

AKTUELLE HERAUSFORDERUNGEN VON ENTWICKLUNGSPROJEKTEN UND DIE LÖSUNG DURCH AGILE SKALIERUNGSFRAMEWORKS

Prof. Dr. Jörg Puchan
Hochschule München
Lothstr. 34, 80335 München
Joerg.puchan@hm.edu

Simon Stieber
INVENSITY GmbH
Giselastr. 3a, 80802 München
Simon.stieber@invesity.com

Janine Strube
INVENSITY GmbH
Brahmsallee 48, 20144 Hamburg
Janine.strube@invesity.com

KEYWORDS

Scaled Agile, Skalierung, agile Skalierungsframeworks, Entwicklungsmethoden, Agilität, Herausforderungen der Entwicklung, Entwicklungsprojekte, agile Methoden

ABSTRACT

Um die Vorteile agiler Methoden abseits reiner Softwareentwicklungen, disziplinübergreifend, mit mehreren agilen Teams und im ganzen Unternehmen zu nutzen, steht eine Reihe an agilen Skalierungsframeworks zur Verfügung, welche darauf ausgelegt sind, agile Methoden zu skalieren. Sie basieren auf den Werten aus dem agilen Manifest, unterscheiden sich jedoch in ihrer Umsetzung. Die (anwendungsnahe) Forschung befasst sich aktuell stark mit dem Vergleich dieser agilen Skalierungsframeworks. Agilität breitet sich zudem auf Branchen und Bereiche aus, die typischerweise nach klassischen Vorgehensmodellen arbeiten, da der klassische Ansatz den Herausforderungen nicht mehr gerecht wird. In dieser Veröffentlichung sollen die Herausforderungen in der Entwicklung von komplexen Produkten definiert und kurz erörtert werden. Basierend darauf werden Kriterien abgeleitet, anhand derer die agilen Skalierungsframeworks verglichen werden können. Dabei soll ein wissenschaftlicher, generell gültiger Ansatz verfolgt werden. Ausgehend davon sollen die möglichen Lösungen der Frameworks auf die Herausforderungen der Entwicklung erläutert werden. Folgendes sind die Kernfragen: Welche Herausforderungen gibt es in Entwicklungsprojekten? Welche können durch agile Skalierungsframeworks gelöst werden? Wie lösen agile Skalierungsframeworks diese Herausforderungen?

HERAUSFORDERUNGEN VON ENTWICKLUNGSPROJEKTEN

Branchenübergreifend begegnen komplexen Entwicklungsprojekten ähnliche Herausforderungen, die durch interne wie externe Faktoren getrieben werden. Die externen Faktoren sind geprägt durch Marktanforderungen und Regulierungen, welche für die jeweilige Branche gelten. Der Markt treibt branchenübergreifend Entwicklungsprojekte durch immer dynamischere Kundenanforderungen, kürzer werdende Technologielebenszyklen sowie den Kosten- und Effizienzdruck (Adelberger und Haft-Zboril 2015;

Artelt und Leonhard 2017; Breu 2018; Gelec und Wagner 2014; Haller 2014; Matharu et al. 2015; Niederer 2003; Stelzhammer und Wolter 2017).

Hinzu kommen Trends wie die Digitalisierung, die zur Folge haben, dass die Komplexität in Entwicklungsprojekten steigt (Etezadzadeh 2016; Gelec und Wagner 2014; Hanna und Kuhnert 2014; Lachmann & Rink o. J.; Matharu et al. 2015; Strumberger 2016; Thommen und Ruoff 2016).

Gleichzeitig erhöht sich auch das Regulierungsniveau aufgrund von steigenden Qualitäts-, Sicherheits- oder Prozessstandards, die Entwicklungsprojekte erfüllen müssen. Konkret können dies Prozessreifegradmodelle wie SPICE und CMMI sein oder Sicherheitsnormen wie die IEC 61508 und die entsprechenden branchenspezifischen Implementierungen wie z.B. die ISO 26262 in der Automobilindustrie (Allesch et al. 2009; Hillenbrand 2012; Stelzhammer und Wolter 2017; Tarnowski 2007).

Die internen Herausforderungen ergeben sich vor allem durch die notwendige Zusammenarbeit mehrerer Teams in großen Entwicklungsprojekten. Diese Teams sind national oder auch international verteilt und setzen sich aus verschiedenen Disziplinen zusammen. In heutigen Entwicklungsprojekten arbeiten z.B. Systementwickler, Hardware-, Mechanik-, Software- und Testteams international zusammen (Dvir et al. 1998; Lee-Kelley und Sankey 2008; May und Carter 2001; Miller 2006; Müller et al. 2017; Picot und Baumann 2007; Schäuuffe 2016; Sekitoleko et al. 2014; Torry-Smith et al. 2014).

Diese Kooperation der verschiedenen Disziplinen hat zur Folge, dass Abhängigkeiten zwischen den Teams koordiniert werden müssen (Ambler 2015b; Malone und Crowsten 1994; Sekitoleko et al. 2014; Torry-Smith et al. 2014). Dazu gehört unter anderem, die unterschiedlichen Entwicklungszyklen der einzelnen Teams zu synchronisieren und die Integration der einzelnen Subprodukte zu planen und zu koordinieren (Ambler 2015c, 2015d; cPrime. Inc. 2014; Edwards 2018; Heikkilä et al. 2013; Lima et al. 2015; Müller et al. 2017). Darüber hinaus wird die Gesamtplanung in komplexen Entwicklungsprojekten aufgrund des Umfangs und der möglichen Vielzahl der Veränderung immer herausfordernder (Bennett und Lemoine 2014; Johansen et al. 2016; Mathis 2016; Roock und Wolf 2016; Suomalainen et al. 2015).

Klassische Vorgehensmodelle bieten keine Antwort auf diese Herausforderungen (Matharu et al. 2015; Nerur et al. 2005; Sutherland 2017; Weber 2015). Agile Methoden und Frameworks wie Scrum lösen diese Herausforderungen auf Teamebene (Larman et al. 2017; Sekitoleko et al. 2014; Sutherland und Scrum Inc. 2018). Die Größe der aktuellen, komplexen Entwicklungsprojekte erfordert jedoch eine Skalierung der agilen Methoden (it-agile GmbH o. J.; Mathis 2016; Sekitoleko et al. 2014).

SKALIERUNG AGILER METHODEN

In den ersten Jahren nach dem agilen Manifest (2001) lag der Fokus der Forschung der agilen Methoden auf einzelne Teams in der Softwareentwicklung. Das Wissen zu agilen Methoden hat sich im Laufe der Zeit jedoch erweitert und die Vorteile, welche mit einer agilen Arbeitsweise einhergehen, wollen auch von größeren und komplexeren Vorhaben genutzt werden. Auf der XP2010 Konferenz wurde „Agil im Großen“ schließlich zur dringlichsten Forschungsfrage gewählt (Mathis 2016, S. 2; Moe et al. 2016, S. 1).

Diese Thematik wird auch unter dem Begriff „Skalierung“ geführt. Im agilen Umfeld bezeichnet Skalierung die Situation: „wenn mehrere Teams am selben Produkt arbeiten und/oder Agilität für das ganze Unternehmen verwendet werden soll“ (it-agile GmbH o. J.). Dabei kann zwischen vertikaler und horizontaler Skalierung unterschieden werden (siehe Abbildung 1).

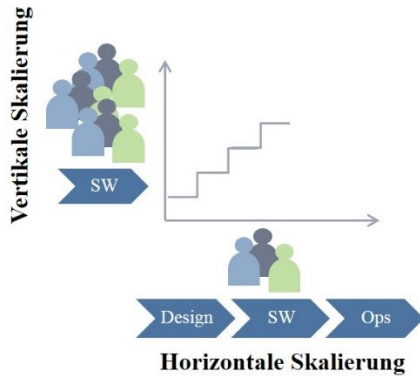


Abbildung 1: Vertikale und Horizontale Skalierung (in Anlehnung an it-agile GmbH o.J.)

Demnach bezeichnet vertikale Skalierung, wenn mehrere Teams für den gleichen Teil der Wertschöpfung eines Unternehmens verantwortlich sind, also beispielsweise, wenn mehrere Teams an einem Softwareprodukt arbeiten. Horizontale Skalierung beschreibt, wenn ein Team einen größeren Teil der Wertschöpfung übernimmt, also z.B. neben der reinen Softwareentwicklung, auch den Betrieb der Software verantwortet (ebd.).

Das Scaling Agile Model, welches im Rahmen des Scalare Projekts entwickelt wurde, bezeichnet die vertikale Skalierung als „Size“ und die horizontale Skalierung als „Value Stream“. Weiterhin beschreibt das Scaling Agile Model eine dritte Skalierungsebene

„Offerings“, welche das Ziel verfolgt weitere Wertströme zu installieren, um weitere Produkte oder Dienstleistungen anzubieten (Morin 2015, S. 8f.).

In diesem Artikel bezeichnet Skalierung die Ausdehnung agiler Methoden auf mehrere Teams und/oder das ganze Unternehmen, in Anlehnung an die Definition von it-agile GmbH o. J.

AGILE SKALIERUNGSFRAMEWORKS

Agile Methoden, wie beispielsweise Scrum, bieten keine Ansätze die Herausforderungen der Skalierung zu lösen (Czudek 2015). Daher wurden agile Skalierungsframeworks entwickelt: agile Rahmenwerke, welche in einem skalierten Umfeld zur Anwendung kommen. Sie fassen Methoden und Praktiken zusammen, um mit mehreren agilen Teams eine agile Entwicklung durchzuführen (Daut 2015, 2016; Schiller 2016). Übergreifend wird das Thema unter Scaled Agile zusammengefasst (Laanti 2014). Obwohl Skalierung agiler Methoden ein relativ junges Forschungsgebiet ist, findet sich eine große Anzahl an agilen Skalierungsframeworks. Im Rahmen der Recherche für diese Arbeit wurden insgesamt 18 Skalierungsframeworks identifiziert (siehe Tabelle 1), welche zum Großteil öffentlich auf ihren Homepages zugänglich sind. Neben offiziellen Konferenzen und Artikeln wird auch zunehmend auf Internetplattformen oder Blogs über Skalierung diskutiert (Arnold 2015; Francino 2016; Hastie 2014).

Tabelle 1: Skalierungsframeworks

Werte	Transformation	Konkrete Vorgehensweise
ScALeD (Beck et al. o. J.)	Agility Path / Evidence-Based Change (Verheyen 2013)	Scaled Agile Framework (SAFe) (Scaled Agile, Inc. 2017a)
	Agile Scaling Cycle (Roock 2014)e	Large-Scale Scrum (LeSS) (The LeSS Company B.V. 2017d)
	LeadingAgile (LeadingAgile 2018)	Nexus (Schwaber und Scrum.org 2018b)
	Scrum Lean in Motion (SLIM) (Laing 2013)	Scrum@Scale (Sutherland und Scrum Inc. 2018)
	Enterprise Transition Framework (agile42 2018)	Disciplined Agile (DA) (Lines und Ambler 2017a)
	SCARE (Heusser 2014)	Spotify Engineering Culture (Kniberg und Ivarsson 2012)
		FAST Agile Scaled

		Technology (Cron Technologies LLC 2017)
		Enterprise Scrum (Beedle und Enterprise Scrum Inc 2010)
		Matrix of Services (Maxos 2016)
		Enterprise Unified Process (Ambler 2013b)
		XSCALE (Merel 2016)

Die recherchierten, agilen Skalierungsframeworks können in ihrer Ausprägung unterschieden werden. ScALeD behandelt ausschließlich die einer agilen Skalierung zugrundeliegenden Werte und Prinzipien (Beck et al. o. J.). Sechs der 18 recherchierten Skalierungsframeworks, wie beispielsweise Agility Path, beschäftigen sich mit dem Entwurf eines individuellen Skalierungsframeworks (Verheyen 2013). Die Vielzahl jedoch, insgesamt elf Skalierungsframeworks wie z.B. SAFe oder Nexus, beschreiben (teilweise zusätzlich zu den Werten, Prinzipien und Implementierungshinweisen) eine Vorgehensweise oder Struktur zur agilen, skalierten Produktentwicklung (Scaled Agile, Inc. 2017a; Schwaber und Scrum.org 2018b).

Nachfolgend werden fünf Frameworks aus der Kategorie „konkrete Vorgehensweise“ vorgestellt und behandelt: Disciplined Agile (DA) Framework, Large-Scale Scrum (LeSS), Nexus, Scaled Agile Framework (SAFe) und Scrum@Scale.

Gründe für die Auswahl dieser fünf Skalierungsframeworks aus dieser Kategorie sind die Bekanntheit und Verbreitung der Frameworks, was sich auch dadurch zeigt, dass sie vermehrt in Vergleichen thematisiert werden (Czudek 2015; Heusser 2015; Alqudah und Razali 2016; Mathis und Fish 2015).

Informationen zu den fünf betrachteten Frameworks sind ebenfalls online erhältlich. SAFe, DA und LeSS bieten informative Websites mit intelligenten Infografiken, aber auch Bücher und Whitepaper, welche teilweise noch mehr ins Detail gehen (Ambler und Lines 2012, 2017; Larman et al. 2017; Lines und Ambler 2017a; Mathis 2016; Scaled Agile, Inc. 2017a; The LeSS Company B.V. 2017d).

