interne und externe Akteure an dem Projekt teilhaben, was zu interpersonellen oder sozialen Konflikten führen kann. Zusätzlich ist eine kulturelle Vielfalt oder räumliche Distanz der Teams nicht ausgeschlossen.

Der Projektumfang ist riesig und hat erhebliche Kosten zu Folge. Die Projektmerkmale, Ressourcen oder Phasen können nicht klar definiert werden oder sind doppeldeutig. Das Projekt ist kompliziert und im Laufe der Zeit können sich die technologischen oder fachlichen Anforderungen wandeln oder neue dazu kommen.

Das Projekt unterliegt starken externen Markt-, Kunden- und Umwelteinflüssen, die auch von behördlicher oder staatlicher Seite reguliert werden können.³⁴²³⁴³

³³⁸ Vgl. Malik (2004), S. 36

³³⁹ Lucht (2019), S. 15, zit. n. Röpke (1977) S. 21

³⁴⁰ Vgl. Grimm (2009), S. S. 28 f.

³⁴¹ Vgl. Lucht (2019), S. 12

³⁴² Vgl. Lucht (2019), 12, zit. n. Kerzner/Belack (2010), S. 7

Der Artikel des PMI's 2013 Pulse of the Profession™ gibt an, dass das Budget für komplexe Projekte im Schnitt doppelt so hoch ist wie bei einfachen Projekten.³⁴⁴ Daraus ergibt sich ein erhöhtes Risiko, wenn bei gestiegenem Kapitaleinsatz ein unveränderten Erfolgswert gegenübersteht.³⁴⁵

Aus den Kriterien lassen sich Merkmale bilden, die ein komplexes Projekt beschreiben:

- Große Anzahl an Stakeholdern, starke externe Einflüsse und regulative Vorgaben
- Unklare, komplizierte Projektbestandteile und ein hoher Innovationsgrad des Produktes
- Hohes unternehmerisches Risiko

4.4.2 Analyse der Frameworks hinsichtlich des Komplexitätsgrads des Projektes

In dem vorherigen Kapitel wurden die Kriterien eines komplexen Projekts beschrieben. Im folgenden Kapitel wird untersucht, welches der beiden Frameworks besser mit den Kriterien verfahren kann.

Große Anzahl an Stakeholder, starke externe Einflüsse und regulative Vorgaben

Stakeholder sind Anspruchsgruppen, die Einfluss auf die Projektaktivitäten und die involvierten Personen oder das Unternehmen haben und organisationales Handeln auslösen können.³⁴⁶ Jedes Projekt verfolgt Ziele, Stakeholder haben Interesse an den Zielen, in dem sie diese unterstützen, verändern oder ablehnen. In einem Projekt müssen die Interessen der Stakeholder ermittelt und zufriedengestellt werden.³⁴⁷

In LeSS sind die Product Owner für das Stakeholdermanagement zuständig. Sie stellen sicher, dass alle Interessenvertreter befriedigt werden. Indem eine direkte Verbindung zwischen den Stakeholder und den Teams hergestellt wird, soll eine Zusammenarbeit ermöglicht werden. Larman und Vodde geben zu bedenken, dass sich Product Owner in

³⁴³ Vgl. Project Management Institute: Pulse of the Profession – In-Depth Report: Navigating Complexity (2013), S. 4 (online)

³⁴⁴ Vgl. Project Management Institute: PMI's Pulse of the Profession – The High Cost of Low Performance (2013), S. 4 ff. (online)

³⁴⁵ Vgl. Project Management Institute: Pulse of the Profession – In-Depth Report: Navigating Complexity (2013), S. 6 (online)

³⁴⁶ Vgl. Lucht (2019), S. 323 zit. n. Freeman (2004), S. 231

³⁴⁷ Vgl. Lucht (2019), S. 326 f.

LeSS leicht überarbeiten können. Bei komplexen Projekten mit einer Vielzahl von internen und externen Stakeholdern, kann die einfache Rollenverteilung schnell an seine Grenzen stoßen.³⁴⁸

SAFe hat hingegen neben dem Product Management, das die Rolle des Product Owners wahrnimmt, drei weitere Teams in die Projektplanung involviert. Das Releaseteam, bestehend aus wichtigen internen Stakeholdern, nimmt Einfluss auf den Verlauf des Projektes, indem es die Anforderungen und Interessen der Stakeholder direkt in das Projekt einbindet.³⁴⁹ Außerdem Systemarchitekten, die System- und Integrationsanforderungen der Stakeholder berücksichtigen.³⁵⁰ Und abschließend ein Business Owner Team, das aus drei bis fünf Stakeholdern besteht und die Managementverantwortung innehat. Sie sorgen dafür, dass Einigkeit unter den Stakeholdern herrscht.³⁵¹

Auch kann SAFe im Gegensatz zu LeSS besser mit starken externen Umwelteinflüssen umgehen, da die unterschiedlichen Projektmanagementteams Einflüsse, Probleme, Chancen und Anforderungen besser identifizieren können. Sicherheitsrelevante oder staatliche Regularien benötigen eine umfangreiche Dokumentation, die durch das klassische Projektmanagement der Rollen im SAFe besser durchgeführt werden kann. Zusammengefasst lässt sich sagen, dass LeSS eine Vielzahl neuer Rollen benötigt, um mit einer großen Anzahl von Stakeholdern, Umwelteinflüssen und Regularien umgehen zu können, während SAFe eine Palette vorgefertigter Rollen zur Verfügung stellt.

