



Problembereiche verteilter agiler Teams: Literaturanalyse und Praxisimplikationen

Lukas Haslinger · Hermann Sikora · René Riedl

Eingegangen: 1. August 2020 / Angenommen: 10. August 2020 / Online publiziert: 1. Oktober 2020
© Der/die Autor(en) 2020

Zusammenfassung Seit Bestehen des Agilen Manifests 2001 bildeten sich vermehrt agile Softwareentwicklungs-Teams, die sich in Aufbau und Funktionsweise voneinander unterscheiden. Viele dieser Teams verteilen sich auf mehrere Standorte. Der Grad der Verteilung kann dabei von „in unterschiedlichen Gebäuden auf einem Firmengelände“ bis hin zu „anderer Kontinent“ reichen. Da das Agile Manifest auf stetige informelle Kommunikation (kurze, unbürokratische Wege) und vergleichsweise wenig Dokumentation setzt, sind bei verteilten Teams Schwierigkeiten und Probleme zu erwarten, wenn die Werte des Manifests konsequent verfolgt werden. Ein verteiltes Softwareentwicklungs-Projekt birgt an sich bereits Risiken (z. B. im Wissenstransfer zwischen den Entwicklern), wobei bei einem verteilten agilen Projekt zusätzliche Probleme auftreten können. Die Fachliteratur befasst sich seit rund 15 Jahren mit dieser Problematik. Befunde empirischer Studien beschreiben verschiedene Problembereiche; zudem werden Lösungswege erläutert. Im gegenständlichen Beitrag wird der aktuelle Stand der Literatur analysiert, um die wichtigsten Problembereiche herauszuarbeiten. Die Beschreibung dieser Problembereiche hilft insbesondere Handlungs- und Entscheidungsträgern in der Praxis, potenzielle Schwierigkeiten verteilter agiler Teams zu erkennen. Dies ist eine wesentliche Grundlage für proaktives Management, um die möglichen negativen Wirkungen zu vermeiden.

L. Haslinger (✉) · H. Sikora
Raiffeisen Software GmbH, Goethestraße 80, 4020 Linz, Österreich
E-Mail: lukahaslinger95@gmail.com

L. Haslinger · H. Sikora · R. Riedl
Institut für Wirtschaftsinformatik – Information Engineering, Johannes Kepler Universität Linz,
Altenberger Straße 69, 4020 Linz, Österreich

R. Riedl
Fakultät für Wirtschaft & Management, Abteilung für Digital Business, Fachhochschule
Oberösterreich, Wehrgrabengasse 1–3, 4400 Steyr, Österreich

Schlüsselwörter Agile Softwareentwicklung · Verteilte Agilität · Globale Softwareentwicklung · Literaturrecherche · Literaturanalyse · Scrum · Softwareentwicklung

Challenges of Distributed Agile Teams—Literature Review and Implications for Practice

Abstract Since the foundation of the Agile Manifesto in 2001, increasingly more agile software development teams have been formed, which differ in terms of structure and organization. Many of these teams are spread across multiple locations. The degree of distribution can range from “in different buildings on one company site” to “another continent”. As the Agile Manifesto focuses on constant informal communication (short unbureaucratic channels) and comparatively little documentation, difficulties and problems can be expected in distributed teams if one consistently follows the values of the manifesto. A distributed software development project already involves risks (e.g., transfer of knowledge between developers), and additional problems can arise in a distributed agile project. The academic literature has been dealing with this problem for around 15 years. Findings from empirical studies describe various problem areas and solutions are also outlined. The present article analyzes the current state of the literature in order to identify the most important problem areas. The description of these problem areas helps decision-makers in practice to better understand potential difficulties of distributed agile teams. This is an essential basis for proactive management in order to avoid possible negative effects.

Keywords Agile Software Development · Distributed Agile · Global Software Development · Literature Research · Literature Review · Scrum · Software Development

1 Einleitung

Die Expansion von Softwareunternehmen in andere Länder oder sogar auf andere Kontinente gewinnt immer mehr an Bedeutung, man denke hierbei etwa an Nearshoring oder Offshoring. Rund um die Uhr verfügbare Entwicklungsteams sowie die weltweite Präsenz mit Zugang zu neuen Märkten können mit Effektivitäts- und Effizienzsteigerungen einhergehen, bringen aber auch Herausforderungen mit sich. Diese Herausforderungen verschärfen sich im Kontext agiler Softwareentwicklung. In der Fachliteratur existieren zahlreiche Werke, die sich mit Agilität in standortübergreifenden Konstellationen beschäftigen („verteilte Agilität“), wobei dort die mit der Verteilung verbundenen Schwierigkeiten und Probleme als systemimmanent angesehen werden und der Fokus daher auf der Milderung resultierender Risiken liegt. Jene Beiträge, die sich konkret mit den Hintergründen und Ursachen der Probleme beschäftigen, tun dies entweder auf einer vergleichsweise oberflächlichen Ebene oder greifen ein einzelnes, dezidiertes Problem analytisch heraus. Ziel des vorliegenden Beitrags ist es, die in der Fachliteratur diskutierten und durch empirische

Studien identifizierten Problembereiche systematisch darzustellen. Eine solche Darstellung hilft Praktikern, mögliche Schwierigkeiten verteilter agiler Teams proaktiv zu behandeln. Dies leistet einen wirksamen Beitrag zum erfolgreichen Management agiler Entwicklungsprojekte in verteilten Umgebungen.