Nexus und Scrum@Scale werden wie Scrum in einem schlanken Guide beschrieben (Schwaber und Scrum.org 2018a; Schwaber und Sutherland 2016)).

Alle Frameworks bieten zudem offizielle Trainings und Zertifizierungen an, veröffentlichen Case Studies und betreiben eine Community (Disciplined Agile

Consortium 2017; Scaled Agile, Inc. 2017a; Scrum Inc. 2017; Scrum.org 2018; The LeSS Company B.V. 2017d).

Verbreitung und Nutzung der Skalierungsframeworks in der Praxis

Laut dem „Status Quo Report 2017“ der Hochschule Koblenz nutzen momentan 22% der Befragten ein Skalierungsframework oder eine Skalierungsmethode. Davon nutzen die Hälfte SAFe, gefolgt von LeSS und einem individuell gestalteten Framework. Weniger oft – mit jeweils 8% der Befragten – werden Scrum@Scale oder Nexus genutzt (Komus 2017, S. 125).

Der „11th Annual State of Agile Report“ aus dem Jahr 2017 gibt andere Werte an, jedoch ist auch hier SAFe mit 28% das populärste agile Framework. LeSS nannten 3%, DA und Nexus wurde von jeweils 1% der Befragten genannt (siehe Abbildung 2) (VersionOne 2017, S. 14).

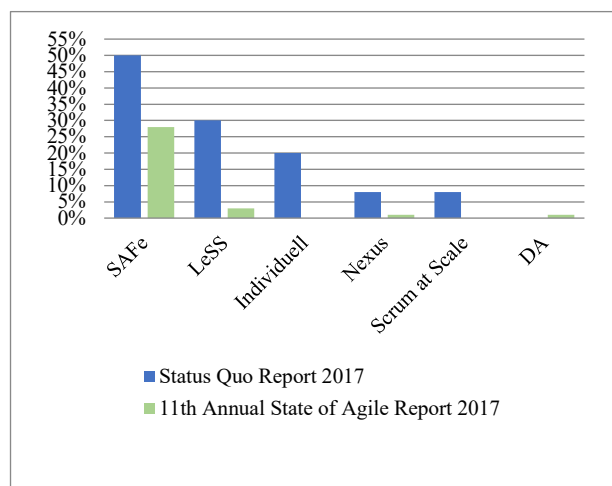


Abbildung 2: Verbreitung der agilen Frameworks (Komus 2017, S. 125; VersionOne 2017, S. 14)

Kurzvorstellung Skalierungsframeworks

Scaled Agile Framework – SAFe

SAFe ist ein freizugängliches Framework, das Rollen, Artefakte und Events beschreibt, und dadurch Unternehmen hilft, schlanke und agile Prinzipien auf das ganze Unternehmen auszubreiten, wodurch die zunehmende Systemkomplexität beherrscht werden kann. Das Skalierungsframework basiert auf vier Kernwerten, einem Lean-Agile Mindset, SAFe Prinzipien sowie Lean-Agilen Leadern (Brenner und Wunder 2015; Mathis 2016; Prowareness GmbH 2015; Scaled Agile, Inc. 2017b).

SAFe ist seit der Version 4.5 konfigurierbar und daher anpassbar auf die jeweilige Unternehmensstruktur. Es besteht aus vier Ebenen (Team, Program, Large Solution und Portfolio), die in vier Konfigurationen verknüpft werden können: Vom „Essential SAFe“ für Projekte von kleinerer Skalierung (50-125 Beteiligten) bis hin zum „Full SAFe“ für mehrere hundert oder tausend Beteiligte,

indem alle vier Ebenen im Unternehmen vorhanden sind (Prowareness GmbH 2015; Scaled Agile, Inc. 2017b). Zentrales Element bei allen Konfigurationen ist der „Agile Release Train“ der Program-Ebene, in welchem mehrere agile Teams, Stakeholder und andere Beteiligte (50-125 Personen) zusammenkommen und gemeinsam inkrementell eine Lösung entwickeln und ausliefern. Synchronisiert werden die Teams und Beteiligten des Agile Release Trains über die gemeinsame Iteration, die zwischen acht bis zwölf Wochen bzw. typischerweise auf fünf Iterationen der agilen Entwicklungsteams festgesetzt werden und bei SAFe als „Program Increment“ bezeichnet wird (Brenner und Wunder 2015, S. 1; Scaled Agile, Inc. 2017j, 2017n).

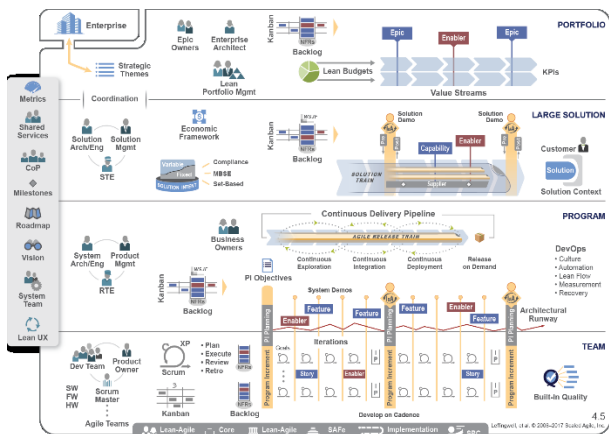


Abbildung 3: Scaled Agile Framework (Scaled Agile, Inc. 2017a)

Large-Scale Scrum – LeSS

Large-Scale Scrum (LeSS) beruht auf Prinzipien, Regeln und Guidelines. Die Regeln definieren die grundlegende Struktur des Skalierungsframeworks, die Verantwortlichkeiten und den Ablauf (Prowareness GmbH 2015; The LeSS Company B.V. 2017c).

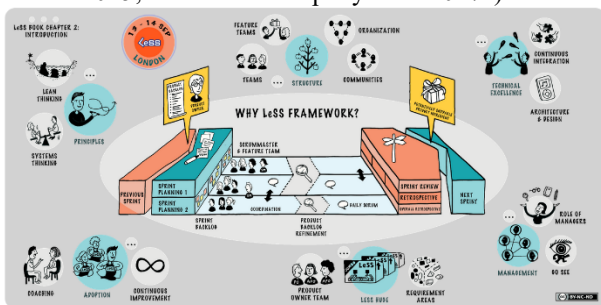


Abbildung 4: Large-Scale Scrum (LeSS) (The LeSS Company B.V. 2017d)

Es gibt zehn Prinzipien, welche die Implementierung von LeSS in einem spezifischen Kontext erleichtern sollen. Ein zentrales Prinzip ist „Large-Scale Scrum is Scrum“, wodurch die zentrale Bedeutung von Scrum für das Skalierungsframework ausgedrückt wird, denn LeSS erweitert Scrum nur um wenige Skalierungselemente., wie die Aufteilung des Sprint Plannings und des Product Backlog Refinements in einen teamübergreifenden Teil

und einen nachfolgenden teaminternen zweiten Teil, das Daily Scrum, bei welchem sich die Teams gegenseitig beobachten können, Regeln der Koordination, das gemeinsame Sprint Review und die gemeinsame Retrospektive (Prowareness GmbH 2015; The LeSS Company B.V. 2017e, 2017b).

Zwei Varianten des Frameworks werden vorgeschlagen: LeSS und LeSS Huge. LeSS ist für Produktentwicklungen von bis zu acht Teams mit jeweils acht Mitgliedern geeignet, LeSS Huge ist für größere Entwicklungen von bis zu mehreren Tausend Personen, die an einem Produkt arbeiten, ausgelegt (The LeSS Company B.V. 2017b).

LeSS wurde bereits bei Produktentwicklungen mit 2 Teams bis hin zu großen Unternehmen mit 2500 Personen angewendet (The LeSS Company B.V. 2017a).

Disciplined Agile Framework (DA)

Das Disciplined Agile (DA) Framework ist ein Entscheidungsframework, welches Anleitung bietet, die Prozesse kontextspezifisch anzupassen und damit eine Grundlage für Agilität im ganzen Unternehmen zu schaffen. Dafür beschreibt es das Zusammenspiel der verschiedenen Bereiche eines Unternehmens wie Produktentwicklung, DevOps, IT, aber auch Portfolio Management oder die Unternehmensarchitektur. Für alle Aktivitäten gibt das DA Framework mögliche Methoden vor und beschreibt die Vor- und Nachteile (Ambler und Lines 2012; Lines und Ambler 2017a, 2017c).

DA kann in vier Skalierungsbereiche aufgeteilt werden (siehe Abbildung 5). Ausgehend von Disciplined Agile Delivery (DAD), welches den End-to-End-Lebenszyklus von der Entwicklung und Lieferung von IT-Lösungen definiert, über Disciplined DevOps und Disciplined Agile IT (DAIT) bis zum gesamtheitlichen Disciplined Agile Enterprise (DAE), welches das ganze Unternehmen mit einer agilen Kultur und schlanken sowie agilen Prozessen umfasst, die eine schnelle Reaktion des Unternehmens auf Veränderungen des Marktes erleichtert (Lines und Ambler 2017b).

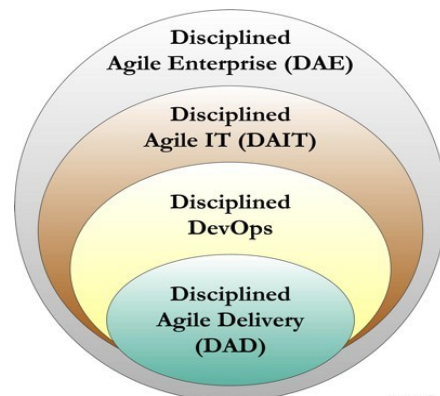


Abbildung 5: DA Framework (Lines und Ambler 2017c)

Nexus

Das Nexus-Framework wird als Exoskelett bezeichnet, welches mehrere Scrum Teams zusammenfasst, die

gemeinsam an einem Produkt arbeiten. Es ist für eine skalierte Entwicklungsumgebung von 3-9 Teams geeignet (Schwaber und Scrum.org 2018a).

Basierend auf Scrum besteht Nexus aus Rollen, Events und Artefakten sowie den Regeln, welche die einzelnen Elemente miteinander verknüpfen (Schwaber und Scrum.org 2018a; Scrum.org 2018).

Das Nexus Framework verwendet einen iterativen und inkrementellen Ansatz, um Produkt- oder Softwareentwicklung zu skalieren und ergänzt Scrum nur minimal durch das Integrationsteam als zusätzliche Rolle und die Nexus-Events und –Artefakte, welche die Scrum Events und Artefakte ergänzen oder ersetzen (Prowareness GmbH 2015; Schwaber und Scrum.org 2018a; Scrum.org 2018).

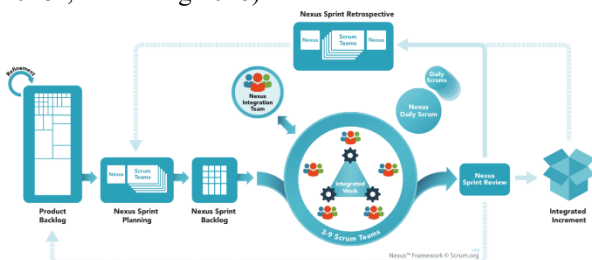


Abbildung 6: Nexus Framework (Scrum.org 2018)

Scrum@Scale

Scrum@Scale ist ein high-level Framework, das ebenso auf Scrum beruht und einen modularen Ansatz verfolgt. Es besteht aus zwei Zyklen – den Scrum Master Zyklus und dem Product Owner Zyklus – welche sich an zwei Stellen treffen, beim Modul „Team-Level Process“ und beim Modul „Product & Release Feedback“. Insgesamt besteht dieses Skalierungsframework aus zwölf Modulen (siehe Abbildung 6), welche lose miteinander verknüpft sind, was es erlaubt, diese inkrementell einzuführen und sich dabei auf jedes unabhängige Modul einzeln zu fokussieren. Erfolgreiche Ansätze für die Module werden gesammelt, die in verschiedenen Kontexten verwendet werden können (Sutherland 2014; Sutherland und Brown 2014).

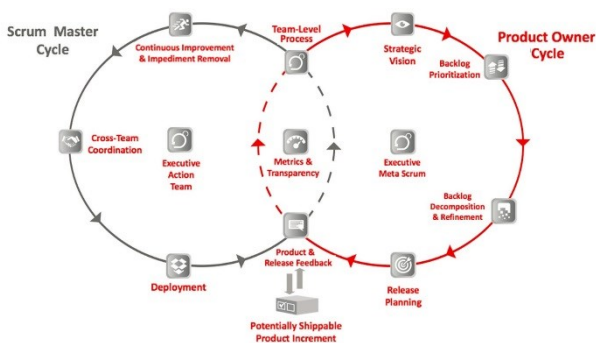


Abbildung 6: Scrum@Scale Framework (Sutherland und Scrum Inc. 2018, S. 4)

Jedes Modul wird durch seine Ziele, die benötigten Inputs und die zu generierenden Outputs definiert und darüber auch mit den anderen Modulen verbunden. Wie die Inhalte der Module umgesetzt werden, bleibt dem

Team überlassen (Sutherland 2014; Sutherland et al. 2015).