Unklare Projektbestandteile und hoher Innovationsgrad des Produktes

Ein innovatives Produkt existiert in seiner Form kein zweites Mal. Demzufolge beinhaltet es Projektbestandteile wie Ressourcen oder Phasen, die unbekannt sind und nicht geplant werden können. Anders gesagt ist bei diesen Projekten das Ziel zwar bekannt, jedoch sind die Aufgaben und Arbeitspakete, die zur Zielerreichung führen sollen, nicht ersichtlich. Die Bearbeitung eines komplexen Projektes ist ein Problemlösungsprozess, bei der routinierte Arbeit schwerlich möglich ist und individuelle Ergebnisse erst erschlossen werden müssen, bis die Zielerreichung eingetreten ist.³⁵² Die Bearbeitung solcher komplexen Probleme kann nur umgesetzt werden, wenn die

³⁴⁸ Vgl. Larman/Vodde (2017), S. 188

³⁴⁹ Vgl. Mathis (2018), S. 83

³⁵⁰ Vgl. Mathis (2018), S. 86

³⁵¹ Vgl. Mathis (2018), S. 88

³⁵² Vgl. Lucht (2019), S. 215

Probleme separiert und diese dann einzeln und nacheinander gelöst werden.³⁵³ Sequentielle Prozesse und eine starre unveränderliche Planung sind für die Zielerfüllung eher hinderlich. Es benötigt flexible Planungsprozesse, die sich an die sich ändernden Umweltbedingungen und Anforderungen anpassen.³⁵⁴ Umgesetzt wird dies zum einen mit den iterativen Vorgehensmodellen, in denen nach Lucht eine Operation mehrfach ausgeführt wird, “ [...] um sich dem Ziel schrittweise zu nähern.“³⁵⁵ Zum anderen erfolgt die Umsetzung mit den inkrementellen Vorgehensmodellen, dort werden möglichst kleine Einzelteile entwickelt, um das große komplexe Problem modular zu lösen.³⁵⁶ Bei beiden Vorgehensmodellen wird von einem agilen Ansatz gesprochen.³⁵⁷ Dementsprechend sind agile Arbeitsweisen für die Zielerreichung von innovativen Produkten vorteilhaft.

Auf LeSS und SAFe bezogen heißt dies, dass der agile Anteil der Prozessstruktur besser mit einer unklaren Zielerreichung und unbekanntem Projektbestandteilen umgehen kann. Beide Frameworks arbeiten auf der operativen Ebene agil. Während LeSS mit Scrum umgesetzt wird,³⁵⁸ kann in SAFe Scrum, Kanban oder XP Programmierung genutzt werden.³⁵⁹ Zwar bemängeln manche Autoren, wie in Kapitel 4.2.2 beschrieben, die Agilität von SAFe, trotzdem liegt der Fokus eindeutig auf der Entwicklung von Productinkrementen in Iterationen. Die Struktur von SAFe ist jedoch vorgegeben und wirkt schwerfällig. LeSS ist dagegen durch seine reduzierte Struktur flexibler und orientiert sich damit stärker an den agilen Werten. Nichtsdestotrotz können in SAFe hinderliche Rollen und Prozesse auch weggelassen oder verändert werden.³⁶⁰ Überhaupt rät Mathis die Rollen abhängig vom Produkt zu besetzen. Er schreibt, dass man bei SAFe „[...] mit einem minimalen Satz [Strukturen] startet und den Rest nach Bedarf hinzufügt oder entfernt.“³⁶¹

Zusammenfassend lässt sich sagen, dass ein komplett umgesetztes SAFe schwerer mit sehr individuellen und innovativen Produkten umgehen kann. Manche Rollen, die dem klassischen Projektmanagement ähneln und versucht sind Pläne, wie die Roadmap oder

³⁵³ Vgl. Lucht (2019), S. 220 zit. n. Rittel/Webber (1973), S. 160 ff.

³⁵⁴ Vgl. Lucht (2019), S. 218 zit. n. Elbe/Peters (2016), S. 95

³⁵⁵ Lucht (2019), S. 221

³⁵⁶ Vgl. Goll/Hommel (2015), S. 2

³⁵⁷ Vgl. Kusay-Merkle (2018) S. 23, zit. n. Project Management Institute (2017) S. 17

³⁵⁸ Vgl. Larman/Vodde (2017), S. 8

³⁵⁹ Vgl. Mathis (2018), S. 9

³⁶⁰ Vgl. Mathis (2018), S. 223

³⁶¹ Mathis (2018), S. 223

Meilensteine zu erstellen, könnten überflüssig sein oder schlimmstenfalls falsche Anforderungen definieren. LeSS hingegen bleibt leichtfüßig und flexibel. Trotzdem kann auch SAFe reduziert angewandt werden – alle Rollen und Prozesse dienen lediglich zur Orientierung und sollen helfen Experimente und Versuche abzukürzen.