2 Methode der Literaturrecherche

Für die Literaturrecherche wurden folgende acht Datenbanken ausgewählt, die in ihren jeweiligen Themengebieten fundierte Inhalte für die Bearbeitung der gegenständlichen Thematik enthalten: ACM Digital Library, EBSCOhost (Business Source Premier), Emerald, IEEE Xplore, ScienceDirect, SpringerLink, Web of Science, Wiley Online Library.

Beim Festlegen des Suchbegriffes wurde der Ansatz gewählt, Keywords in ihrer Granularität soweit herunterzubrechen, bis sich eine repräsentative Ergebnismenge aus den jeweiligen Datenbanken erkennen lässt. Dies führte schlussendlich zu dem Suchbegriff (String) „agile distributed“. Die damit verbundenen Ergebnisse stellten eine fundierte Basis mit notwendiger Breite dar.

Im nächsten Schritt wurde je Paper eine grobe Inhaltsanalyse durchgeführt. Aus den Treffern konnten 36 relevante Papers identifiziert werden, die durch Backward-Search noch um weitere vier Artikel ergänzt werden konnten. Dies führte im Ergebnis zu insgesamt 40 relevanten Artikeln, die im Anhang angeführt sind.

Der Großteil der Artikel sind Fallstudienberichte (17), gefolgt von Praxisberichten (11) und Literaturanalysen (8). Der Rest (4) sind Studien anderer Art wie etwa als „deskriptive Studien“ bezeichnete Untersuchungen. Die den Artikeln zugrunde liegenden Datenerhebungsformen waren fast ausschließlich Interview und Beobachtung. Für die Fallstudien und Praxisberichte wurden dabei Stichproben von zumeist ein bis drei Unternehmen herangezogen, bei den Literaturanalysen wurden zwischen 14 und 86 Papers untersucht. Beim geografischen Umfeld war oft die Konstellation „USA–Indien“, aber auch „USA–Europa/Asien“ und generell „Europa“ genannt. Das vorherrschende agile Vorgehensmodell in den untersuchten Artikeln ist *Scrum*.

3 Methode der Literaturanalyse

Nach der Identifikation der relevanten Artikel wurden alle 40 Beiträge einer genauen Inhaltsanalyse unterzogen, die zur Identifikation von zehn Problembereichen (Konzepten) führte. Diese Problembereiche sind zwar nicht auf die „verteilte Agilität“ beschränkt (sie sind für allen Formen verteilter, dislozierter Zusammenarbeit relevant), allerdings haben sie bei der Agilität besondere Relevanz, da das agile Wertesystem ganz besonders auf das „Funktionieren“ gerade dieser Konzepte setzt.

Kommunikation Hier wird vor allem die fehlende persönliche Kommunikation bei verteilten Standorten thematisiert, aber auch Schwierigkeiten bei „virtueller“ Kommunikation (z. B. Telefon, Chat, Mail, Videokonferenz), die an Stelle der persönlichen Kommunikation treten muss.

Kultur Da verteilte Standorte oft auch unterschiedliche (Unternehmens-/Standort-) Kulturen aufweisen, sind Probleme hinsichtlich unterschiedlicher Arbeitseinstellungen (Mindset-Themen, z. B. hinsichtlich gelebter Resultatsorientierung), Offenheit (z. B. direkte vs. indirekte Thematisierung von Problemen) oder Auffassungen von Hierarchien (z. B. „Oben/unten“-Mentalität vs. „Augenhöhe“-Empfinden) zu beobachten, zu benennen und in Folge zu managen.

Vertrauen Durch den fehlenden unmittelbaren persönlichen Kontakt ist es typischerweise schwer für die Team-Mitglieder, Vertrauen zu Mitgliedern an anderen Standorten aufzubauen, insbesondere dann, wenn sie diese noch nie persönlich getroffen haben. Dies kann die Kommunikation hemmen und damit die agile Zusammenarbeit erheblich erschweren.

Sprachbarrieren Ähnlich wie bei (Unternehmens-/Standort-)Kulturen sind auch unterschiedliche Sprachen ein Problembereich bei überregional oder transkontinental verteilten Standorten. Fehlinterpretationen aufgrund mangelnder Sprachkenntnisse bei gleichzeitiger Präzisionsanforderung, was insbesondere im Bereich der Entwicklung von Mission Critical Software relevant ist, sind dabei die Folge.

Zeitverschiebung Das Arbeiten in unterschiedlichen Zeitzonen bringt die unvermeidliche Schwierigkeit mit sich, zeitlich für alle Mitglieder passende Meetings zu organisieren bzw. überlappende Arbeitszeiten herzustellen, insbesondere dann, wenn die zeitliche Komponente mit einer organisatorischen Notwendigkeit einhergeht (z. B. Check-In-/Check-Out-Prozesse von Software-Teilen in einem exakt definierten Zeitfenster).

Technische Limitationen Viele Erfolgsfaktoren der Zusammenarbeit in verteilten Teams basieren auf technischen Hilfsmitteln (z. B. Kollaborationsplattformen). Wenn diese nicht oder nicht ausreichend wirksam vorhanden sind, hat dies negative Auswirkungen auf die Produktivität.

Formalismus Verteilte Teams müssen vermehrt Formalismen in Form von streng definierten Prozessen und mitunter aufwändigen Dokumentationserfordernissen beachten, um die Nachteile des fehlenden unmittelbaren, persönlichen Team-Kontakts und den damit verbundenen „kurzen Wegen“ zu kompensieren. Diese Formalismen stehen jedoch im Widerspruch zu den Werten des Agilen Manifests, das sich durch wenig formalisierte Prozessabläufe und minimalistischer Dokumentation auszeichnet.