LÖSUNGEN FÜR DIE HERAUSFORDERUNGEN DURCH DIE SKALIERUNGSFRAMEWORKS

Extern

Die externe Beeinflussung von Entwicklungsprojekten durch den Markt, durch vorhandene Regulierung und durch neue Technologien, können durch die Einführung von agilen Skalierungsframeworks nicht geändert werden, da sie nicht im Einflussrahmen eines einzelnen Unternehmens liegen.

Dennoch bieten die Skalierungsframeworks Ansätze und Vorgehensweisen, um den Herausforderungen aus externen Faktoren besser begegnen zu können.

Um schneller auf dynamische Kunden- und Marktanforderungen zu reagieren, sind agile Methoden prädestiniert (Maruping et al. 2009).

Auch die agilen Skalierungsframeworks nutzen agile Methoden, wie Scrum, auf der Entwicklungsebene und bieten zudem noch Strukturen und Methoden, um auch die anderen Unternehmensbereiche agiler zu gestalten. Damit kann das ganze Unternehmen agil und flexibel auf sich verändernde und dynamische Anforderungen des Kunden oder des Markts reagieren.

SAFe bindet den Kunden während der Entwicklung als ein zentrales Mitglied ein, und der Kunde ist aktiver Teilnehmer an verschiedenen Events. Er nimmt in der Rolle des Business Owners am Product Increment Planning (Teamübergreifendes Planungsevent) teil, unterstützt die Evaluierung der Produktinkremente bei System Demos und nimmt an den Inspect&Adapt (I&A) Workshops teil. Zudem arbeitet er durchgehend eng mit dem Solution und Program Management zusammen und unterstützt dieses beim Managen von Umfang, Zeit und bei der Erstellung von Roadmaps. Dadurch hat der Kunde einen starken Einfluss auf die Vision des Produkts und auf die Priorisierung der umzusetzenden Arbeit (Scaled Agile, Inc. 2017p).

Zudem fordert SAFe in einem gewissen Rhythmus sogenannte „Innovation & Planning“-Iterationen, in denen die Teams intensiv zu neusten Trends und Technologien recherchieren und sich mit den Markt- und Kundenanforderungen auseinandersetzen (Scaled Agile, Inc. 2017g).

Auch beim Disciplined Agile (DA) Framework sollen alle Stakeholder in den Entwicklungsablauf involviert sein. DA bezeichnet sich als Entscheidungsframework, weshalb es generell viele Möglichkeiten vorgibt, die Prozesse auszugestalten (Lines und Ambler 2017a, 2017c). Daher ist es mit dem DA-Framework möglich, durch eine flexible Work Item Liste Anforderungen, Prioritäten und Umfänge jederzeit im Projekt anzupassen, sogar während der aktuellen Iteration (Ambler und Lines 2012). Auch Nexus und LeSS binden

den Kunden aktiv in das übergeordnete Sprint Review ein, welches am Ende der gemeinsamen Sprints aller Entwicklungsteams stattfindet, um direkt ein Feedback zum aktuellen Entwicklungsstand des Produktinkrements zu erhalten und auch entsprechende Anpassungen durchzuführen (Schwaber und Scrum.org 2018b; The LeSS Company B.V. 2018i).

Bei Scrum@Scale erfolgt die Abstimmung über MetaScrums – Events, die mindestens einmal pro Sprint stattfinden und in welchen alle Produkt Owner, Stakeholder und der Kunde zusammenkommen, um die Prioritäten des Product Backlogs festzulegen. Insbesondere der Chief Product Owner, welcher die Anforderungen an das gesamte Produkt über alle Teams hinweg koordiniert, überwacht das Kundenfeedback und lässt dieses in das Product Backlog einfließen (Sutherland und Scrum Inc. 2018).

Auch beim Thema einer kürzeren Time-to-Market spielt Agilität an sich eine große Rolle, da durch agile Methoden die Time-to-Market häufig verkürzt wird, nachdem in kleineren Einheiten entwickelt wird und die Entwicklungsphasen parallelisiert werden (Rahimian und Ramsin 2008; Rook und Wolf 2016).

Alle Skalierungsframeworks basieren auf einer Entwicklungsmethode wie Scrum oder Kanban auf der Entwicklungsebene und empfehlen dort die häufige bis hin zur kontinuierlichen Integration, wodurch Fehler direkt entdeckt und behoben werden können. Zudem soll mindestens zum Ende des Sprints ein integriertes und getestetes Inkrement zur Verfügung stehen, welches potenziell auslieferbar ist, wodurch die Time-to-Market erheblich verkürzt wird. Durch die kurzen Feedbackschleifen mit dem Kunden im Sprint Review, können Anpassungen früh erfolgen und dadurch der gewünschte Zustand des Produkts schneller erreicht werden (Ambler und Lines 2017; Scaled Agile, Inc. 2017k, 2017m; Schwaber und Scrum.org 2018b; Sutherland und Scrum Inc. 2018; The LeSS Company B.V. 2018b, 2018i).

Eine kürzere Time-to-Market durch den Einsatz der Skalierungsframeworks wird auch durch Implementierungsbeispiele der Skalierungsframeworks gestützt. In diesen werden die Verbesserungen, welche durch die Einführung des jeweiligen Skalierungsframeworks erzielt wurden, in Erfahrungsberichten zusammengefasst (Disciplined Agile Consortium 2018; Scaled Agile, Inc. 2018; Scrum.org 2018; Scrum@Scale 2018; The LeSS Company B.V. 2018d). Nexus berichtet anhand von vier konkreten Use Cases, dass in diesen Fällen häufiger integrierte Inkremente erstellt, diese leichter ausgeliefert und dadurch auch die Time-to-Market verkürzt wurde (Scrum.org 2018). Ebenso geben LeSS, DA und Scrum@Scale, welche jeweils mehrere Beispiele aus verschiedensten Branchen bereitstellen, an, die Time-to-

Market verkürzt zu haben. So z.B. LeSS bei Agfa Healthcare, DA bei Barclays und Scrum@Scale bei Bosch (Howard 2017; Linders 2016; Smet 2018). Scaled Agile, Inc. fasst die Ergebnisse der SAFe Implementierungsbeispiele zusammen und gibt in konkreten Werten an, dass Unternehmen, die SAFe implementieren, ihre Time-to-Market zwischen 30% und 75% reduzieren (Scaled Agile, Inc. 2017d).

Zu den Themen Kostenminimierung und Effizienz nimmt Scaled Agile, Inc. ebenso sehr konkret Stellung und gibt an, dass sich die Produktivität um 20% bis 50% erhöht (Scaled Agile, Inc. 2017d). Die anderen Frameworks quantifizieren die Erfolge zwar nicht, jedoch finden sich in ihren Case Studies oder in den Beschreibungen zu den Frameworks Bezüge zur Verbesserung der Effizienz bzw. eine Erhöhung der Produktivität (Schwaber und Scrum.org 2018b; Scott Ambler + Associates 2016; Sofer 2018; Sutherland und Scrum Inc. 2018).

Besonders sind an dieser Stelle die häufigen Abstimmungen zwischen den Teams, aber auch mit den koordinierenden Teams oder Ebenen zu nennen. Dadurch wird die Transparenz der Artefakte erhöht, es werden Abstimmungsfehler vermieden und Herausforderungen und „Impediments“ können angesprochen und schnell und direkt beseitigt werden. Insgesamt können so alle Teams besser planen und effizienter arbeiten, wodurch auch die Zufriedenheit der Mitarbeiter und schlussendlich auch des Kunden erhöht wird (Scaled Agile, Inc. 2017d, 2017q; Schwaber und Scrum.org 2018b; Sutherland und Scrum Inc. 2018).

Generell können Abstimmungen zwischen den Teams zu Beginn und am Ende der gemeinsamen Iterationen, täglich im Daily Scrum und auch ad-hoc bei Bedarf stattfinden. Ein gemeinsamer Austausch über die Teams hinweg findet bei allen Skalierungsframeworks an den Start- bzw. Endmeetings der Iterationen statt. Nexus und Scrum@Scale planen darüber hinaus ein tägliches teamübergreifendes Abstimmungsevent ein – Nexus nennt dies ein „Nexus Daily Scrum“, Scrum@Scale ein „Scaled Daily Scrum“ im Rahmen eines „Scrum of Scrums“ – in welchem die Scrum Master oder sonstige Vertreter der Teams zusammenkommen und über Hindernisse und Abhängigkeiten zwischen den Teams sprechen (Schwaber und Scrum.org 2018a, S. 9; Sutherland und Scrum Inc. 2018, S. 5). LeSS verfolgt hier einen flexibleren Ansatz, denn die Teams können den Daily Scrums anderer Teams als Zuhörer beiwohnen und die Erkenntnisse in ihr Team zurücktragen. Generell wird bei LeSS ein Kommunizieren nach Bedarf empfohlen (The LeSS Company B.V. 2018b). DA gibt als Entscheidungsframework auch eine Auswahl an Strategien zur Abstimmung zwischen den Teams. So kann eine Ad-hoc-Strategie verfolgt werden, ebenso wie ein tägliches Koordinationsmeeting im Team (Daily

Coordination Meeting) und ein Scrum of Scrums stattfinden (Ambler und Lines 2012).

Durch die Kombination der Vorteile der Skalierungsframeworks – kürzere Time-to-Market, höhere Produktivität sowie zufriedenerer Mitarbeiter und Kunden – werden auch die wirtschaftlichen Ziele von skalierten Entwicklungsprojekten besser erreicht (Scaled Agile, Inc. 2017d; Roock und Wolf 2016).

Häufig wird der Standpunkt vertreten, dass agile Methoden nicht geeignet sind, sollte die Einhaltung von strikten Standards nachgewiesen werden müssen. Dieser Nachweis wird meist in Form von Dokumentation erbracht (Kittel 2013; Theunissen et al. 2003; Wolfig 2016b). Da es eine Vielzahl an Regulierungen gibt, welche sich in den Branchen und auch in den geforderten Nachweisen unterscheiden, wird nun die prinzipielle Kombination der Skalierungsframeworks mit der Einhaltung von gesetzlichen Standards oder Vorgaben der Industrie überprüft (Ambler 2013a).

In diesem Bereich zeichnen sich große Unterschiede zwischen den Skalierungsframeworks ab. Während LeSS und Nexus keine Angabe bezüglich der Einhaltung eines Standards machen, stellen SAFe und DA konkrete Hinweise als auch konkrete Beispiele bereit. SAFe installiert für den Umgang mit Normen und Standards ein agiles Qualitätsmanagementsystem und DA gibt verschiedene Strategien vor, wie mit Standards umgegangen werden kann. Beispielsweise soll der gesamte Lebenszyklus betrachtet werden, da auch die Standards häufig den gesamten Lebenszyklus abdecken, und es wird empfohlen einen zielgerichteten Ansatz zu verfolgen – beides sind Kernelemente von DA (Ambler 2013a; Lines und Ambler 2012; Scaled Agile, Inc. 2017i). Scrum@Scale bindet den Bereich „Legel / Compliance“ als eigenes Scrum Team in die Gesamtorganisation ein, gibt jedoch keine konkreten Hinweise. Die Anwendung von Scrum@Scale in einem regulatorischen Umfeld wird durch die Case Studies bestätigt, z.B. durch die Entwicklung eines Jets (Furuhjelm et al. 2017; Scrum@Scale 2018).

Eine weitere externe Herausforderung ist die hohe Komplexität der Entwicklungsprojekte, die sich durch die Digitalisierung in vielen Bereichen ergibt (Peters et al. 2016; Schuh et al. 2017).

Diese steigende Komplexität lässt sich nur schwer mit klassischen Methoden beherrschen, daher liegt es nahe, die Vorteile von agilen Methoden für komplexe Projekte zu untersuchen. Der Zusammenhang agiler Methoden, wie z.B. Scrum, mit komplexen Produktentwicklungsprojekten kann mit der Stacey-Matrix erklärt werden, welche als Entscheidungsgrundlage für die Wahl eines geeigneten Vorgehensmodells dienen kann (siehe Abbildung 7) (Bergen 2016; Gelec und Wagner 2014; Roock und Wolf 2016; Wolfig 2016a).

Dort werden Projekte als komplex bezeichnet, wenn die Anforderungen unklar und der Weg der Durchführung (Technologie) unsicher sind. Agile Vorgehensweisen sind für diese Projekte am besten geeignet, denn im Vergleich zu komplizierten Projekten kann der Ursache-Wirkungszusammenhang bei komplexen Projekten erst im Nachhinein sicher festgestellt werden. Langfristige Planung (wie bei komplizierten Projekten) funktioniert nicht, sondern eine kurzfristige Planung, mit einer anschließenden Analyse sowie einer darauf basierenden Neuplanung der agilen Frameworks sind hier geeignet (Roock und Wolf 2016).