Hohes unternehmerisches Risiko

Bei komplexen Projekten ist es schwer erkennbar, welche Alternative den kleinsten finanziellen Aufwand und das geringste Risiko trägt.³⁶² Im klassischen Projektmanagement sollen Hilfsmittel wie das Risikomanagement Risiken bestmöglich identifizieren und daraus Handlungen ableiten. Tatsächlich hat eine Umfrage von PMI über den Einsatz von Projektmanagementtechniken erfolgreicher Unternehmen ergeben, dass Risikomanagement, Leistungsmessung oder das Projekt- und Portfoliomanagement regelmäßig genutzt wird, um ein komplexes Projekt erfolgreich umzusetzen.³⁶³ Das bestätigt Lucht, indem er im Rahmen von agilen Projekten, auch klassische Hilfsmittel als eine sinnvolle Erweiterung der Managementinstrumente sieht.³⁶⁴ Nach Meyer kann gutes Management einem komplexen Projekt helfen. Des Weiteren kann eine gute Vorplanung Änderungen schon im Vorfeld entdecken, damit diese gar nicht erst auftreten.³⁶⁵ Somit scheint eine situative Zuhilfenahme klassischer Instrumente dem Projekterfolg nicht zu schaden.³⁶⁶ Das Portfoliomanagement ist tatsächlich in SAFe integriert. Es beinhaltet die Entwicklung und die Investitionsentscheidungen für das Portfolio.³⁶⁷ Außerdem sind eine Leistungsmessung und ein Risikomanagement zu finden.³⁶⁸ Verglichen wird der Erfolg mit Hilfe von key performance indicators (KPIs).³⁶⁹ Der traditionellen Budgetvergabe sperrt SAFe sich jedoch und hat seine eigene Investitionspraktik, das sogenannte „Lean Budgets“ entwickelt.³⁷⁰

³⁶² Vgl. Lucht (2019), S. 246

³⁶³ Vgl. Project Management Institute: Pulse of the Profession – In-Depth Report: Navigating Complexity (2013), S. 9 (online)

³⁶⁴ Vgl. Lucht (2019), S. 246

³⁶⁵ Vgl. Meyer (2014), S. 150 f.

³⁶⁶ Vgl. Lucht (2019), S. 246

³⁶⁷ Vgl. Mathis (2018), S. 101

³⁶⁸ Vgl. Scaled Agile, Inc: Lean Portfolio Management (online)

³⁶⁹ Vgl. Scaled Agile, Inc: Value Stream KPIs (online)

³⁷⁰ Vgl. Scaled Agile, Inc: Lean Budgets (online)

Auf der operativen Ebene sollen in SAFe Risiken minimiert werden, indem im PI Planungsmeeting, mit Hilfe einer Vertrauensabstimmung am Ende des Planungsprozesses, jedes Teammitglied seine Bedenken äußern kann.³⁷¹

LeSS ist hingegen ganz auf die Agilen Werte ausgelegt – besser auf Änderungen reagieren zu können, wenn kein expliziter Plan verfolgt wird. Dementsprechend finden in LeSS keine klassischen Hilfsmittel Anwendung, was das unternehmerische Risiko von komplexen Projekten vergrößern kann.

Nachdem die drei Eigenschaften der Projektkomplexität hinsichtlich der beiden Frameworks analysiert wurden, lässt sich zusammenfassen, dass SAFe besser komplexe Projekte handhaben kann. Seine ausgeprägten Rollen mit definierten Verantwortungsbereichen sind gut dafür geeignet, mit vielen Stakeholdern umzugehen. Die traditionelle Organisationsstruktur beinhaltet planerische und risikoabsichernde klassische Projektmanagementinstrumente, die das unternehmerische Risiko komplexer Projekte reduzieren.

Im Gegensatz dazu, verringert LeSS bei weniger komplexen Projekten das unternehmerische Risiko, da seine minimalistische Struktur Kosten spart. LeSS ist sehr flexibel und kann schnell auf ändernde Umwelteinflüsse reagieren. Weniger umfangreiche Stakeholdergruppen können von den Mitarbeitern mit einer sehr schnellen Kommunikation und Entscheidungsfindung bearbeitet werden.

Nur bei innovativen Produkten lassen sich keine Unterschiede zwischen LeSS und SAFe feststellen. Beide Frameworks enthalten eine iterative, sowie eine inkrementelle Arbeitsweise. Dadurch können sie unbekannte Projektmerkmale mit einem hohen Innovationsgrad handhaben.

4.5 Gesamtüberblick und Darstellung der Eignung der untersuchten Frameworks

Nachfolgend sind die Ergebnisse der Analyse bezüglich der Frameworks und ihren Umgang mit den relevanten Dimensionen zusammengefasst und in Diagrammen abgebildet. Dadurch ist es für die Entscheidungsträger von Projekte möglich, eine grobe Einordnung zu unterziehen, um darauf aufbauend eines der beiden Frameworks für eine tiefgreifendere Betrachtung auszuwählen.

³⁷¹ Vgl. Mathis (2018), S. 60 ff.