Team-Balance Wenn die Arbeitskräfte an den verschiedenen Standorten hinsichtlich Erfahrung und/oder Fach-/Methoden-/Werkzeug-Know-How ungleich verteilt sind, kann dies zu kontraproduktiven Abhängigkeiten zwischen Standorten führen, wenn sich diese Defizite nicht relativ zeitnahe nach dem Projektstart beheben oder zumindest reduzieren lassen. Außerdem können sich Team-Mitglieder an einem Standort isoliert fühlen, wenn lokale Ansprechpartner fehlen, um diese Defizite konkret bearbeiten zu können.

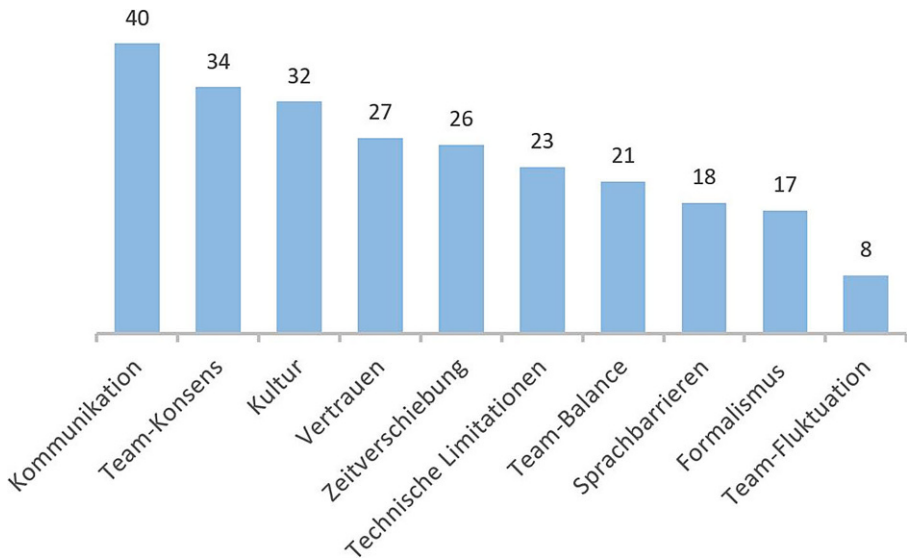


Abb. 1 Absolutes Vorkommen der Problembereiche ($N=40$ Artikel)

Team-Fluktuation Die Neuaufnahme von Team-Mitgliedern in einem verteilten Umfeld ist eine persönliche und organisatorische Herausforderung, da das Kennenlernen und der Vertrauensaufbau zeitintensive Prozesse sein können. Umgekehrt ist es insbesondere in einem dislozierten Umfeld schwierig, verloren gegangenes Wissen von ausscheidenden Mitgliedern zu bewahren und standortübergreifend neu zu verteilen.

Team-Konsens Über Standorte verteilt ist es bedeutend schwieriger als an einem einzelnen Standort, allen Team-Mitgliedern eine Vision „inhaltlich ident“ zu vermitteln sowie gemeinsame Sichtweisen auf eine Problemstellung abzusichern. Die „Informationen zwischen den Zeilen“, die den Kommunizierenden selbst oftmals gar nicht bewusst sind, können „virtuell“ zumeist nicht gleich wirkmächtig vermittelt werden. Daraus folgende unterschiedliche Wissensstände und Ansichten an den Standorten können Unmut hervorrufen und Fehlentwicklungen verursachen.

Im Zuge der Erstellung der Literaturanalyse wurde auch eine Auswertung erstellt, in der die Anzahl der Nennungen eines jeden Konzepts dokumentiert wurde. Daraus resultiert die in Abb. 1 dargestellte Rangordnung der Problembereiche. Man sieht, dass Kommunikation der am häufigsten und in jedem der 40 untersuchten Artikel genannte Problembereich ist. Am Ende der „Top-10-Problembereiche“ liegt Team-Fluktuation mit 8 Nennungen.

4 Diskussion der Konzepte

In diesem Kapitel werden die „Top-10-Problembereiche“ diskutiert. Dabei wird nach einem konzeptorientierten Ansatz vorgegangen, da ein autorenenorientierter Ansatz

nicht geeignet ist, die Konzepte zusammenzufassen und übergreifend zu betrachten. Mit diesem Aufbau wird erreicht, dass die Erkenntnisse der Literaturanalyse zu den einzelnen Konzepten (Problembereichen) übersichtlich dargestellt werden können (Webster und Watson 2002).

4.1 Kommunikation

In standortübergreifenden (verteiltern) agilen Teams ist es die zentrale Herausforderung, herauszufinden, wie die Kommunikationskanäle optimal gestaltet werden können – alle anderen Konzepte hängen direkt oder indirekt mit dem Kommunikationskonzept zusammen. Wenn man sich an den agilen Werten dogmatisch orientiert, stellen E-Mail oder Instant-Messaging-Dienste keine ausreichende Alternative für die tägliche „face-to-face“-Kommunikation dar.