Dieses Konzept wird auch in den betrachteten agilen Frameworks umgesetzt, in dem die kontinuierliche Verbesserung auf Basis des erworbenen Wissens ermöglicht wird – sowohl auf Teamebene als auch teamübergreifend in gemeinsamen Retrospektiven am Ende der Iterationen (Lines und Ambler 2018; Scaled Agile, Inc. 2017g; Schwaber und Scrum.org 2018b; Sutherland und Scrum Inc. 2018; The LeSS Company B.V. 2017b).

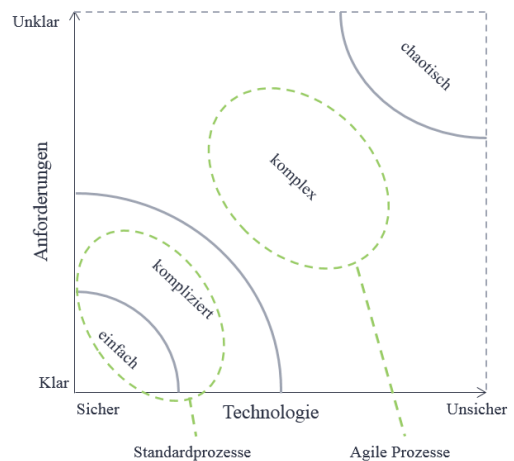


Abbildung 7: Stacey-Matrix (in Anlehnung an Bergen 2016; Roock und Wolf 2016)

Die technische Komplexität durch Digitalisierung kann durch einen fachlichen Austausch über die cross-funktional besetzten Teams hinweg in Communities of Practice adressiert werden. In diesen Communities kommen Personen auf freiwilliger Basis zusammen, welche ein gemeinsames Interesse an einem bestimmten technischen oder rollengebundenen Thema haben. Diese Communities of Practice sind sowohl bei LeSS, SAFe und DA in das Skalierungsframework integriert (Lines und Ambler 2018; Scaled Agile, Inc. 2017r; The LeSS Company B.V. 2018a).

Intern

Im Gegensatz zu den externen können die internen Herausforderungen durch die Entwicklung nach einem agilen Skalierungsframework verbessert oder gelöst werden, da sie geeignete Ansätze bieten.

Seit Jahren werden Teile von Entwicklungsprojekten ausgelagert – auch an internationale Standorte. Der erhöhte Kommunikationsaufwand zwischen den Entwicklungsteams wird jedoch in einem globalen Kontext durch die unterschiedlichen Zeitzonen und kulturellen Unterschiede erschwert (Lee-Kelley und Sankey 2008; May und Carter 2001; Picot und Baumann 2007). Nachdem diese dezentralen und verteilten Teams hauptsächlich über elektronische Informations- und Kommunikationsmedien wie Telefon, E-Mail oder Videokonferenzen kommunizieren und sich koordinieren, werden sie auch als „virtuelle Teams“ bezeichnet (Anderson et al. 2007; Hertel et al. 2005; May und Carter 2001).

Auch die Skalierung von agilen Methoden in einem stark verteilten Kontext ist herausfordernd (Lee-Kelley und Sankey 2008; May und Carter 2001; Moe et al. 2016).

LeSS unterscheidet bei der verteilten Entwicklung, ob ein Entwicklungsteam verstreut ist oder ob es verschiedene Entwicklungsteams an mehreren Orten gibt. Ein Team soll bei LeSS immer an einem Ort sein. Verteilten Teams, welche selbst immer an einem Ort zusammensitzen, steht LeSS offen gegenüber und beschreibt den Ablauf der teamübergreifenden Meetings in einer Story, welche den Einsatz von elektronischen Hilfsmitteln wie eine Videoübertragung und geteilte Dokumente, die gemeinsam bearbeitet werden können, empfiehlt (The LeSS Company B.V. 2018e).

SAFe behandelt das Thema verteilte Teams insoweit, als dass es den Einsatz von elektronischen Hilfsmitteln wie Webcams oder Sofortnachrichtendienste empfiehlt. Dennoch wird die Nähe des Teams bzw. der Teams und die dadurch vereinfachte ständige Kommunikation betont und so soll auch das PI Planning Meeting prinzipiell ein „Face-to-Face“ Meeting sein und nur in Ausnahmefällen die genannten elektronischen Hilfsmittel verwendet werden (Scaled Agile, Inc. 2017f, 2017l, 2017m).

Der Scrum@Scale Guide erkennt die Notwendigkeit einer verteilten Entwicklung, um beispielsweise benötigte Expertise zu erreichen. Die Organisation bei Scrum@Scale soll organisch wachsen und kann ebenso wie das Skalierungsframework komponentenbasiert aufgebaut werden, wodurch eine verteilte Entwicklung möglich ist. Weiter Hinweise lassen sich jedoch nicht finden (Sutherland und Scrum Inc. 2018). Nexus macht keine Angaben zur Verteilung von Teams (Schwaber und Scrum.org 2018a). DA hingegen bindet die geographische Verteilung in die Komplexitätsfaktoren der Skalierung ein und diskutiert Vorteile, Risiken sowie neun Strategien, um die Risiken einer verteilten Entwicklung zu reduzieren, so z.B., dass Kernmitglieder am Projektbeginn zusammenkommen sollen oder dass regelmäßig integriert werden soll (Ambler 2014a, 2015a). Abschließend kann festgestellt werden, dass eine verteilte Entwicklung eine Herausforderung ist, jedoch durch die gemeinsamen Meetings und die Unterstützung

durch elektronische Hilfsmittel, welche von den Skalierungsframeworks vorgegeben werden, kann die Kommunikation zwischen den Teams verbessert werden.

Eine der größten Herausforderungen in einer skalierten Produktentwicklung ist die große Anzahl an Abhängigkeiten zwischen den Aktivitäten und Artefakten der Teams, welche als technische Abhängigkeiten bezeichnet werden können. Herausfordernd hierbei ist die Unvorhersehbarkeit der Abhängigkeiten, unterschiedliche Implementierungsprioritäten von abhängigen Komponenten, ständige Prioritätsänderungen der Teams oder unterschiedliche Releasezyklen. Die Ursprünge der technischen Abhängigkeiten liegen in einer hohen Planungsunsicherheit, Veränderung der Aufgabepriorisierung, eine ungenügende Wissensteilung, in der Codequalität oder in Herausforderungen während der Integration (Sekitoleko et al. 2014).

Leider werden die Abhängigkeiten im skalierten Umfeld oft nicht richtig behandelt und es werden nur wenige Praktiken implementiert, die darauf ausgerichtet sind, Abhängigkeiten zu finden, zu analysieren und anschließend zu managen (Moe et al. 2016). In einem Umfeld von Abhängigkeiten ist daher Koordination von großer Bedeutung, welche als „Managen der Abhängigkeiten zwischen Aktivitäten“ (Malone und Crowsten 1994, S. 90) definiert werden kann, denn sollten keine Abhängigkeiten vorhanden sein, ist auch keine Koordination notwendig (ebd.).

Bei Nexus ist die Koordination von Abhängigkeiten ein Kerngedanke, denn sowohl beim Product Backlog Refinement, beim Nexus Sprint Planning und im Nexus Daily Scrum sollen Abhängigkeiten identifiziert sowie entfernt oder minimiert werden. Weiterhin hat das Nexus Integration Team die Verantwortung Abhängigkeiten aufzudecken und den Entwicklungsteams Werkzeuge bereitzustellen und Praktiken zu erläutern, damit diese die Abhängigkeiten erkennen können. Gefundene Abhängigkeiten werden im Nexus Sprint Backlog visualisiert und dadurch transparent dargestellt (Schwaber und Scrum.org 2018a). Auch bei LeSS führen die Teams die Verfeinerung der Product Backlog Items durch, wodurch diese Items für den kommenden Sprint vorbereitet werden. Während dieser Aktivitäten sollen auch stark vernetzte Items identifiziert werden (The LeSS Company B.V. 2018f). Bei einer Entwicklung nach SAFe werden Abhängigkeiten durchgehend durch die ständige Kommunikation im Team und teamübergreifend behandelt. Ein besonderes Augenmerk auf Abhängigkeiten wird im Program Increment Planning gelegt, bei dem ein gemeinsamer Plan für das kommende Program Increment erstellt wird. Ein Ergebnis dieser Planung ist das Program Board, das aufkommende Lieferdaten, Meilensteine und auch die Abhängigkeiten zwischen den Teams aufzeigt (Scaled Agile, Inc. 2017f, 2017l). Auch bei DA gibt es ein physisches oder elektronisches Board, das die funktionalen und technischen Abhängigkeiten darstellt. Zudem werden die Abhängigkeiten zwischen agilen

Teams, zwischen agilen und traditionell arbeitenden Teams und zwischen agilen Teams und Teams, die Lean arbeiten, diskutiert und mögliche Strategien zum Umgang mit diesen Abhängigkeiten erläutert. Insbesondere der Product Owner und der Architecture Owner eines Disciplined Agile Teams spielen bei der Lösung von Abhängigkeiten eine zentrale Rolle (Ambler 2014b, 2014c, 2014d). Ein Modul des Scrum Master Cycle von Scrum@Scale ist die teamübergreifende Koordination. Ein Ergebnis dieses Moduls ist das Managen von teamübergreifenden Abhängigkeiten, damit diese nicht zu Hindernissen werden. Dafür wird im täglich stattfindenden Scaled Daily Scrum der Scrum Master die Frage nach neu identifizierten oder gelösten Abhängigkeiten gestellt. Zudem gibt es die Rolle des Scrum of Scrums Masters, welcher die vorhandenen Hindernisse priorisiert und dabei einen besonderen Fokus auf teamübergreifenden Abhängigkeiten legt (Sutherland und Scrum Inc. 2018).

Durch die Zusammenarbeit von unterschiedlichsten Disziplinen in einem Produkt treten weitere Herausforderungen wie die Synchronisation durch unterschiedlichste Entwicklungszyklen auf, da Software schneller und mit geringeren Kosten geändert werden kann als Hardware (cPrime. Inc. 2014; Edwards 2018; Lima et al. 2015; Müller et al. 2017).

Bei SAFe sollen die Teams eines Agile Release Trains eine gleiche Iterationsdauer vorweisen, damit am Ende die Inkremente integriert, getestet und Fehler früh gefunden werden können (Scaled Agile, Inc. 2017j). Mehrere Agile Release Trains können in der Large Solution Ebene in einem Solution Train zusammengefasst werden. Dadurch ergeben sich mehr Möglichkeiten, denn die gemeinsame Kadenz auf Solution Train Ebene ist das Program Increment, welches über alle Agile Release Trains die gleiche Dauer besitzen muss. Jeder Agile Release Train kann jedoch seine Sprintdauer innerhalb des Program Increments selbst festlegen. Dadurch können auch längere Entwicklungszyklen von unterschiedlichen Entwicklungsdisziplinen berücksichtigt werden, indem beispielsweise ein Agile Release Train für jede Disziplin eingerichtet wird, und dennoch auf Large Solution Ebene eine Synchronisation stattfindet (Scaled Agile, Inc. 2017k; Yakyma und Koehnemann 2016). Zudem erweitert SAFe reines Scrum um Mechanismen für eine agile Architektur. Dadurch wird eine grobe Architektur für den Agile Release Train bzw. Solution Train vom System Architect bzw. Solution Architect entworfen, um die Schnittstellen besser zu koordinieren ohne den Teams den Freiraum in der konkreten Umsetzung zu nehmen (Scaled Agile, Inc. 2017o). Nexus und LeSS geben keine Hinweise zu unterschiedlichen Entwicklungszyklen und Nexus konzentriert sich generell stark auf die Softwareentwicklung. Bei LeSS und LeSS Huge wird vorausgesetzt alle Teams die gleiche Sprintdauer mit gleichen Anfangs- und Enddaten einhalten, Nexus folgt

prinzipiell der gleichen Regel und nur wenn es absolut notwendig ist, können auch unterschiedliche Sprintlängen gewählt werden, jedoch mit einem gemeinsamen Nenner (Bourk und Kong 2016; The LeSS Company B.V. 2018h; Schwaber und Scrum.org 2018a). Scrum@Scale macht keine Angabe zur Sprintdauer, jedoch basiert Scrum@Scale auf einer empirischen Entscheidungsfindung, welche auf den Säulen Transparenz, Inspektion und Anpassung beruht. Dadurch kann eine kontinuierliche Verbesserung der Organisation angestrebt werden, wodurch auch die Sprintdauer entsprechend den Erfahrungen der Entwicklungsteams angepasst werden kann (Sutherland und Scrum Inc. 2018). Disciplined Agile Delivery (DAD) als Kernbereich von DA auf Entwicklungsebene wird als ein lernorientierter, hybrid-agiler Ansatz für die Lieferung von IT-Lösungen bezeichnet. DAD ergänzt Scrum um Methoden wie Agile Modeling (AM), Extreme Programming (XP), Kanban und viele weitere und beschreibt fünf Entwicklungszyklen, die den vollständigen End-to-End-Lebenszyklus abdecken (Lines und Ambler 2012). Dadurch sind die Teams in der Wahl freier und können auch einen leanen oder kontinuierlichen Ansatz ohne Iterationen verfolgen, je nachdem was für das Team am geeignetsten ist. DA lässt den Anwender daraufhin entscheiden, welchen Ansatz er zur Koordination der verschiedenen Teams verfolgt, ob nun durch gemeinsame Iterationsdauern, durch unterschiedliche Iterationslängen mit einem gemeinsamen Vielfachen oder durch unterschiedliche Iterationslängen, wodurch jedoch Abstimmungs-herausforderungen entstehen können (Ambler 2015b).