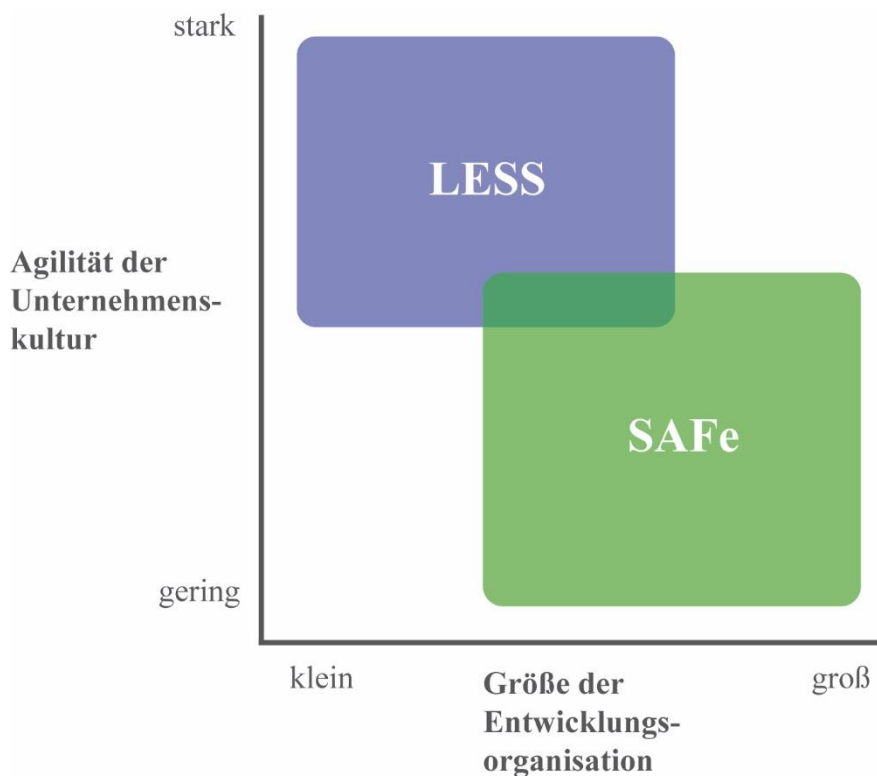


Abb. 7: Darstellung der Frameworks anhand der Dimensionen Agilität der Unternehmenskultur und Größe der Entwicklungsorganisation

Bei der Analyse der Unternehmenskultur konnte festgestellt werden, dass LeSS sich hauptsächlich für eine tief bis mittel tief verankerte agilen Unternehmenskultur eignet. Die Organisation sollte Erfahrung mit agilen Ansätzen haben und auf der Teamebene mit Scrum arbeiten. Bei der Einführung von LeSS wird radikal mit der traditionellen Organisationsstruktur gebrochen. Im Verlauf der Einführung von LeSS müssen agile Prozesse und Methoden über Teamgrenzen hinweg entwickelt werden, was stark ausgeprägte agile Werte und Kompetenzen bei den Organisationsmitgliedern benötigt. Dementsprechend wurde LeSS in den Diagrammen Abb. 7 und Abb. 9 bei einer starken bis mittleren agilen Organisationskultur eingeordnet.

SAFe hat dagegen das Ziel eine traditionellere Organisation zu agil oder lean zu transformieren. Die agile Unternehmenskultur braucht sich durch starrere Organisationvorgaben erst nach der Einführung zu bilden. Fehlende agile Kompetenzen werden mit Hilfe eines umfangreichen Werkzeugkastens ausgeglichen. Das hat zur Folge, dass SAFe in dem Diagramm Abb. 7 und Abb. 9 bei einer niedrigen bis mittleren agilen Unternehmenskultur zugeordnet wird.

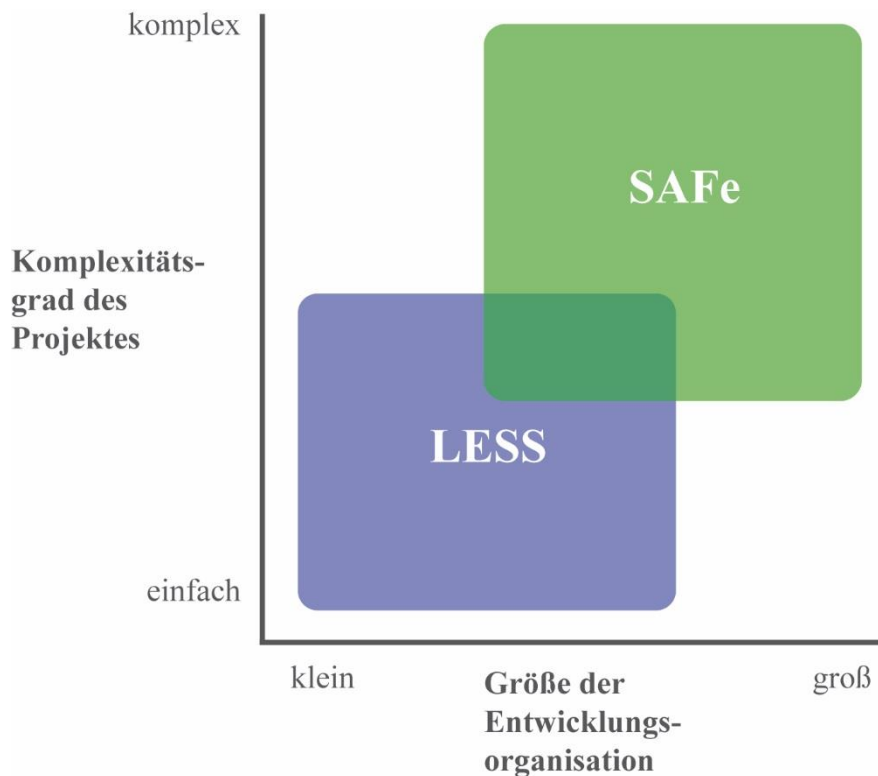


Abb. 8: Darstellung der Frameworks anhand der Dimensionen Komplexitätsgrad des Projektes und Größe der Entwicklungsorganisation

Für eine große Entwicklungsorganisation ist in der Regel SAFe besser geeignet. Die umfangreichen Rollen und Verantwortlichkeiten verringern den Koordinations- und Steuerungsaufwand auf der Teamebene, der sich bildet, wenn viele Projektakteure miteinander agieren müssen. Für kleinere Projekte fällt SAFe jedoch zu mächtig und schwerfällig aus. Die Case Studies von SAFe zeigen die Spanne der Projektakteure von hundert bis mehreren tausend Teammitgliedern. Dementsprechend wurde SAFe in Abb. 7 und Abb. 8 bei mittleren bis sehr großen Entwicklungsorganisationen eingeordnet. LeSS ist mit seiner flexiblen und kommunikationsfördernden Scrum-Struktur und seinen agilen Grundprinzipien gut für kleinere Projekte bis acht Teams geeignet. Darüber hinaus wird das größere LeSS Hufe angewandt. Hier fehlen jedoch weitere Rollen und Prozesse, die mit dem erhöhten Abstimmungsbedarf, der bei vielen Projektakteuren anfällt, umgehen kann. Angesichts dessen wurde in Abb. 7 und Abb. 8 eine Einteilung in kleine bis mittelgroße Projekte vorgenommen.