Bei audiounterstützter Real-Time-Kommunikation entfallen zumindest die Antwortzeiten asynchroner Kommunikationsmittel, problematisch wird es aber dennoch, wenn z. B. Bildschirmhalte geteilt werden, diese aber nicht bei allen Teilnehmern synchron verarbeitet werden. Somit kann es auch beim Kundenkontakt zu falschen Auffassungen von Prioritäten, Meinungen oder Feedback kommen. Zusätzlich kann es bei Verzögerungen der Tonspuren zu überlappenden Aussagen kommen, die es schwer machen, eine strukturierte Konversation zu führen (Kajko-Mattsson et al. 2010). In den letzten Monaten haben sich durch die Corona-Krise bedingt neue Phänomene als bedeutsam herauskristallisiert, unter anderem „video call fatigue“ (Ames 2020) sowie „Zoom fatigue“ (Wiederhold 2020). Im Kern geht es hierbei darum, dass intensive, auf Videokonferenzsystemen basierende Kommunikation nach einiger Zeit zu ausgeprägten Ermüdungs- und Stresserscheinungen führen kann. Daraus folgt, dass bei der fehlenden Möglichkeit von „face-to-face“-Kommunikation und der sich daraus ergebenden Notwendigkeit von Videokonferenzen langfristig ungünstige Wirkungen für das Wohlbefinden und die Gesundheit von Mitarbeiter resultieren können.

Oftmals besteht zwischen verteilten Teams und dem Kunden selbst ebenfalls eine (geografische) Distanz, die sich natürlich auch wieder auf die Häufigkeit und Form der Kommunikation niederschlägt. Im agilen Umfeld, wo es gilt, stets rasch auf Anforderungsänderungen zu reagieren und diese effizient zu kommunizieren, stellt die durch die Dislozierung der Beteiligten verursachte „unregelmäßige Kommunikation“ ein bedeutendes Problem dar (Shrivastava und Rathod 2014).

Schriftliche Kommunikation kann ebenfalls zu Missverständnissen führen, insbesondere dann, wenn man mit der Person, mit der man E-Mail- oder Chat-Verkehr hat, noch nie vorher oder nur selten persönlich gesprochen hat. Jede Person hat eigene Normen in ihrer Kommunikation, die oftmals nur „face-to-face“ identifizierbar sind, zumal diese Art der Kommunikation zusätzlich durch Mimik und Gestik unterstützt wird (Rothman und Hastie 2013).

Aus verteilter Kommunikation ergeben sich Zusatzaufwand und Zusatzkosten. Durch die verstärkte Nutzung von E-Mails wird auch „unfreiwillig“ eine Art Dokumentation geschaffen, die eben Zeit beansprucht und Kosten verursacht (umgekehrt aber auch durchaus Nutzen stiften kann). Ein hohes Maß schriftlicher Kommunikation steht aber jedenfalls im Widerspruch mit den agilen Werten (Cottmeyer 2008).

Ein weiteres Hindernis bei der Benutzung von asynchronen Kommunikationskanälen ist die Tatsache, dass sich z. B. der Absender nicht sicher sein kann, ob der Empfänger eine Mail gelesen oder diese vielleicht übersehen hat (selbst dann, wenn das Instrument der Empfangsbestätigung verwendet wird). Ein häufiger Kritikpunkt des Instruments „E-Mail“ ist darin begründet, dass oftmals über den E-Mail-Verkehr ein Problem nur längere Zeit geschoben wird, was es in weiterer Folge noch komplizierter macht („Produktivitätskiller“). Im schlimmsten Fall kann es passieren, dass Entscheidungen auf Basis von veralteten Informationen getroffen werden (Korkala und Maurer 2014).

Bei der Anwendung synchroner Kommunikation (z. B. Telefon, Live-Chat, Video-Call) kann wiederum die Hemmung bestehen, sein Gegenüber direkt zu kontaktieren. Bei zwei Parteien, die sich noch nicht (gut) kennen, wie es bei verteilter agiler Softwareentwicklung oft der Fall ist, wird vielfach der Weg der indirekten Kommunikation gewählt, statt bewusst den direkten Kontakt zu suchen (Rizvi et al. 2015).

Mit Reisen zwischen den Standorten und damit innerhalb der Teams kann man die Schwierigkeiten zumindest punktuell überwinden und auch „face-to-face“-Kommunikation ermöglichen. Oftmals ist aber kein Budget vorhanden, um diese in regelmäßigen Intervallen durchzuführen; das Verhältnis der Team-Mitglieder untereinander bleibt distanziert (Shrivastava und Rathod 2017).

Ein Folgeproblem, das aus der eingeschränkten Kommunikation resultieren kann, sind „isolierte Meetings“ an den jeweiligen Standorten. Dazu gehören zum Beispiel separate Daily-Meetings, die aufgrund ihrer Einfachheit durchgeführt werden und damit zumindest einen Teil der Probleme bei Audio-/Videokonferenzen vermeiden sollen (Lous et al. 2018).

4.2 Team-Konsens

Das Problem eines fehlenden Team-Konsenses ist zumeist das Resultat qualitativ und/oder quantitativ unzureichender Kommunikation mit dem Kunden. Jene Team-Mitglieder, die am selben Standort wie der Kunde (oder ihm zumindest nahe) arbeiten, pflegen quasi „automatisch“ deutlich mehr kommunikative Nähe zu diesem. Bei Meetings mit Kunden werden die Visionen von Projekten abgesteckt und Anforderungen face-to-face besprochen. Für entfernte Team-Mitglieder kann – trotz aller Bemühungen, „Remote-Nachteile“ durch alternative Kommunikationsmechanismen zu kompensieren – der Fall eintreten, dass der Gesamtüberblick verloren geht und Anforderungen nicht oder nicht korrekt aufgefasst und anschließend fehlerhaft implementiert werden (Kajko-Mattsson et al. 2010).