Bereits im agilen Manifest ist festgehalten, dass die Reaktion auf Veränderung für die Anhänger der agilen Vorgehensweise höher wertgeschätzt wird, als einem Plan zu folgen. Dieser Wert darf jedoch nicht mit „keiner Planung“ gleichgesetzt werden, denn auch bei einem agilen Entwicklungsprojekt wird geplant. Jedoch werden die Grenzen einer Planung in einem sich verändernden Markt erkannt und daher wird nicht wie bei plangetriebenen, klassischen Methoden zu Beginn ein weitreichender Plan erstellt und versucht alle Aufgaben bis ins Detail abzuschätzen und zu planen (Beck et al. 2001; Highsmith 2001; Smits 2008). Alle betrachteten Skalierungsframeworks gliedern daher ein tägliches Abstimmungs- und Planungsevent und ein Event zur Iterationsplanung ein. Die unterschiedlichen Ausprägungen dieser Planungs- und Koordinationsrunden wurden bereits beschrieben (Ambler 2015b; Lines und Ambler 2012; Scaled Agile, Inc. 2017e; Schwaber und Scrum.org 2018a; Sutherland und Scrum Inc. 2018; The LeSS Company B.V. 2018c). Da bei einer agilen Skalierung viele Teams und weitere Beteiligte zusammenarbeiten, ist eine rein iterationsbasierte Planung nicht ausreichend, da dadurch der Blick auf längerfristige Auswirkungen und auf das Gesamtprodukt verloren gehen (Scaled Agile, Inc. 2017h; Smits 2008).

Nexus und LeSS geben keine Hinweise zu einer Planung, welche über die Iterationen hinausgehen. LeSS verankert die Warteschlangentheorie als eines der Prinzipien, wodurch es auf mögliche Warteschlange in der Entwicklung hinweist und gleichzeitig die Länge der ausdefinierten Product Backlog Items verkürzt werden soll. Dadurch zeigt sich, dass bei Nexus und LeSS der Fokus auf kurzfristiger Planung liegt (Schwaber und Scrum.org 2018a; The LeSS Company B.V. 2018g). Scrum@Scale definiert das Modul Release Planning innerhalb des Product Owner Cycle, welches zum Ziel hat, die Lieferdaten der Kernfunktionen zu prognostizieren, Stakeholder über den Release Plan zu informieren und auf Basis des Stakeholder-Feedbacks den Release Plan und die Prioritäten anzupassen. Verantwortlich ist das MetaScrum-Team, welches aus den Product Ownern aller Teams besteht. Der Chief Product Owner, der das MetaScrum koordiniert und die Erstellung eines priorisierten Product Backlogs über alle Teams verantwortet, arbeitet eng mit dem Scrum of Scrum Master zusammen, damit die Teams den Release Plan effizient umsetzen können. Über welchen Zeitraum der Release Plan Daten angibt, wird nichts beschrieben (Sutherland und Scrum Inc. 2018). Konkreter wird SAFe, welches eine Roadmap in die Konfigurationen Large Solution SAFe, Portfolio SAFe und Full SAFe eingliedert. Diese Roadmap spannt sich über einen Zeitraum von drei Program Increments bzw. ca. 30 Wochen auf. Sie enthält aufkommende Meilensteine, die vereinbarten Funktionalitäten des aktuellen Program Increments und gibt einen Überblick über mögliche Inhalte der zwei folgenden Program Increments. Betont wird, dass es sich bei der Roadmap nicht um eine weitreichende Liste abzuarbeitender Features handelt, sondern nur das aktuelle Program Increment verbindlich abgestimmt ist. Bei den beiden nachfolgenden Program Increments handelt es sich lediglich um eine Prognose, bei welcher Änderungen noch stattfinden können. Dennoch bietet diese Grobplanung einen besseren Überblick über das Projekt und lässt gleichzeitig die Reaktion auf Veränderung zu. Ergänzt wird die Roadmap durch eine langfristige Schätzung der Aufgaben, eine grobe Ressourcenallokation und Metriken wie die Agile Release Train Velocity (Scaled Agile, Inc. 2017h, 2017c). Releases können am Ende jedes Program Increments oder entkoppelt von den synchronisierten Entwicklungsiterationen und Program Increments stattfinden. SAFe bietet dazu die Möglichkeit häufiger während der Program Increments, seltener oder nach Bedarf zu releasen. Zudem kann sich die Releasestrategie innerhalb eines Projekts bei den Produktanteilen oder den Entwicklungsebenen unterscheiden (Scaled Agile, Inc. 2017j). Auch DA bietet innerhalb Disciplined DevOps Auswahlpunkte für den Bereich des Release Managements, das den unternehmensweiten Release Plan und die zugrundeliegende Strategie bestimmt. Zusätzlich zu den Strategien von SAFe schlägt DA noch vor, Zeiträume zu identifizieren, in denen Releases freigegeben werden können und in denen keine Releases stattfinden dürfen. Zudem sollen Releaseabhängigkeiten

gemanagt und der Release Plan kommuniziert werden. Weiterhin muss die Häufigkeit der Releases der Teams festgelegt werden, weshalb sich dieser Entscheidungspunkt auch im Program Management innerhalb Disciplined Agile Delivery wiederfindet. Dabei besteht die Wahl aus drei Releasestrategien – eine kontinuierliche Auslieferung, eine inkrementelle Auslieferung oder eine unregelmäßige Auslieferung. Zudem wird durch das Program Management das Programm geplant. Auch hier besteht die Möglichkeit einer jährlichen Planung, einer Ad-hoc Planung oder einer Rolling-Wave Planung, welche von DA präferiert wird (Ambler 2015b, 2015c). Diese Präferenz gilt auch für das Erstellen einer Business und / oder Produkt Roadmap durch das Product Management. Bei diesem Ansatz der Rolling-Wave Planung werden Funktionalitäten, welche zeitnah ausgeliefert werden, sehr detailliert geplant und zukünftige Aktivitäten und Releases nur grob und auf einem hohen Level geplant. Generell empfiehlt DA, dass Produkte strategisch auf lange Sicht geplant, jedoch taktisch kurzfristig implementiert werden sollten (Ambler 2015d).

FAZIT

Die eingangs gestellten Kernfragen:

- Welche Herausforderungen gibt es in Entwicklungsprojekten?
- Welche können durch agile Skalierungsframeworks gelöst werden?
- Wie lösen agile Skalierungsframeworks diese Herausforderungen?

können wie folgt beantwortet werden.

Die Herausforderungen in aktuellen Entwicklungsprojekten sind bestimmt durch interne und externe Faktoren. Die externen Herausforderungen sind geprägt durch Marktanforderungen und die Regulierungen, welche für die jeweilige Branche gelten. Die internen Herausforderungen ergeben sich vor allem durch die notwendige Zusammenarbeit von mehreren und/oder interdisziplinären Teams in großen Entwicklungsprojekten.

Wie im Hauptteil erläutert, lösen die agilen Skalierungsframeworks nicht direkt die externen Herausforderungen, die durch einen volatilen Markt, neue Technologien und die zu berücksichtigenden Regulierungen entstehen. Jedoch können Unternehmen die agilen Skalierungsframeworks verwenden, um auf die beschriebenen Herausforderungen deutlich besser reagieren zu können.

Die aufgezeigten internen Herausforderungen der Zusammenarbeit in großen Entwicklungsprojekten wie Effizienz, Synchronisation, Komplexitätsbeherrschung, und Abhängigkeiten, denen die interdisziplinären Teams begegnen, können hingegen durch agile Skalierungsframeworks direkt verbessert werden.

Zentrale Elemente, die dies sicherstellen, sind die jeweiligen Rollen, Artefakte und Events, welche die unterschiedlichen Skalierungsframeworks beschreiben.

Abschließend ist kritisch anzumerken, dass der Einsatz eines agilen Skalierungsframeworks voraussetzt, dass die einzelnen Teams bereits die grundlegenden agilen Techniken wie Scrum anwenden, da dies in den meisten Fällen die Basis für die Skalierung ist. Somit müssen Unternehmen im ersten Schritt die grundlegenden agilen Ansätze und Methoden teamübergreifend einführen, um im zweiten Schritt die Verbesserungspotentiale durch den Einsatz agiler Frameworks heben zu können.

LITERATURVERZEICHNIS

- Adelberger, Werner; Haft-Zboril, Nicole (2015): Systematischer Ansatz zur projekthaften Steuerung von Entwicklungskosten. In: *Controlling* 27 (1), S. 49–56.
- agile42 (2018): agile42 Enterprise Transition Framework. Online verfügbar unter <http://www.agile42.com/en/agile-transition/etf/>, zuletzt geprüft am 16.05.2017.
- Allesch, Raimund; Hosp, Josef; Prostednik, Daniel; Steiner, Martin (2009): EMV und funktionale Sicherheit. In: *e & i Elektrotechnik und Informationstechnik* 126 (6), 21–26. DOI: 10.1007/s00502-009-0649-z.
- Alqudah, Mashal; Razali, Rozilawati (2016): A Review of Scaling Agile Methods in Large Software Development. In: *International Journal on Advanced Science Engineering Information Technology* 6 (6), S. 828–837. Online verfügbar unter https://www.researchgate.net/profile/Mashal_Alqudah/publication/311916796_A_Review_of_Scaling_Agile_Methods_in_Large_Software_Development/links/58bc5de692851c471d5638f3/A-Review-of-Scaling-Agile-Methods-in-Large-Software-Development.pdf, zuletzt geprüft am 06.09.2017.
- Ambler, Scott (2013a): Agile and Regulatory Compliance. Online verfügbar unter <http://www.disciplinedagiledelivery.com/regulatorycompliance/>, zuletzt aktualisiert am 09.10.2013, zuletzt geprüft am 29.03.2018.
- Ambler, Scott (2014a): Geographically Distributed Agile Teams. Online verfügbar unter <http://www.disciplinedagiledelivery.com/agility-at-scale/geographically-distributed-agile-teams/>, zuletzt aktualisiert am 16.06.2016, zuletzt geprüft am 07.04.2018.
- Ambler, Scott (2014b): Managing Requirements Dependencies Between Agile Teams. Online verfügbar unter <http://www.disciplinedagiledelivery.com/managing-requirements-dependencies-between-agile-teams/>, zuletzt geprüft am 08.04.2018.
- Ambler, Scott (2014c): Managing Requirements Dependencies Between Agile and Lean Teams. Online verfügbar unter <http://www.disciplinedagiledelivery.com/managing-requirements-dependencies-between-agile-and-lean-teams/>, zuletzt geprüft am 08.04.2018.
- Ambler, Scott (2014d): Managing Requirements Dependencies Between Agile/Lean Teams and Traditional/Waterfall Teams. Online verfügbar unter <http://www.disciplinedagiledelivery.com/dependencies-agile-traditional/>, zuletzt geprüft am 08.04.2018.
- Ambler, Scott (2015a): Tactical Agility at Scale. Online verfügbar unter <http://www.disciplinedagiledelivery.com/agility-at-scale/tactical-agility-at-scale/>, zuletzt aktualisiert am 21.03.2018, zuletzt geprüft am 07.04.2018.
- Ambler, Scott (2015b): Program Management. Online verfügbar unter <http://www.disciplinedagiledelivery.com/agility-at-scale/program-management/>, zuletzt aktualisiert am 24.10.2016, zuletzt geprüft am 08.04.2018.
- Ambler, Scott (2015c): Release Management. Online verfügbar unter <http://www.disciplinedagiledelivery.com/release-management/>, zuletzt aktualisiert am 16.06.2017, zuletzt geprüft am 10.04.2018.
- Ambler, Scott (2015d): Product Management. Online verfügbar unter <http://www.disciplinedagiledelivery.com/agility-at-scale/product-management/>, zuletzt aktualisiert am 05.01.2018, zuletzt geprüft am 10.04.2018.
- Ambler, Scott W. (2013b): Enterprise Unified Process (EUP): Strategies for Enterprise Agile. Amblysoft Inc. Online verfügbar unter <http://www.enterpriseunifiedprocess.com/>, zuletzt aktualisiert am 28.01.2018, zuletzt geprüft am 28.03.18.
- Ambler, Scott W.; Lines, Mark (2012): *Disciplined Agile Delivery. A Practitioner's Guide to Agile Software Delivery in the Enterprise.* Foreword by Dave West. Upper Saddle River, N.J., London: IBM Press.
- Ambler, Scott W.; Lines, Mark (2017): *An Executive's Guide to Disciplined Agile: Winning the Race to Business Agility.* Winning the Race to Business Agility: CreateSpace Independent Publishing Platform.
- Anderson, A. H.; McEwan, R.; Bal, J.; Carletta, J. (2007): Virtual team meetings. An analysis of communication and context. In: *Computers in Human Behavior* 23 (5), S. 2558–2580. DOI: 10.1016/j.chb.2007.01.001.
- Arnold, Woody (2015): What's the difference between the Nexus Framework and other Scaled Scrum Frameworks? Online verfügbar unter <https://www.quora.com/Whats-the-difference-between-the-Nexus-Framework-and-other-Scaled-Scrum-Frameworks>, zuletzt geprüft am 11.04.2018.
- Artelt, Dirk; Leonhard, Sebastian (2017): F&E zwischen Kreativität und Wirtschaftlichkeit. In: *Controlling & Management Review* 61 (3), S. 8–15.
- Beck, Kent; Beedle, Mike; van Bennekum, Arie; Cockburn, Alistair; Cunningham, Ward; Fowler, Martin et al. (2001): *Manifesto for Agile Software Development.* Online verfügbar unter <http://agilemanifesto.org/>, zuletzt aktualisiert am 01.09.2016, zuletzt geprüft am 01.08.2017.
- Beck, Peter; Gärtner, Markus; Mathis, Christoph; Roock, Stefan; Schliep, Andras (o. J.): *ScALeD Agile Lean Development – Die Prinzipien.* Online verfügbar unter <http://scaledprinciples.org/de/>, zuletzt geprüft am 05.10.2017.
- Beedle, Mike; Enterprise Scrum Inc (2010): *Enterprise Scrum.* Enterprise Scrum Inc. Online verfügbar unter <http://www.enterprisescrum.com/enterprise-scrum/>, zuletzt geprüft am 30.05.2017.
- Bennett, Nathan; Lemoine, G. James (2014): What a difference a word makes: Understanding threats to performance in a VUCA world. In: *Business Horizons* 57 (3), S. 311–317.
- Bergen, Katja von (2016): *Projekt-Management: Agil ja - aber nicht um jeden Preis.* Online verfügbar unter <https://www.computerwoche.de/a/agil-ja-aber-nicht-um-jeden-preis,3325568>, zuletzt aktualisiert am 25.10.2016, zuletzt geprüft am 24.08.2017.
- Bourk, Simon; Kong, Patricia (2016): An Introduction to the Nexus™ Framework. In: *Scrum.org Whitepapers.* Online verfügbar unter https://s3.amazonaws.com/scrumorg-website-prod/drupal/2016-06/An%20Introduction%20to%20the%20Nexus%20Framework%20-%20June%202016_0.pdf, zuletzt geprüft am 13.04.2017.
- Brenner, Richard; Wunder, Stefan (2015): Scaled Agile Framework: Presentation and real world example. In: 2015 IEEE Eighth International Conference on Software Testing, Verification, and Validation Workshops (ICSTW). Proceedings : April 13-17, 2015, Graz, Austria. 2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW). Graz, Austria. [Piscataway, New Jersey]: IEEE, S. 1–2.