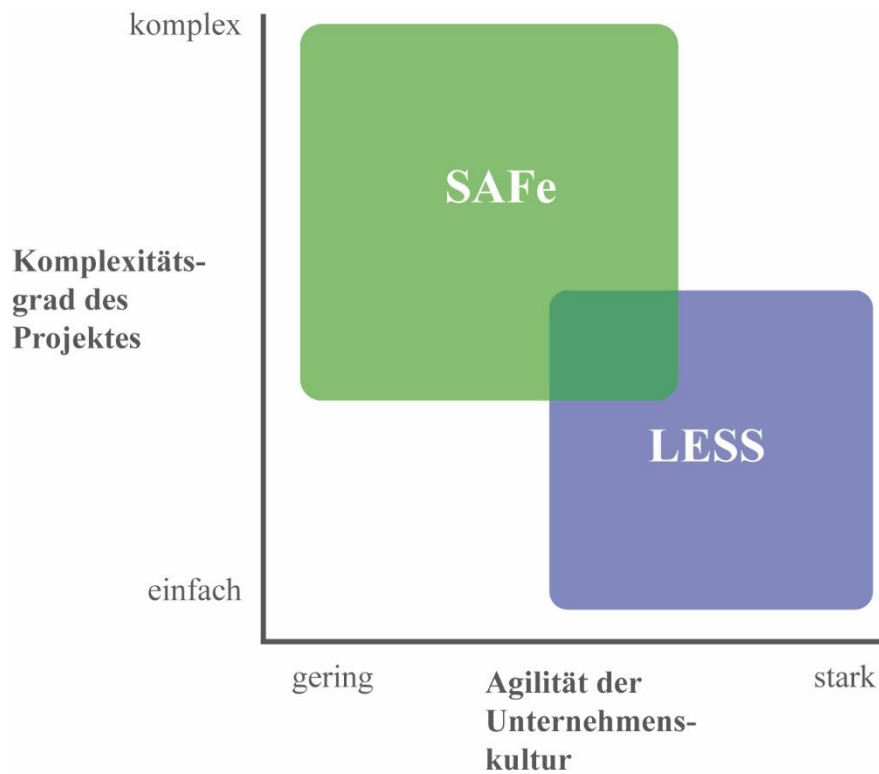


Abb. 9: Darstellung der Frameworks anhand der Dimensionen Komplexitätsgrad des Projektes und Agilität der Unternehmenskultur

SAFe sollte außerdem bei der Entwicklung von komplexen Projekten zum Einsatz kommen (Abb. 8 oder Abb. 9), da es bei allen drei identifizierten Kriterien, die komplexe Projekte ausmachen, überzeugen konnte. Die Einbindung klassischer Instrumente und Unternehmensbereiche kann das unternehmerische Risiko abmildern. Eine Vielzahl neuer Rollen und Gremien schafft Schnittstellen, um mit der wachsenden Anzahl von Stakeholdern umgehen zu können.

Die agile Arbeitsweise, mit der sehr innovative Produkte mit unbekanntem Eigenschaften erst entdeckt und dann entwickelt werden, ist andererseits bei beiden Frameworks vorhanden. Die Einführung von LeSS ist durch die minimalistische Struktur kostengünstiger. Die kommunikationsfreudigen Prozesse lässt die Teammitglieder direkt mit den Stakeholdern kommunizieren. Dadurch kann sehr schnell auf veränderte Anforderungen reagiert werden. Jedoch sind die Teammitglieder und die Product Owner bei zu vielen Stakeholdern überfordert. Die Flexibilität kann dadurch nicht mehr gewährleistet werden.

5 Schlussbetrachtung

5.1 Zusammenfassung

In dieser Arbeit wurden zu Beginn die agilen Grundlagen beschrieben und die agilen Methoden wie Scrum und Kanban erläutert. Anschließend sind die skalierten Frameworks LeSS und SAFe erklärt worden. Mit Hilfe von Projektcharakteristika wurden Dimensionen gebildet, welche die Unterschiede von LeSS und SAFe sichtbar machen, um den Leser für sein agiles Projekt eine Entscheidungshilfe zu bieten.

Die Auswahlkriterien der Frameworks wurden mit Hilfe von Fachliteratur herausgearbeitet und beziehen sich auf die Agilität der Unternehmenskultur, der Größe der Entwicklungsorganisation und dem Komplexitätsgrad des Projektes.

Abschließend fand eine Gegenüberstellung von LeSS und SAFe pro Dimension statt. Die Vor- und Nachteile der Frameworks bei verschieden starker Ausprägung der Dimensionen im Projekt wurden dargelegt und sind visualisiert worden.