Aber auch innerhalb eines Teams können verschiedene Auffassungen zu einer Anforderung oder zu Umsetzungsrichtlinien und -standards auftreten. So kann es dazu kommen, dass sich an den verschiedenen Standorten unterschiedliche Coding-Richtlinien oder Designentscheidungen etablieren, diese aber mangels Kommunikation sowie Schwächen in der Durch- und Umsetzung (z. B. als Resultat einer Führungsschwäche oder eines Erfahrungsdefizits des Scrum Masters) nicht ausreichend klar weitergetragen werden (Shrivastava und Rathod 2014).

Unterschiedliche Auffassungen über Rollen und Verantwortungsbereiche hemmen ebenfalls die Produktivität verteilter agiler Teams. So kann es sein, dass „ein

Standort vom anderen glaubt“, dass er für eine Sache zuständig ist bzw. beide sich in der Verantwortung sehen, wodurch es zu fehlenden oder doppelten Implementierungen kommen kann (Šmite et al. 2010).

Uneinigkeiten in einem Team bergen erhebliches Konfliktpotential, wenn Personen nicht von ihrer Meinung oder ihrem Prozess abweichen wollen. Ein Mensch nimmt – quasi „von Natur aus“ – an, dass sein Gegenüber (Team Peer) ähnlichen Normen und Werten folgt, wie sie selbst gelebt werden. Diese Annahme ist oft unrealistisch, weshalb es Rollen in agilen Teams gibt (insbesondere Scrum Master), die in ihrer Führungsarbeit dafür sorgen müssen, allfällig auftretende Defizite zu beheben oder erst gar nicht wirksam werden zu lassen. In verteilten, agilen Teams ist diese potenzielle Kluft noch größer als in lokalen Teams (Rothman und Hastie 2013). Durch die bereits vorher erwähnte „emotionslose Kommunikation“ in verteilten Teams kommt es auch vermehrt gerade in sensiblen oder heiklen Themen zu Fehlinterpretationen und missverstandenen Ansichten (Alzoubi et al. 2016).

Ein Folgeproblem der fehlenden oder unterschiedlich interpretierten Visionen eines Projekts ist, dass Entwicklern dadurch oftmals die Möglichkeit fehlt, zukunftsfähige Designentscheidungen zu treffen (z. B. hinsichtlich der Architektur wirkmächtig und wartungsfreundlich). Dadurch werden einfache Möglichkeiten für die Implementierung künftiger Features nicht erkannt, was in Folge zu aufwändigem Code-Refactoring führt (Therrien 2008).

Da in verteilten Teams oftmals auch dritte Unternehmen als externe Vertragspartner eingebunden sind, kann es (beispielsweise aus Gründen der Sicherheit oder des bewussten Verbergens von spezifischem Know-how) zu Zugangsbeschränkungen für diese Partner kommen. Das hindert den Prozess des gemeinsamen Wissensaufbaus und hinterlässt ein Ungleichgewicht, was wiederum dem Team-Konsens entgegensteht (Cichocki und Maccari 2008). Diese Thematik ist nicht beschränkt auf agile Methoden, allerdings wirkt sie auf Grund des zu lebenden agilen Wertesystems potenziell stärker negativ als bei anderen Entwicklungsparadigmen.

Bei großen Projekten, in denen mehrere agile Teams separat an verteilten Standorten arbeiten, werden oft nur lokale und eher selten verteilte Team-Meetings abgehalten. Es mangelt in Folge an Transparenz, was an dem jeweiligen Standort momentan geschieht, die gesamtheitliche Projektsicht im nötigen Detaillierungsgrad geht verloren (Paasivaara et al. 2009). Das betrifft auch die Übersicht, an welchen Aufgaben ein Team zu einem bestimmten Zeitpunkt konkret arbeitet (Aslam et al. 2017).

4.3 Kultur

Wenn verteilte agile Teams länderübergreifend eingerichtet werden, treffen im Regelfall unterschiedliche Kulturen aufeinander. Daraus resultieren Herausforderungen, die zumeist mit kulturell geprägten Arbeitseinstellungen zu tun haben (Beispiele: Perfektionsstreben vs. Fehlertoleranz; Nachfragefreudigkeit vs. Nachfragevermeidung; Konfliktfreudigkeit vs. Konfliktvermeidung; proaktive Problemmunikation vs. Angst vor Gesichtsverlust). Innerhalb einer Kultur werden dieselben oder ähnliche Ansichten, Werte oder Hierarchien geteilt, die sich in einem länder- und kulturübergreifenden Team gegenüberstehen können (Kajko-Mattsson et al.

2010). Auch der konkrete Umgang mit agilen Praktiken kann sich je Kultur deutlich unterscheiden und zu Schwierigkeiten führen (Summers 2008).