- Breu, Leo (2018): Wie Investments keine eigene "Kohle" kosten. In: *Produktion - Technik und Wirtschaft für die deutsche Industrie 57* (5), S. 6–7.
- cPrime. Inc. (2014): Agile Processes for Hardware Development. Unter Mitarbeit von Kevin Thompson, John Carter und Scott Elliott. Online verfügbar unter https://www.cprime.com/wp-content/uploads/woocommerce_uploads/2015/10/Agile-Processes-for-Hardware-Development-cPrime.pdf, zuletzt geprüft am 08.04.2018.
- Cron Technologies LLC (2017): FAST Agile™ Scaled Technology. Home. Online verfügbar unter <http://www.fast-agile.com/>.
- Czudek, Sacha (2015): Agile Modelle im Vergleich: Wo passt welches? SwissQ Consulting AG. Online verfügbar unter <http://swissq.it/de/agile/unternehmensweite-agilitaet-ein-muss/>, zuletzt geprüft am 31.05.2017.
- Daut, Patrick (2015): agile@scale: Do more with LeSS or be SAFe? Ansätze zur Skalierung - ein Überblick. In: Martin Engstler, Masud Fazal-Baqaie, Eckhart Hanser, Martin Mikusz und Alexander Volland (Hg.): *Projektmanagement und Vorgehensmodelle 2015. Hybride Projektstrukturen erfolgreich umsetzen. Gemeinsame Tagung der Fachgruppen Projektmanagement (WI-PM) und Vorgehensmodelle (WI-VM) im Fachgebiet Wirtschaftsinformatik der Gesellschaft für Informatik e.V. Elmshorn, 22.10.2015-23.10.2017. Gesellschaft für Informatik e.V. (GI). Bonn: Gesellschaft für Informatik e.V. (GI) (Lecture Notes in Informatics (LNI) Proceedings, Volume P-250)*, S. 159–167. Online verfügbar unter <https://subs.emis.de/LNI/Proceedings/Proceedings250/159.pdf>, zuletzt geprüft am 05.10.2017.
- Daut, Patrick (2016): Scaled Agile Framework - eine Entscheidungshilfe. Agil im Großen: Das richtige Framework für Sie. In: *Projekt Magazin* 14. Online verfügbar unter https://www.projektmagazin.de/artikel/agil-im-grossen-das-richtige-framework-fuer-sie_1111385, zuletzt geprüft am 19.05.2017.
- Disciplined Agile Consortium (2017): Disciplined Agile Consortium. Online verfügbar unter <https://disciplinedagiledelivery.wildapricot.org/>.
- Disciplined Agile Consortium (2018): Case Studies. Online verfügbar unter <https://disciplinedagileconsortium.org/Disciplined-Agile-Case-Study>, zuletzt geprüft am 29.03.2018.
- Dvir, Dov; Lipovetsky, Stan; Shenhar, Aaron; Tishler, Asher (1998): In search of project classification: a non-universal approach to project success factors. In: *Research policy* 27 (9), S. 915–935.
- Edwards, Dibbe (2018): Connecting hardware and software lifecycles to build the Internet of Things. Online verfügbar unter <https://www.ibm.com/blogs/internet-of-things/hardware-software-lifecycle/>, zuletzt geprüft am 08.04.2018.
- Etezadzadeh, Chirine (2016): Digitalization. In: Chirine Etezadzadeh (Hg.): *Smart City – Future City?* Wiesbaden: Springer Fachmedien Wiesbaden (essentials), S. 37–45.
- Francino, Yvette (2016): Large-scale agile frameworks compared: SAFe vs DAD. Online verfügbar unter <https://techbeacon.com/large-scale-agile-frameworks-compared-safe-vs-dad>, zuletzt aktualisiert am 16.10.2017, zuletzt geprüft am 16.10.2017.
- Furuhjelm, Jörgen; Segertoft, Johan; Justice, Joe; Sutherland, J. J. (2017): *Owning the Sky with Agile*. Scrum Inc.; Saab Aeronautics. Online verfügbar unter https://www.scruminc.com/wp-content/uploads/2017/02/Release-version_Owning-the-Sky-with-Agile.pdf.
- Gelec, Erdem; Wagner, Frank (2014): Future Trends and key challenges in R&D Management - Results of an empirical study within industrial R&D in Germany. In: *The R&D Management Conference. Management of Applied R&D. Connecting High Value Solutions with Future Markets. Proceedings. R&D Management Conference. Stuttgart, 03.06.2014-06.06.2014: Fraunhofer IAO; IAT Universität Stuttgart* (6), S. 920–926.
- Haller, Dominik (2014): „Agiles Projektmanagement – Projektentwicklung mit Scrum, Kanban & Co.“ – das Whitepaper zum Download. Online verfügbar unter <https://www.techdivision.com/blog/agiles-projektmanagement-projektentwicklung-mit-scrum-kanban-co-das-whitepaper-zum-download/>, zuletzt aktualisiert am 23.06.2014, zuletzt geprüft am 08.10.2017.
- Hanna, Rick; Kuhnert, Felix (2014): How to be No. 1. Facing future challenges in the automotive industry. PwC. Online verfügbar unter <http://www.pwc.com/gx/en/automotive/industry-publications-and-thought-leadership/assets/pwc-how-to-be-no-1-facing-future-challenges-in-the-automotive-industry.pdf>, zuletzt geprüft am 31.05.2017.
- Hastie, Shane (2014): Examining Different Approaches to Scaling Agile. Online verfügbar unter <https://www.infoq.com/news/2014/07/compare-agile-scaling>, zuletzt aktualisiert am 08.07.2014, zuletzt geprüft am 25.07.2017.
- Heikkilä, Ville T.; Paasivaara, Maria; Lassenius, Casper; Engblom, Christian (2013): Continuous Release Planning in a Large-Scale Scrum Development Organization at Ericsson. In: Wil van der Aalst, John Mylopoulos, Michael Rosemann, Michael J. Shaw, Clemens Szyperski, Hubert Baumeister und Barbara Weber (Hg.): *Agile processes in software engineering and extreme programming. 14th International Conference, XP 2013, Vienna, Austria, June 3-7, 2013. Proceedings, Bd. 149. 1st edition. New York: Springer (Lecture Notes in Business Information Processing, 149)*, S. 195–209, zuletzt geprüft am 25.04.2017.
- Hertel, Guido; Geister, Susanne; Konradt, Udo (2005): Managing virtual teams. A review of current empirical research. In: *Human Resource Management Review* 15 (1), S. 69–95. DOI: 10.1016/j.hmr.2005.01.002.
- Heusser, Matt (2014): Introducing the SCARE Method. Online verfügbar unter <http://itknowledgeexchange.techtarget.com/uncharted-waters/introducing-the-scare-method/>, zuletzt aktualisiert am 11.08.2014, zuletzt geprüft am 28.03.2018.
- Heusser, Matt (2015): Comparing scaling agile frameworks. CIO. Online verfügbar unter <http://www.cio.com/article/2974436/agile-development/comparing-scaling-agile-frameworks.html>, zuletzt aktualisiert am 21.08.2015, zuletzt geprüft am 31.05.2017.
- Highsmith, Jim (2001): *History: The Agile Manifesto*. Agile Alliance. Online verfügbar unter <http://agilemanifesto.org/history.html>, zuletzt aktualisiert am 01.09.2016, zuletzt geprüft am 11.08.2017.
- Hillenbrand, Martin (2012): Funktionale Sicherheit nach ISO 26262 in der Konzeptphase der Entwicklung von Elektrik/Elektronik Architekturen von Fahrzeugen: Universität Karlsruhe Universitätsbibliothek. Online verfügbar unter <https://books.google.de/books?id=IsUxPfoqjRoC>.
- Howard, Annie (2017): Case Study: Bosch Embracing Agility. Online verfügbar unter <https://www.scrumatscale.com/wp-content/uploads/Annie-Howard-Bosch-Slides.pdf>, zuletzt geprüft am 29.03.2018.
- it-agile GmbH (o. J.): *Agile Skalierung über die Prinzipien*. Online verfügbar unter <https://www.it-agile.de/wissen/agile-skalierung-ueber-die-prinzipien/>, zuletzt geprüft am 30.05.2017.
- Johansen, Agnar; Eik-Andresen, Petter; Dypvik Landmark, Andreas; Ekambaram, Anandasivakumar; Rolstadås, Asbjørn (2016): Value of Uncertainty: The Lost Opportunities in Large Projects. In: *Administrative Sciences* 6 (3), S. 11. DOI: 10.3390/admsci6030011.