5.2 Fazit

Zusammenfassend lässt sich sagen, dass die in der Problemstellung beschriebenen Ziele erreicht werden konnten. Eine Einteilung und Unterscheidung beider Frameworks hinsichtlich relevanter Projekteigenschaften wurde vorgenommen. Anhand der herausgearbeiteten Ergebnisse lässt sich gut erkennen, dass beide Frameworks für agile Projekte geeignet sind. Gleichzeitig beruhen sie jedoch auf unterschiedlichen Herangehensweisen. Während SAFe versucht, agile Projekte durch feste Rahmenbedingungen zu etablieren wird bei LeSS die Skalierung von Scrum bottom-up über die Teamebene hinaus eingeführt. Durch ihre unterschiedliche Herangehensweise ergeben sich ganz verschiedene Vor- und Nachteile wie sich bei der Betrachtung der Dimensionen ablesen lässt. Daher ist zu empfehlen, bei jedem Projekt individuell die Auswahl des Frameworks anhand der gezeigten und auch weiterer, hier nicht betrachteter Dimensionen, vorzunehmen.

Schlussendlich reicht es jedoch nicht aus, sich lediglich für ein Framework zu entscheiden. Das Framework und das eigene Projekt müssen gründlich untersucht und verstanden werden. Nur so kann herausgefunden werden, wo die passenden Antworten

für die eigene Skalierung liegen. Sicher ist jedoch, dass das Wort „Framework“ wörtlich genommen werden sollte. Es handelt sich um Rahmen, in denen Unternehmen ihre Projekte selbst gestalten sollten, weil jedes Vorhaben variiert. Auch wenn SAFe den Eindruck macht ein fertiges Framework zu liefern, ist es eher als Starterkit mit vielen Werkzeugen zu sehen, deren Regeln und Praktiken sinnvoll eingesetzt werden sollten. Dementsprechend können auch Methoden beider Frameworks in einem Projekt situativ eingesetzt werden.

Bei der Auswahl der agilen Literatur ist Vorsicht geboten, viele subjektive Berichte aus der Praxis können Analysen oder gesicherte Erkenntnisse nicht ersetzen.

Auch wenn die agilen Werte den Eindruck machen, sollte das klassische Projektmanagement nicht ausgeschlossen werden. Die traditionellen Instrumente sind erprobt und bewährt. Scheuklappendenken könnte dazu führen, Methoden von vornherein auszuschließen, obwohl sie für das Projekt vorteilhaft sein könnten.

5.3 Ausblick

Der technische Fortschritt, die Individualität der Kundenanforderungen und die Dynamik der Märkte wird dafür sorgen, dass die Komplexität im Projektmanagement immer weiter zunimmt. Auch in Zukunft wird in der Softwareentwicklung kein Weg an den agilen Methoden vorbeiführen. Ob sich skalierbare agile Frameworks längerfristig in Organisationen halten, oder sich doch wieder klassische oder eine hybride Form durchsetzen, kann bis jetzt noch nicht abgeschätzt werden. Dafür sind die eingeführten Frameworks noch zu jung. Es kann jedoch davon ausgegangen werden, dass auch in den nächsten Jahren viele Unternehmen einen Change Prozess im Unternehmen anstoßen wollen, um im Rahmen einer agilen oder Lean-Transformation wettbewerbsfähig zu bleiben. Die Einführung eines skalierbaren Frameworks ist jedoch kein Garant dafür. Vielmehr sind es die Menschen, die hinter dem Vorhaben stehen, den Wandel befürworten und motiviert sind neue Methoden auszuprobieren und daraus zu lernen.

Literaturverzeichnis

Anderson, David / Carmichael, Andy (2016): Essential Kanban Condensed URL: <https://leankanban.com/guide/>, (Datum der Recherche: 19.08.2019).

Beedle, Mike / van Bennekum, Arie / Cockburn, Alistair / Cunningham, Ward / Fowler, Martin / Highsmith, Jim / Hunt, Andrew / Jeffries, Ron / Kern, Jon / Marick, Brian / Martin, Robert / Schwaber, Ken / Sutherland, Jeff / Thomas, Dave (2001): Manifesto for Agile Software Development, URL: <http://agilemanifesto.org>, (Datum der Recherche: 19.08.2019).

Bergsmann, Johannes (2015): Requirements Engineering für die agile Softwareentwicklung – Methoden, Techniken und Strategien, 2. Auflage, Heidelberg: dpunkt.verlag.

Bibliographisches Institut GmbH: Duden, URL: <https://www.duden.de/suchen/dudenonline/agil>, (Datum der Recherche: 19.08.2019).

Bibliographisches Institut GmbH: Duden, URL: <https://www.duden.de/rechtschreibung/skalieren>, (Datum der Recherche: 19.08.2019).

Bleek, Wolf-Gideon / Wolf, Henning (2008): Agile Softwareentwicklung – Werte, Konzepte und Methoden, 1. Auflage, Heidelberg: dpunkt.verlag.

Czudek, Sacha (2015): Ist SAFe das neue Scrum? In: Netzwoche 17/2015, S. 41.

Daut, Patrick (2015): agile@scale: Do more with LeSS or be SAFe? Ansätze zur Skalierung – ein Überblick. In: Projektmanagement und Vorgehensmodelle 2015, Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, 2015, S. 159-167.

Deutsches Institut für Normung (2009): DIN-Taschenbuch: Projektmanagement – Netzplantechnik und Projektmanagementsysteme, 1. Auflage, Berlin: Beuth Verlag.

Fiedler, Rudolf (2016): Controlling von Projekten – Mit konkreten Beispielen aus der Unternehmenspraxis – Alle Aspekte der Projektplanung, Projektsteuerung und Projektkontrolle, 7. Auflage, Wiesbaden: Springer Vieweg.