So kann es beispielsweise die Eigenschaft einer Kultur sein, bei Problemen eher nicht oder so spät wie möglich um Hilfe zu fragen, wohingegen in einer anderen Kultur ohnehin erst dann unterstützt wird, wenn direkt um Hilfe gebeten wird (Yap 2005). Wegen dieser Unterschiede ist es oftmals schwierig zu erkennen, ob ein Team Anforderungen oder Aufgaben vollinhaltlich verstanden hat (Paasivaara et al. 2008). In diesen Konstellationen wird beispielsweise oft auch nicht oder nicht rechtzeitig zugegeben, wenn Sprint-Ziele nicht erreicht werden können (Angst vor Gesichtsverlust). Dies führt zum späten Erkennen von Risiken und verleitet oder zwingt die Entwickler dazu, übermäßig viele Überstunden (ggf. auch verdeckt) zu absolvieren. Dies führt zu falschen Projektbildern und vermeidbaren Risiken (Zaghoul et al. 2015). Diese negativen Beispiele können auch innerhalb *einer* Kultur auftreten, allerdings erhöhen sich die Risiken dafür erheblich, wenn kulturübergreifend verteilt gearbeitet wird und der Scrum Master keine wirksamen Gegenmaßnahmen in seine Arbeit einfließen lässt.

In vielen Ländern werden die Hierarchien und Abgrenzungen der Rollen in einem Projekt sehr strikt wahrgenommen. So fühlen sich in manchen Kulturkreisen erfahrene Entwickler qua Seniorität dazu ermächtigt, nur Implementierungsaufgaben durchzuführen, während Aufgaben des Testens und Dokumentierens eher für junge, in der Hierarchie weiter unten angesiedelte Entwickler vorgesehen sind (Kajko-Mattsson et al. 2010). Auch und gerade das Verständnis von Softwarequalität kann sich je nach Kultur deutlich unterscheiden (Pries-Heje und Pries-Heje 2011).

Durch strenge Hierarchien kann auch der gemeinsame Wissensaustausch oder zumindest seine Effektivität und Effizienz verloren gehen, wenn etwa Team-Mitglieder glauben, sie sollten nur der Führungskraft, also vertikal, etwas berichten. Durch diesen Mangel an horizontaler Kommunikation werden die Kollegen nicht eingebunden und der Konsens wird gehemmt (Rizvi et al. 2015).

In Indien wird beispielsweise die Eskalation eines Themas sehr grob und rigoros behandelt bzw. per se als sehr ernst gewertet. In dieser Kultur sollte man daher mit einer allfälligen Eskalation sehr vorsichtig verfahren, um unnötige Auseinandersetzungen zu vermeiden bzw. die dort ansässigen Team-Mitglieder nicht (im Sinne eines Kollateralschadens für das Projekt) einzuschüchtern (Cottmeyer 2008).

Öffentliche Ferien oder beliebte Urlaubszeiten sind über Kulturen hinweg auch durchwegs unterschiedlich. Diese nur scheinbar weichen Faktoren sind unbedingt im Vorhinein in der Projektplanung zu berücksichtigen, da es andernfalls zwangsläufig zu eigentlich vermeidbaren Frustrationen kommt, wenn während dieser Zeiten gearbeitet werden muss (Kajko-Mattsson et al. 2010). Religiöse Einflussfaktoren können diese Problematik noch weiter verschärfen.

4.4 Vertrauen

Für verteilte Teams stellt es oftmals ein Hindernis dar, durch die fehlende persönliche Nähe, das für den gemeinsamen Projekterfolg nötige „Grund-Vertrauen“ aufzubauen und in Richtung eines professionellen Team-Vertrauens wachsen zu lassen. Durch das fehlende Vertrauen ist es für die Team-Mitglieder schwer, persönliche

Schwächen (z. B. Erfahrungsmangel) oder Verletzbarkeit (z. B. bei verbaler Kommunikation) zu zeigen bzw. offen in der Kommunikation zu sein (Kajko-Mattsson et al. 2010). Ohne Vertrauen kann schnell eine „krampfhaft Professionalität“ entstehen, die dazu führt, dass es in Gesprächen nur mehr um „die Arbeit“ (z. B. abzuarbeitende Aufgaben) geht, was das gegenseitige Kennenlernen und damit einhergehend den Vertrauensaufbau hemmt (Young und Terashima 2008).

Vertrauen stellt per se die Basis für offene, zielgerichtete Zusammenarbeit dar, da ohne Vertrauen Aufgaben nur ungern abgegeben werden, auch wenn dies das Kompetenz-Profil des Teams erfordern würde. Somit wird auch die Offenlegung komplexer Themenstellungen nicht gefördert (Šmite et al. 2010). Scrum Master sind dazu angehalten, Maßnahmen zu ergreifen, die den Aufbau von Vertrauen befördern (z. B. gemeinsame Aktivitäten, ein wertschätzendes Klima offener Kommunikation schaffen).

Besteht zwischen Standorten dieses „Grund-Vertrauen“ nicht (ausreichend), kann es dazu kommen, dass standortinterne Informationen bzw. an gewissen Standorten angefertigte Dokumentationen nur ungern oder gar nicht geteilt werden, was insgesamt zu Missverständnissen sowie Verzögerungen und damit negativen Projektkonsequenzen führen kann (Korkala und Maurer 2014).

Der Grad an Vertrauen, den verschiedene Team-Mitglieder empfangen oder ausdrücken können bzw. möchten, kann auch kulturell-regional stark variieren, was es noch schwieriger macht, Vertrauen aufzubauen (Alzoubi et al. 2016). So ist es etwa für Entwickler aus Indien ohne ausreichende Vertrauensbasis sehr schwierig, Hindernisse klar zu benennen oder einen ehrlichen und somit ungeschönten Projektstatus mitzuteilen (Kahya und Seneler 2018).