- Kittel, Kai (2013): Agilität von Geschäftsprozessen trotz Compliance. In: *Wirtschaftsinformatik*, S. 967–981, zuletzt geprüft am 29.03.2018.
- Kniberg, Henrik; Ivarsson, Anders (2012): Scaling Agile @ Spotify. with Tribes, Squads, Chapters & Guilds. Online verfügbar unter <https://dl.dropboxusercontent.com/u/1018963/Articles/SpotifyScaling.pdf>, zuletzt geprüft am 20.04.2017.
- Komus, Ayelt (2017): Abschlussbericht: Status Quo Agile 2016/2017. 3. Studie über Erfolg und Anwendungsformen von agilen Methoden. Unter Mitarbeit von Scrum.org und GPM - Deutsche Gesellschaft für Projektmanagement e.V. Hochschule Koblenz University of Applied Sciences. Online verfügbar unter https://www.hs-koblenz.de/fileadmin/media/fb_wirtschaftswissenschaften/Forschung_Projekte/Forschungsprojekte/Status_Quo_Agile/Studie_2016/SQA-Abschlussbericht-de_V.1.01_-_Interessenten.pdf, zuletzt geprüft am 20.06.2017.
- Laanti, Maarit (2014): Characteristics and Principles of Scaled Agile. In: Torgeir Dingsøy, Nils Brede Moe, Roberto Tonelli, Steve Counsell, Cigdem Gencel und Kai Petersen (Hg.): *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation*, Bd. 199. Cham: Springer International Publishing (Lecture Notes in Business Information Processing), S. 9–20.
- Lachmann & Rink (o. J.): Iterative Entwicklung. Online verfügbar unter <https://www.lachmann-rink.de/index.php/iterativ-entwickeln.html>, zuletzt geprüft am 08.10.2017.
- Laing, Sam (2013): SLIM - an agile scaling pattern. SLIM – Scrum Lean In Motion framework. Online verfügbar unter <https://www.growingagile.co.za/2013/08/slim-an-agile-scaling-pattern/>, zuletzt aktualisiert am 29.08.2013.
- Larman, Craig; Vodde, Bas; Jensen, Björn; Marquart, Alexander (2017): *Large-Scale Scrum. Scrum erfolgreich skalieren mit LeSS*. Heidelberg: dpunkt.verlag.
- LeadingAgile (2018): Considering Agile? Let us show you how to safely and pragmatically introduce agile into any size organization. Online verfügbar unter <https://www.leadingagile.com/>, zuletzt aktualisiert am 2018.
- Lee-Kelley, Liz; Sankey, Tim (2008): Global virtual teams for value creation and project success. A case study. In: *International Journal of Project Management* 26 (1), S. 51–62. DOI: 10.1016/j.ijproman.2007.08.010.
- Lima, Glaydson Luis Bertoze; Ferreira, Guilherme, Augusto Lopes; Saotome, Osamu; Cunha, Adilson Marques da; Dias, Luis Alberto Vieira (2015): Hardware Development: Agile and Co-Design. 2015 12th International Conference on Information Technology - New Generations. In: Shahram Latifi (Hg.): 2015 12th International Conference on Information Technology - New Generations (ITNG). 13 - 15 April 2015, Las Vegas, Nevada, USA. International Conference on Information Technology - New Generations; ITNG. Piscataway, NJ: IEEE.
- Linders, Ben (2016): Benefits of Agile Transformation at Barclays. Online verfügbar unter <https://disciplinedagileconsortium.org/resources/Pictures/Case%20studies/Benefits%20of%20Agile%20Transformation%20at%20Barclays.pdf>, zuletzt geprüft am 29.03.2018.
- Lines, Mark; Ambler, Scott (2012): Introduction to Disciplined Agile Delivery (DAD). Disciplined Agile Consortium. Online verfügbar unter <http://www.disciplinedagiledelivery.com/introduction-to-dad/>, zuletzt aktualisiert am 02.08.2017, zuletzt geprüft am 07.08.2017.
- Lines, Mark; Ambler, Scott (2017a): The Disciplined Agile (DA) Framework. A Foundation for Business Agility. Home. Disciplined Agile Consortium. Online verfügbar unter <http://www.disciplinedagiledelivery.com/>, zuletzt geprüft am 07.08.2017.
- Lines, Mark; Ambler, Scott (2017b): The Disciplined Agile Enterprise (DAE). Disciplined Agile Consortium. Online verfügbar unter <http://www.disciplinedagiledelivery.com/dae/>, zuletzt geprüft am 07.08.2017.
- Lines, Mark; Ambler, Scott (2017c): The Disciplined Agile (DA) Framework. Disciplined Agile Consortium. Online verfügbar unter <http://www.disciplinedagiledelivery.com/agility-at-scale/disciplined-agile-2/>, zuletzt aktualisiert am 05.10.2017, zuletzt geprüft am 21.10.2017.
- Lines, Mark; Ambler, Scott (2018): Continuous Improvement. Online verfügbar unter <http://www.disciplinedagiledelivery.com/agility-at-scale/continuous-improvement/>, zuletzt aktualisiert am 22.01.2018, zuletzt geprüft am 29.03.2018.
- Malone, Thomas; Crowsten, Kevin (1994): The Interdisciplinary Study of Coordination. In: *ACM Computing Surveys (CSUR)* 26 (1), S. 87–119. Online verfügbar unter <https://dspace.mit.edu/bitstream/handle/1721.1/2500/SWP-3630-31204001-CCS-157.pdf?sequence=1>, zuletzt geprüft am 10.08.2017.
- Maruping, Likoebe M.; Venkatesh, Viswanath; Agarwal, Ritu (2009): A Control Theory Perspective on Agile Methodology Use and Changing User Requirements. In: *Information Systems Research* 20 (3), S. 377–399. DOI: 10.1287/isre.1090.0238.
- Matharu, Gurpreet Singh; Mishra, Anju; Singh, Harmeet; Upadhyay, Priyanka (2015): Empirical Study of Agile Software Development Methodologies. In: *SIGSOFT Softw. Eng. Notes* 40 (1), S. 1–6. DOI: 10.1145/2693208.2693233.
- Mathis, Christoph (2016): SAFe. Das Scaled Agile Framework: Lean und Agile in großen Unternehmen skalieren. Unter Mitarbeit von Dean Leffingwell. 1. Auflage. Heidelberg: dpunkt.verlag.
- Mathis, Christoph; Fish, Stuart (2015): *Scaling Frameworks*. imprevu. München, 27.01.2015, zuletzt geprüft am 31.05.2017.
- Maxos (2016): Continuous Enterprise. Matrix of Services. Maxos. Online verfügbar unter http://www.continuousagile.com/unblock/ea_matrix.html, zuletzt aktualisiert am 19.12.2016.
- May, Andrew; Carter, Chris (2001): A case study of virtual team working in the European automotive industry. In: *International Journal of Industrial Ergonomics* 27 (3), S. 171–186. DOI: 10.1016/S0169-8141(00)00048-2.
- Merel, Peter (2016): XSCALE. Blueprint for an Agile Organization. Online verfügbar unter <http://xscale.wiki/#Solution%20Constraint>, zuletzt aktualisiert am 04.09.2017, zuletzt geprüft am 28.03.2018.
- Miller, Stefan (2006): Rückwärts- und Vorwärtsgerichtete Verfolgung von Fehlern für die modellbasierte Entwicklung eingebetteter Systeme. In: *GI Jahrestagung* (2), S. 210–214.
- Moe, Nils Brede; Olsson, Helena Holmström; Dingsøy, Torgeir (2016): Trends in Large-Scale Agile Development: A Summary of the 4th Workshop at XP2016. In: Peggy Gregory und Katie Taylor (Hg.): *Proceedings of the XP2016 Scientific Workshops*. Edinburgh, Scotland, May 24, 2016. the Scientific Workshop XP2016. Edinburgh, Scotland, UK, 24.05.2016 - 24.05.2016. New York, NY, USA: ACM (ICPS: ACM international conference proceeding series), S. 1–4, zuletzt geprüft am 07.07.2017.
- Morin, Ola (2015): Get your big organization up to speed. The Scaling Agile Model. In: *Scalare Magazine* (1), S. 7–9, zuletzt geprüft am 13.07.2017.
- Müller, Markus; Hörmann, Klaus; Dittmann, Lars; Zimmer, Jörg (2017): Stufenweise. Automotive SPICE: Agilität bei Autobauern. In: *iX Special. Magazin für professionelle Informationstechnik - Sonderheft*. iX. Hannover: Heise, Christian; Heise, Ansgar, 96-100.

- Nerur, Sridhar; Mahapatra, RadhaKanta; Mangalaraj, George (2005): Challenges of migrating to agile methodologies. In: *Commun. ACM* 48 (5), S. 72–78. DOI: 10.1145/1060710.1060712.
- Niederer, H. (2003): Von der Idee zur Produktion. Contributor-Konzept. In: *MegaLink* 10, S. 28–29.
- Peters, Steven; Chun, Jung-Hoon; Lanza, Gisela (2016): Digitalization of automotive industry – scenarios for future manufacturing. In: *Manufacturing Rev.* 3, S. 1. DOI: 10.1051/mfreview/2015030.
- Picot, Arnold; Baumann, Oliver (2007): Modularität in der verteilten Entwicklung komplexer Systeme: Chancen, Grenzen, Implikationen. In: *Journal für Betriebswirtschaft* 57 (3), S. 221–246. DOI: 10.1007/s11301-007-0029-8.
- Prowareness GmbH (2015): Agil skalieren. Online verfügbar unter <https://www.scrum.de/agil-skalieren/>, zuletzt aktualisiert am 28.09.2015, zuletzt geprüft am 30.05.2017.
- Rahimian, Vahid; Ramsin, Raman (2008): Designing an Agile Methodology for Mobile Software Development: A Hybrid Method Engineering Approach. In: Colette Rolland, Martine Collard, Oscar Pastor, André Flory und Jean Louis Cavarero (Hg.): *Research Challenges in Information Science. The Second International Conference on. RCIS 2008. Marrakesch, Marokko, 03.06.2008-06.06.2008*: IEEE, S. 337–342. Online verfügbar unter http://sharif.edu/~ramsin/index_files/Publications_PDF/Rahimian_Ramsin_RCIS_2008_Methodology_Mobile_Software_Hybrid.pdf, zuletzt geprüft am 17.10.2017.
- Rook, Stefan (2014): Agile Skalierung. Was Sie als Führungskraft wissen müssen. Version 2. Online verfügbar unter https://www.it-agile.de/fileadmin/docs/Whitepaper/Agile_Skalierung_f%C3%BCr_F%C3%BChrungskr%C3%A4fte_V2.pdf, zuletzt geprüft am 28.03.2018.
- Rook, Stefan; Wolf, Henning (2016): *Scrum - verstehen und erfolgreich einsetzen*. 1. Aufl. Heidelberg, Neckar: dpunkt.verlag (it-agile).
- Scaled Agile, Inc. (2017a): Scaled Agile Framework – SAFe for Lean Enterprises. Online verfügbar unter <http://www.scaledagileframework.com/#>, zuletzt geprüft am 04.08.2017.
- Scaled Agile, Inc. (2017b): What is SAFe? Online verfügbar unter <http://www.scaledagileframework.com/what-is-safe/>, zuletzt aktualisiert am 21.06.2017, zuletzt geprüft am 04.08.2017.
- Scaled Agile, Inc. (2017c): What's new in SAFe 4.5? Online verfügbar unter <http://www.scaledagileframework.com/whats-new-in-safe-45/>, zuletzt aktualisiert am 22.06.2017, zuletzt geprüft am 04.08.2017.
- Scaled Agile, Inc. (2017d): Why do Businesses Need SAFe? Online verfügbar unter <http://www.scaledagileframework.com/why-safe/>, zuletzt geprüft am 17.10.2017.
- Scaled Agile, Inc. (2017e): Iterations. Online verfügbar unter <http://www.scaledagileframework.com/iterations/>, zuletzt aktualisiert am 08.06.2017, zuletzt geprüft am 09.08.2017.
- Scaled Agile, Inc. (2017f): PI Planning. Online verfügbar unter <http://www.scaledagileframework.com/pi-planning/>, zuletzt aktualisiert am 14.07.2017, zuletzt geprüft am 09.08.2017.
- Scaled Agile, Inc. (2017g): Innovation and Planning Iteration. Online verfügbar unter <http://www.scaledagileframework.com/innovation-and-planning-iteration/>, zuletzt aktualisiert am 17.07.2017, zuletzt geprüft am 16.10.2017.
- Scaled Agile, Inc. (2017h): Roadmap. Online verfügbar unter <http://www.scaledagileframework.com/roadmap/>, zuletzt aktualisiert am 19.07.2017, zuletzt geprüft am 21.10.2017.
- Scaled Agile, Inc. (2017i): Achieving Regulatory and Industry Standards Compliance with the Scaled Agile Framework (SAFe). A Scaled Agile, Inc. White Paper. Online verfügbar unter https://www.scaledagileframework.com/wp-content/uploads/delightful-downloads/2017/10/White_Paper_Compliance_04-08-17-1.pdf, zuletzt geprüft am 29.03.2017.
- Scaled Agile, Inc. (2017j): Agile Release Train. Online verfügbar unter <http://www.scaledagileframework.com/agile-release-train/>, zuletzt aktualisiert am 12.09.2017, zuletzt geprüft am 15.10.2017.
- Scaled Agile, Inc. (2017k): Solution Train. Online verfügbar unter <http://www.scaledagileframework.com/solution-train/>, zuletzt aktualisiert am 16.09.2017, zuletzt geprüft am 19.10.2017.
- Scaled Agile, Inc. (2017l): Agile Teams. Online verfügbar unter <http://www.scaledagileframework.com/agile-teams/>, zuletzt aktualisiert am 19.09.2017, zuletzt geprüft am 16.10.2017.
- Scaled Agile, Inc. (2017m): Iteration Execution. Online verfügbar unter <http://www.scaledagileframework.com/iteration-execution/>, zuletzt aktualisiert am 26.09.2017, zuletzt geprüft am 17.10.2017.
- Scaled Agile, Inc. (2017n): Program Increment. Online verfügbar unter <http://www.scaledagileframework.com/program-increment/>, zuletzt aktualisiert am 26.09.2017, zuletzt geprüft am 15.10.2017.
- Scaled Agile, Inc. (2017o): Architectural Runway. Online verfügbar unter <http://www.scaledagileframework.com/architectural-runway/>, zuletzt aktualisiert am 11.10.2017, zuletzt geprüft am 16.10.2017.
- Scaled Agile, Inc. (2017p): Customer. Online verfügbar unter <http://www.scaledagileframework.com/customer/>, zuletzt aktualisiert am 12.10.2017, zuletzt geprüft am 15.10.2017.
- Scaled Agile, Inc. (2017q): Core Values. Online verfügbar unter <http://www.scaledagileframework.com/safe-core-values/>, zuletzt aktualisiert am 21.10.2017, zuletzt geprüft am 23.10.2017.
- Scaled Agile, Inc. (2017r): Communities of Practice. Online verfügbar unter <http://www.scaledagileframework.com/communities-of-practice/>, zuletzt aktualisiert am 25.10.2017, zuletzt geprüft am 27.10.2017.
- Scaled Agile, Inc. (2018): Case Studies. Online verfügbar unter <https://www.scaledagileframework.com/case-studies/>, zuletzt geprüft am 29.03.2018.
- Schäuffele, Jörg (2016): *Automotive Software Engineering. Grundlagen, Prozesse, Methoden und Werkzeuge effizient einsetzen*. 6. Auflage. Wiesbaden: Springer Vieweg (ATZ/MTZ-Fachbuch).
- Schiller, Fabian (2016): Agile Skalierung? Kein Problem! - Das LeSS-Framework. Online verfügbar unter <https://www.informatik-aktuell.de/management-und-recht/projektmanagement/agile-skalierung-kein-problem-das-less-framework.html>, zuletzt aktualisiert am 12.04.2016, zuletzt geprüft am 05.10.2017.
- Schuh, Günther; Hoffmann, Jörg; Gruber, Marie; Zeller, Violett (2017): Managing IT complexity in the manufacturing industry: an agenda for action. In: *Journal on Systemics, Cybernetics and Informatics* 15, S. 61–65.
- Schwaber, Ken; Scrum.org (2018a): Nexus™ Guide. Der gültige Leitfaden zur Skalierung von Scrum mit dem Nexus: Die Spielregeln. Unter Mitarbeit von Stefan Mieth, Pascal Naujoks und Florian Sauter. Scrum.org. Online verfügbar unter <https://scrumorg-website-prod.s3.amazonaws.com/drupal/2018-01/2018-Nexus-Guide-German.pdf?nexus-file=https%3A%2F%2Fscrumorg-website-prod.s3.amazonaws.com%2Fdrupal%2F2018-01%2F2018-Nexus-Guide-German.pdf>, zuletzt geprüft am 13.04.2017.
- Schwaber, Ken; Scrum.org (2018b): Online Nexus Guide. Nexus™ Guide. The Definitive Guide to Nexus: The Rules of the Game. Scrum.org. Online verfügbar unter <https://www.scrum.org/resources/online-nexus-guide>, zuletzt geprüft am 28.03.2018.
- Schwaber, Ken; Sutherland, Jeff (2016): *Der Scrum Guide. Der gültige Leitfaden für Scrum: Die Spielregeln*. Online verfügbar unter