Gloger, Boris (2017): Scrum Think b!g – Scrum für wirklich große Projekte, viele Teams und Kulturen, 1. Auflage, München: Carl Hanser Verlag.

Goll, Joachim / Hommel, Daniel (2015): Mit Scrum zum gewünschten System, 1. Auflage, Wiesbaden: Springer Vieweg.

Grimm, Reinhard (2009): Einfach komplex – Neue Herausforderungen im Projektmanagement, 1. Auflage, Wiesbaden: VS Verlag.

Hanschke, Inge (2017): Agile in der Unternehmenspraxis – Fallstricke erkennen und vermeiden, Potenziale heben, 1. Auflage, Wiesbaden: Springer.

Hanser, Eckhart (2010): Agile Prozesse: Von XP über Scrum bis MAP, 1. Auflage, Berlin: Springer Verlag.

Herget, Josef (2018): Unternehmenskultur in der Praxis: Grundlagen – Methoden – Best Practices, 1. Auflage, Wiesbaden: Springer Gabler.

Hildebrand, Kay: Agilen Erfolg skalieren – Die Frameworks LeSS, Nexus und SAFe im Vergleich, URL: https://www.viadee.de/wp-content/uploads/Artikel_HiK_AgilenErfolgskalieren.pdf, (Datum der Recherche: 19.08.2019).

Hruschka, Peter /Rupp, Chris / Starke, Gernot (2009): Agility kompakt – Tipps für erfolgreiche Systementwicklung, 2. Auflage, Heidelberg: Spektrum Akademischer Verlag.

James, Michael (2012): Eine Beispiel-Checkliste für Scrum Master, URL: http://scrummasterchecklist.org/pdf/Scrum-Master-Checkliste_de.pdf, (Datum der Recherche: 19.08.2019).

Komus, Ayelt / Kuberg, Moritz (2017): Abschlussbericht: Status Quo Agile 2016/2017 – 3. Studie über Erfolg und Anwendungsformen von agilen Methoden, URL: <https://www.hs-koblenz.de/index.php?id=7169>, Koblenz: BPM-Labor für Business Process Management und Organizational Excellence, Hochschule Koblenz, (Datum der Recherche: 19.08.2019).

Komus, Ayelt / Kuberg, Moritz (2015): Status Quo Agile – Studie zu Verbreitung und Nutzen agiler Methoden, Koblenz: GPM Deutsche Gesellschaft für Projektmanagement e. V in Kooperation mit der Hochschule Koblenz.

Kusay-Merkle, Ursula (2018): Agiles Projektmanagement im Berufsalltag – Für mittlere und kleine Projekte, 1. Auflage, Berlin: Springer Gabler.

Larman, Craig / Vodde, Bas (2017): Large-Scale Scrum – Scrum erfolgreich skalieren mit LeSS, 1. Auflage, Heidelberg: dpunkt.verlag.

Larman, Craig / Vodde, Bas (2009): Scaling Lean & Agile Development – Thinking and Organizational Tools for Large-Scale Scrum, 1. Auflage, New York u.a.: Addison-Wesley.

Lucht, Dietmar (2019): Theorie und Management komplexer Projekte, 1. Auflage, Wiesbaden: Springer Gabler.

Madauss, Bernd-J. (2017): Projektmanagement – Theorie und Praxis aus einer Hand, 7. Auflage, Berlin: Springer Vieweg.

Malik, Fredmund (2004): Systemisches Management, Evolution, Selbstorganisation: Grundprobleme, Funktionsmechanismen und Lösungsansätze für komplexe Systeme, 4. Auflage, Bern: Haupt.

Mathis, Christoph (2018): SAFe: Das Scaled Agile Framework – Lean und Agile in großen Unternehmen skalieren, 2. Auflage, Heidelberg: dpunkt.verlag.

Maximini, Dominik (2018): Scrum – Einführung in der Unternehmenspraxis – Von starren Strukturen zu agilen Kulturen, 2. Auflage, Berlin: Springer Gabler.

Meyer, Bertrand (2014): Agile! – The Good, the Hype and the Ugly, 1. Auflage, Schweiz: Springer Gabler.

Helga, Meyer/ Reher, Heinz-Josef (2016): Projektmanagement – Von der Definition über die Projektplanung zum erfolgreichen Abschluss, 1. Auflage, Wiesbaden: Springer Gabler.

Opelt, Andreas / Gloger, Boris / Pfarl, Wolfgang / Mittermayr, Ralf (2012): Der agile Festpreis – Leitfaden für wirklich erfolgreiche IT-Projekt-Verträge, 1. Auflage, München: Carl Hanser Verlag.

Project Management Institute: Pulse of the Profession In-Depth Report: Navigating Complexity (2013), S. 4 (online)

Project Management Institute: PMI's Pulse of the Profession™: The High Cost of Low Performance (2013) URL: <https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2013.pdf>, (Datum der Recherche: 19.08.2019).

Project Management Institute (2008): A guide to the project management body of knowledge – (PMBOK guide), 4. Auflage, Newtown Square, Pennsylvania: Project Management Institute, Inc.

Roock, Stefan / Wolf, Henning (2016): Scrum – verstehen und erfolgreich einsetzen, 1. Auflage, Heidelberg: dpunkt.verlag.

Sauter, Roman / Sauter, Werner / Wolfig, Roland (2018): Agile Werte- und Kompetenzentwicklung – Wege in eine neue Arbeitswelt, 1. Auflage, Berlin: Springer Gabler.

Schein, Edgar (2018): Organisationskultur und Leadership, 5. Auflage, München: Verlag Franz Vahlen.