4.5 Zeitverschiebung

Das Organisieren von Meetings stellt sich bei größeren Zeitverschiebungen als große Herausforderung dar, weil man Meetings oft außerhalb der Geschäftszeiten abhalten muss, um allen Team-Mitgliedern die Möglichkeit der Teilnahme zu bieten. Für viele Team-Mitglieder heißt das auch, dass man zuhause zwingend erreichbar sein muss, um gegebenenfalls entfernte Standorte noch unterstützen zu können. Die ausgedehnte (zwingende) Verfügbarkeit und die daraus resultierenden langen Arbeitstage fördern Frustrationen und widersprechen dem agilen Prinzip, möglichst wenige Überstunden zu produzieren (Kajko-Mattsson et al. 2010). Da dies auch ein Eingriff in das persönliche Leben ist, führt dies nicht selten zu Konflikten (Yap 2005). Diese Konflikte treten vor allem dann auf, wenn die Nachteile der Zeitverschiebung immer nur oder meistens eine Zeitzone treffen oder die Nachteile nur in einer Zeitzone arbeitsrechtlich akzeptiert werden können.

Vor allem bei gewissen Rollen ist die persönliche Verfügbarkeit in überlappenden Arbeitszeiten von großer Bedeutung. Wenn etwa Tester und Entwickler zeitzonemäßig zwölf Stunden auseinander liegen, ist es schwierig bis unmöglich, Abstimmungen oder Reviews effektiv und effizient vorzunehmen. In dem kreativen Prozess des Testens gemeinsam mit dem Entwickler ist die synchrone Zusammenarbeit für den Projekterfolg entscheidend (Rothman und Hastie 2013).

Gleiches gilt auch für Feedback-Schleifen, die aufgrund abweichender Arbeitszeiten teilweise nur verzögert bzw. asynchron erfolgen können. Somit geht wichtige Zeit verloren, wenn ein Entwickler ohne Feedback nicht gezielt weiterarbeiten kann und bei Zeitdruck im schlimmsten Fall Annahmen treffen muss (Kahya und Seneler 2018). Außerdem verleiten entfernte Zeitzonen dazu, mehr über asynchrone Kanäle wie etwa E-Mail zu kommunizieren, auch wenn dies im jeweils konkreten Fall nicht die Kommunikationsform der Wahl ist oder sein sollte (Rizvi et al. 2015).

Die Schwierigkeiten bei der Planung von Meetings führen auch dazu, dass der Einfachheit halber Sprint-Planungs-Meetings grundsätzlich nur an einem Standort abgehalten werden (Therrien 2008). In diesem Fall gilt es darauf zu achten, die zugehörigen Zeiten möglichst zu variieren, da sonst immer nur ein Standort zur besonders späten oder frühen Uhrzeit gefordert ist (Paasivaara et al. 2008).

4.6 Technische Limitationen

Verteilte Teams sind stark auf Video- bzw. Audiokonferenzen angewiesen, weshalb gewährleistet werden muss, dass diese ohne technische Probleme durchgeführt werden können. Häufig wird bei Meetings viel Zeit für die Lösung mitunter banaler technischer Probleme beansprucht. Schlechte Audioqualität gepaart mit mehreren Teilnehmern, die gleichzeitig sprechen, tragen negativ zur Effizienz solcher Konferenzen bei (Kajko-Mattsson et al. 2010). Im schlimmsten Fall müssen die Meetings verschoben oder abgebrochen werden, wenn sie aufgrund der technischen Gegebenheiten nicht durch- oder fortgeführt werden können (Korkala und Maurer 2014). Wiederholtes Auftreten technischer Probleme kann dazu führen, dass weniger Meetings abgehalten und somit weniger kommuniziert wird (Rothman und Hastie 2013).

Besonders problematisch sind solche Limitationen bei Sprint-Reviews. Dem Kunden kann die entwickelte Software in ihrer tatsächlichen Erscheinung nicht oder nicht ausreichend veranschaulicht werden, was in Folge nicht nur das Vertrauen des Kunden in die Fähigkeiten der Teamorganisation per se erschüttert, sondern die Kundenabnahme erschwert oder verunmöglicht (Rizvi et al. 2015).

Auch bei der Synchronisation von Code oder beim Deployment können technische Limitationen aufgrund unpassender oder nicht durchgängig verfügbarer Tools standortübergreifende Zusammenarbeit behindern (Yap 2005). Auch beim Synchronisieren von Scrum-Boards oder Backlogs kann es standortübergreifend zu Einschränkungen kommen, was sich negativ auf die Transparenz des Projektfortschritts über die Standorte auswirkt (Vallon et al. 2013).

4.7 Team-Balance

Wenn technische Fähigkeiten, Erfahrung mit Agilität oder Rollen ungleich an den Standorten verteilt sind, kann dies zu langen Kommunikationswegen und lokalen Unsicherheiten führen. Ein verteiltes Team sollte insbesondere hinsichtlich Rollen, Skills und Erfahrungen in der Team-Zusammenstellung auf eine Balance über die Standorte hinweg achten. Sollten erfahrene Senior-Entwickler am lokalen und unerfahrene Junior-Entwickler am entfernten Standort angesiedelt sein, sind diese regelmäßig auf ausgeprägte Unterstützung durch die Personen des lokalen Standorts

angewiesen (Kajko-Mattsson et al. 2010). Daraus resultiert ein kontraproduktives Gefälle im Selbstverständnis der Standorte.