- <http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-German.pdf>, zuletzt geprüft am 24.05.2017.
- Scott Ambler + Associates (2016): ENTERPRISE AGILE TRANSFORMATION: A PRODUCT COMPANY'S SUCCESSFUL ADOPTION OF DISCIPLINED AGILE. Better Quality | Quicker Time to Market | Dependable Delivery. Online verfügbar unter <https://disciplinedagileconsortium.org/resources/Pictures/Case%20studies/Approved%20Open%20Link%20Case%20Study.pdf>, zuletzt geprüft am 29.03.2018.
- Scrum Inc. (2017): Scrum Training & Certification. Online verfügbar unter <https://www.scruminc.com/>, zuletzt geprüft am 30.10.2017.
- Scrum.org (2018): Scaling Scrum with Nexus™. Online verfügbar unter <https://www.scrum.org/resources/scaling-scrum>, zuletzt aktualisiert am 04.08.2017, zuletzt geprüft am 07.08.2017.
- Scrum@Scale (2018): Scrum@Scale Case Study Library. Latest Scrum@Scale Case Studies. Online verfügbar unter <https://www.scrumatscale.com/case-study-library/>, zuletzt geprüft am 29.03.2018.
- Sekitoleko, Nelson; Evbota, Felix; Knauss, Eric; Sandberg, Anna; Chaudron, Michel; Olsson, Helena Holmström (2014): Technical Dependency Challenges in Large-Scale Agile Software Development. In: Wil van der Aalst, John Mylopoulos, Michael Rosemann, Michael J. Shaw, Clemens Szyperski, Giovanni Cantone und Michele Marchesi (Hg.): Agile processes in software engineering and extreme programming, 15th International Conference, XP 2014, Rome, Italy, May 26-30, 2014, Proceedings, Bd. 179. 1st edition. New York: Springer (Lecture Notes in Business Information Processing, 179), S. 46–61. Online verfügbar unter <https://pdfs.semanticscholar.org/be84/33ce71d15c5ea93c0a1f6c29d89a373f6835.pdf>, zuletzt geprüft am 10.08.2017.
- Smet, Jurgen de (2018): Agfa Healthcare. Online verfügbar unter <https://less.works/case-studies/agfa-healthcare.html>, zuletzt geprüft am 29.03.2018.
- Smits, Hubert (2008): 5 Levels of Agile Planning: From Enterprise Product Vision to Team Stand-up. Whitepaper. Rally Software Development Corporation. Online verfügbar unter <http://121.199.48.50/downloads/Five%20Levels%20of%20Agile%20Planning.pdf>.
- Sofer, Elad (2018): VeSecurity. LeSS Adoption for a Safety & Security Management Product. Online verfügbar unter <https://less.works/case-studies/vsecurity.html>, zuletzt geprüft am 29.03.2018.
- Stelzhammer, Markus; Wolter, Olaf (2017): Praxisbeispiel: Agiles Qualitätsmanagement der Siemens AG – Wie wir künftig Qualität managen. In: Peter F.-J. Niemann und Andre M. Schmutte (Hg.): Managemententscheidungen: Methoden, Handlungsempfehlungen, Best Practices. Wiesbaden: Springer Fachmedien Wiesbaden, S. 463–474.
- Strumberger, Klaus (2016): IT-Strategie für Unternehmenserfolg. Das ist die "IT der Zukunft". In: *marconomy.de*.
- Suomalainen, Tanja; Kuusela, Raija; Tihinen, Maarit (2015): Continuous planning: an important aspect of agile and lean development. In: *IJASM* 8 (2), S. 132–162. DOI: 10.1504/IJASM.2015.070607.
- Sutherland, Jeff (2014): Scrum Inc @ Agile2014. Scrum Inc. Online verfügbar unter <https://www.scruminc.com/agile2014/>, zuletzt aktualisiert am 27.07.2014, zuletzt geprüft am 07.08.2017.
- Sutherland, Jeff (2017): The Shu Ha Ri of Scaling. 30 Years of Large-Scale Agile Transitions. Scrum Inc. Scrum Gathering San Diego Apr 2017. San Diego, April 2017. Online verfügbar unter <https://www.scruminc.com/wp-content/uploads/2017/04/Scrum-at-Scale-Keynote-Scrum-Gathering-Apr-2017.pdf>, zuletzt geprüft am 19.09.2017.
- Sutherland, Jeff; Brown, Alex (2014): Scrum at Scale. Go Modular for Greater Success. Agile Alliance. Agile2014. Orlando, 30.07.2014. Online verfügbar unter <https://www.scruminc.com/wp-content/uploads/2014/07/Scrum-at-Scale-A-Modular-Approach.pdf>, zuletzt geprüft am 07.08.2017.
- Sutherland, Jeff; Scrum Inc. (2018): The Scrum@Scale Guide. The Definitive Guide to Scrum@Scale: Scaling that Works. Online verfügbar unter <https://www.scrumatscale.com/wp-content/uploads/Scrum@Scale-Guide.pdf>, zuletzt aktualisiert am 10.02.2018, zuletzt geprüft am 28.03.2018.
- Sutherland, Jeff; Sutherland, J. J.; Justice, Joe (2015): Scrum at Scale III. Expanding the Scaling Conversion. Scrum Inc., 05.02.2015. Online verfügbar unter <https://34slpa7u66f159hfp1fh9aur1-wpengine.netdna-ssl.com/wp-content/uploads/2015/02/Scrum-at-Scale-Pt-III-Slides.pdf>, zuletzt geprüft am 26.04.2017.
- Tarnowski, Michael (2007): Das Beste aus unterschiedlichen Welten. Mögliche Kombination von Prozess- und Metrikmodellen. In: *Qualität und Zuverlässigkeit* 52, S. 16–19.
- The LeSS Company B.V. (2017a): About LeSS - Large Scale Scrum (LeSS). Online verfügbar unter <https://less.works/de/resources/about.html#AboutLeSS>, zuletzt geprüft am 07.08.2017.
- The LeSS Company B.V. (2017b): LeSS Framework. Online verfügbar unter <https://less.works/de/less/framework/index.html>, zuletzt geprüft am 07.08.2017.
- The LeSS Company B.V. (2017c): LeSS Rules (April 2017). Online verfügbar unter <https://less.works/de/less/rules/index.html>, zuletzt geprüft am 07.08.2017.
- The LeSS Company B.V. (2017d): Overview - Large Scale Scrum (LeSS). Online verfügbar unter <https://less.works/de>, zuletzt geprüft am 05.10.2017.
- The LeSS Company B.V. (2017e): Prinzipien. Online verfügbar unter <https://less.works/de/less/principles/index.html>, zuletzt geprüft am 07.08.2017.
- The LeSS Company B.V. (2018a): Communities. Online verfügbar unter <https://less.works/less/structure/communities.html>, zuletzt geprüft am 29.03.2018.
- The LeSS Company B.V. (2018b): Coordination & Integration. Online verfügbar unter <https://less.works/less/framework/coordination-and-integration.html>, zuletzt geprüft am 29.03.2018.
- The LeSS Company B.V. (2018c): Daily Scrum. Online verfügbar unter <https://less.works/de/less/framework/daily-scrum.html>, zuletzt geprüft am 10.04.2018.
- The LeSS Company B.V. (2018d): LeSS Case Studies. Online verfügbar unter <https://less.works/case-studies/index.html>, zuletzt geprüft am 28.03.2018.
- The LeSS Company B.V. (2018e): Multi-Site Teams: Terms & Tips. Online verfügbar unter <https://less.works/de/less/framework/introduction.html#%E2%80%A2Multi-SiteTeams:Terms&Tips%E2%80%A2>, zuletzt geprüft am 07.04.2018.
- The LeSS Company B.V. (2018f): Product Backlog Refinement. Online verfügbar unter <https://less.works/de/less/framework/product-backlog-refinement.html>, zuletzt geprüft am 08.04.2018.
- The LeSS Company B.V. (2018g): Queuing Theory. Online verfügbar unter https://less.works/de/less/principles/queuing_theory.html, zuletzt geprüft am 10.04.2018.
- The LeSS Company B.V. (2018h): Sprint. Online verfügbar unter <https://less.works/de/less/framework/sprint.html>, zuletzt geprüft am 08.04.2018.
- The LeSS Company B.V. (2018i): Sprint Review. Online verfügbar unter <https://less.works/less/framework/sprint-review.html>.

- Theunissen, W. H.; Kourie, Derrick G.; Watson, Bruce W. (2003): Standards and Agile Software Development. In: Proceedings of the 2003 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology: South African Institute for Computer Scientists and Information Technologists.
- Thommen, Jean-Paul; Ruoff, Michèle (2016): Beyond Industrie 4.0. In: *Wenn die Technologie das Managementverständnis verändert* 31 (2), S. 66–72.
- Torry-Smith, Jonas Mørkeberg; Mortensen, Niels Henrik; Achiche, Sofiane (2014): A proposal for a classification of product-related dependencies in development of mechatronic products. In: *Research in Engineering Design* 25 (1), S. 53–74. DOI: 10.1007/s00163-013-0161-5.
- Verheyen, Gunther (2013): Agility Path. Online verfügbar unter https://de.slideshare.net/Ullizee/agility-path/3-Scrumorg_All_Rights_Reserved_33400166127122185 Under, zuletzt aktualisiert am 17.07.2013, zuletzt geprüft am 15.05.2017.
- VersionOne (2017): The 11th Annual State of Agile Report. VersionOne Inc. Online verfügbar unter <https://explore.versionone.com/state-of-agile/versionone-11th-annual-state-of-agile-report-2>, zuletzt geprüft am 19.05.2017.
- Weber, Sergej (2015): Agile in Automotive – State of Practice 2015. Kugler Maag Cie GmbH. Online verfügbar unter http://www.kuglermaag.de/fileadmin/05_CONTENT_PDF/2-22_agile-automotive_survey-2015.pdf, zuletzt geprüft am 21.08.2017.
- Wolfig, Roland (2016a): Agile in the Automotive Industry. BEYA. Online verfügbar unter <https://beya.io/2016/03/agile-in-the-automotive-industry/>, zuletzt aktualisiert am 16.03.2016, zuletzt geprüft am 24.08.2017.
- Wolfig, Roland (2016b): ASPICE and Agile - can it work? | BEYA. Online verfügbar unter <https://beya.io/2016/05/aspice-and-agile-can-it-work/>, zuletzt aktualisiert am 17.05.2016, zuletzt geprüft am 21.09.2017.
- Yakyma, Alex; Koehnemann, Harry (2016): Building Complex Systems with SAFe. Online verfügbar unter <https://blog.versionone.com/building-complex-systems-with-safe-scaled-agile-framework/>, zuletzt aktualisiert am 10.03.2016, zuletzt geprüft am 15.10.2017.