Schwaber, Ken / Sutherland, Jeff (2017): Der Scrum Guide – Der gültige Leitfaden für Scrum: Die Spielregeln, URL: <https://www.scrumguides.org/>, (Datum der Recherche: 19.08.2019).

Sutherland, Jeff (2011): Takeuchi and Nonaka: The Roots of Scrum, URL: <https://www.scruminc.com/takeuchi-and-nonaka-roots-of-scrum/>, (Datum der Recherche: 19.08.2019).

Scaled Agile, Inc: Agile Release Train, URL: <https://www.scaledagileframework.com/agile-release-train/>, (Datum der Recherche: 19.08.2019).

Scaled Agile, Inc: Case Studies von SAFe, URL: <https://www.scaledagileframework.com/case-studies/>, (Datum der Recherche: 19.08.2019).

Scaled Agile, Inc: SAFe Case Study: Edgeverve Systems, URL: <https://www.scaledagileframework.com/case-study-edgeverve-systems/>, (Datum der Recherche: 19.08.2019).

Scaled Agile, Inc: SAFe Case Study: Livestock Improvement Corporation, URL: <https://www.scaledagileframework.com/case-study-lic/>, (Datum der Recherche: 19.08.2019).

Scaled Agile, Inc: SAFe Case Study: Standard Bank, URL: <https://www.scaledagileframework.com/standard-bank-case-study/>, (Datum der Recherche: 19.08.2019).

Scaled Agile, Inc: Dev Team, URL: <https://www.scaledagileframework.com/dev-team/>, (Datum der Recherche: 19.08.2019).

Scaled Agile, Inc: Features and Capabilities, URL: <https://www.scaledagileframework.com/features-and-capabilities/>, (Datum der Recherche: 19.08.2019).

Scaled Agile, Inc: Lean Budgets, URL: <https://www.scaledagileframework.com/lean-budgets/>, (Datum der Recherche: 19.08.2019).

Scaled Agile, Inc: Lean Portfolio Management, URL: <https://www.scaledagileframework.com/lean-portfolio-management/>, (Datum der Recherche: 19.08.2019).

Scaled Agile, Inc: PI-Planning, URL: <https://www.scaledagileframework.com/pi-planning/>, (Datum der Recherche: 19.08.2019).

Scaled Agile, Inc: Program Increment, URL: <https://www.scaledagileframework.com/program-increment/>, (Datum der Recherche: 19.08.2019).

Scaled Agile, Inc: Solution, URL: <https://www.scaledagileframework.com/solution/>, (Datum der Recherche: 19.08.2019).

Scaled Agile, Inc: Value Stream KPIs, URL: <https://www.scaledagileframework.com/value-streams/#VKPI>, (Datum der Recherche: 19.08.2019).

The LeSS Company B.V.: Case Studies von LeSS, URL: <https://less.works/case-studies/index.html>, (Datum der Recherche: 19.08.2019).

The LeSS Company B.V.: LeSS Huge, URL: <https://less.works/less/less-huge/index.html>, (Datum der Recherche: 19.08.2019).

The LeSS Company B.V.: Product Backlog Refinement, URL: <https://less.works/less/framework/product-backlog-refinement.html>, (Datum der Recherche: 19.08.2019).

The LeSS Company B.V.: Scrum Master, URL: <https://less.works/less/framework/scrummaster.html>, (Datum der Recherche: 19.08.2019).

The LeSS Company B.V.: Sprint Planning One, URL: <https://less.works/less/framework/sprint-planning-one.html>, (Datum der Recherche: 19.08.2019).

Trepper, Tobias (2012): Agil-systemisches Softwareprojektmanagement, 1. Auflage, Wiesbaden: Springer Gabler.

Uludağ, Ömer / Kleehaus, Martin / Xu, Xian / Matthes, Florian (2017): Investigating the Role of Architects in Scaling Agile Frameworks. In: IEEE 21st International Enterprise Distributed Object Computing Conference, S. 123-132.

Werro, Evelyn (2018): Gruppendynamische Aspekte agiler Frameworks – Agile Teams im Spannungsfeld von Zugehörigkeit, Macht und Intimität, 1. Auflage, Wiesbaden: Springer Verlag.

Zillmann, Mario (2019): Lünendonk-Studie 2019: Scalable Agility – Von der agilen zur digitalen Transformation, 1. Auflage, Mindelheim: Lünendonk & Hossenfelder GmbH.

Eidesstattliche Erklärung

„Hiermit versichere ich, Adrian Meier, ehrenwörtlich, dass ich die vorliegende Bachelorarbeit mit dem Titel: „Ein Vergleich von Large-Scale Scrum [LeSS] und dem Scaled Agile Framework [SAFe] im Hinblick auf ihre Eignung für den Einsatz in Großprojekten unter Betrachtung ausgewählter Dimensionen“ selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen wurden, sind in jedem Fall unter Angabe der Quelle kenntlich gemacht. Die Arbeit ist noch nicht veröffentlicht oder in anderer Form als Prüfungsleistung vorgelegt worden.

Ich habe die Bedeutung der ehrenwörtlichen Versicherung und die prüfungsrechtlichen Folgen (§ 26 Abs. 2 Bachelor-SPO (6 Semester), § 24 Abs. 2 Bachelor-SPO (7 Semester), § 23 Abs. 2 Master-SPO (3 Semester) bzw. § 19 Abs. 2 Master-SPO (4 Semester und berufsbegleitend) der HdM) einer unrichtigen oder unvollständigen ehrenwörtlichen Versicherung zur Kenntnis genommen.“