Weiter ist auf die Diversität der Team-Mitglieder zu achten. Sollten sich zwei Standorte demografisch und von den Persönlichkeitsprofilen stark unterscheiden und diese Gegebenheiten im täglichen Management zu wenig Beachtung finden, besteht die Gefahr, dass sich aufgrund unterschiedlicher Interessen, Ziele oder Lebenseinstellungen die Standorte hinsichtlich Zusammengehörigkeitsgefühl klar auseinander bewegen statt sich zu einer organisatorisch homogenen Einheit zu entwickeln („zwei Standorte, ein Team“). Es ist wichtig, einen der Projektaufgabe angemessenen „Mix & Fit“ hinsichtlich Persönlichkeit und Demografie der Team-Mitglieder je Standort zu erreichen, um dadurch Diskrepanzen und ihre negativen Wirkungen zu vermeiden (Lous et al. 2018).

Außerdem fühlen sich erfahrene Entwickler durch eine ungleiche Balance oftmals dazu verpflichtet, mehr zu arbeiten, um die Projektgeschwindigkeit konstant zu halten. Dies führt jedoch – bei grundsätzlicher Wertschätzung für diese Bereitschaft – zu exzessiven Überstunden und in weiterer Folge zu Demotivation (Cottmeyer 2008). Jede Rolle an einem Standort, sofern zumindest zweifach vorgesehen, sollte idealtypisch eine (gleichberechtigte) Gegenrolle an anderen Standorten haben. So sollten zum Beispiel an jedem Standort Architekten sein, die sich untereinander austauschen können (Paasivaara et al. 2008).

Sollte es Führungskräfte bzw. Product Owner oder Scrum Master nur an einem Standort geben, wird die Team-Zusammenstellung, der Vorteile wegen, oft auf diesen Standort hin optimiert, wodurch dieser zwar volle Handlungsgewalt hat, der entfernte Standort aber dadurch eher isoliert wird, weil dort Ressourcen fehlen (Therrien 2008). Durch fehlende Führungskräfte an einem Standort („Wir sind hier scheinbar nicht so wichtig“) wirkt die Arbeit dort auch oft intransparent; es fehlt die direkte Beobachtung des Teams (Šmite et al. 2010).

4.8 Sprachbarrieren

In globalen verteilten Teams sind meistens verschiedene Nationalitäten und infolgedessen auch verschiedene Sprachen vertreten. Es gilt, sich auf eine Sprache zu einigen. Die Wahl fällt oft auf Englisch, da in dieser Sprache an den verschiedenen Standorten meist akzeptable Kenntnisse vorliegen. Trotzdem ist es oft so, dass gewisse Standorte mit Englischkenntnissen gegenüber anderen Standorten hinterherhinken. Missverständnisse und Unklarheiten sind die Folge, was sich mit den agilen Prinzipien der direkten, offenen Kommunikation nicht vereinbaren lässt (Kajko-Mattsson et al. 2010). Persönliche Defizite in Sprachkenntnissen können außerdem auch dazu führen, dass man sich generell in Meetings weniger beteiligt, da man konkrete Sachverhalte weniger gut diskutieren kann bzw. Angst davor hat, sich nicht adäquat ausdrücken zu können (Paasivaara et al. 2009). Daraus folgt, dass die Entwicklung eines Projekts verhältnismäßig stärker durch Personen mit besseren Englischkenntnissen beeinflusst wird.

Bei sprachlichen Barrieren in komplexen Sachverhalten wird oftmals als Alternative zu asynchroner, schriftlicher Kommunikation gegriffen. Diese nimmt wiederum

mehr Zeit in Anspruch, ist weniger zielgerichtet und mindestens genau so schwierig zu interpretieren (Shrivastava und Rathod 2019).

Sprachliche Barrieren müssen nicht immer nur aus der Gegebenheit verschiedener Muttersprachen der Team-Mitglieder resultieren. Auch die Art, wie, wie oft und wie klar sich Personen ausdrücken (können), kann innerhalb derselben Sprache unterschiedlich sein und wirken (Rizvi et al. 2015).

4.9 Formalismus

Für verteilte Teams ist es oftmals unvermeidlich, durch die verschiedenen Arten von Distanzen (geografische, zeitliche, sprachliche, kulturelle, mentalitätsmäßige, persönliche) ein notwendiges Maß an Formalismus anzuwenden. Dies kann etwa in Form von formaler Kommunikation, Dokumentation und Standards aller Art erfolgen, was allerdings in Folge die Agilität in diesen Projekten stark negativ beeinflussen kann. Schon der vermehrte Austausch von E-Mails führt zu „unfreiwilliger Dokumentation“ (Kajko-Mattsson et al. 2010) und Mitarbeiterstress (Fischer und Riedl 2017). Ein Großteil dieser Dokumentation ist aber als Wegwerfprodukt anzusehen, den Kosten einer solchen Dokumentation steht kaum ein Nutzen gegenüber (Cottmeyer 2008).

Auch beim Management von Anforderungen (Demands, Backlogs) muss ein Mindestmaß an Formalismus definiert und eingehalten werden. Durch die Unmöglichkeit „impliziter Kontrollen“ (durch Anwesenheit) an entfernten Standorten müssen oft fixe, vorgefertigte Abnahmekriterien definiert werden. Das agile Umfeld ist aber geprägt von laufenden Veränderungen, auch bei Abnahmekriterien. Es schränkt dadurch die Flexibilität des Teams ein (Ramesh et al. 2006).

In verteilten Teams sind auch aufgrund der oftmals mangelnden Kommunikation viele Standards zu implementieren, was aber wiederum die Team-Mitglieder in ihren Freiheiten bei Vorgehensweisen einschränkt (Hossain et al. 2009). Vor allem Tester benötigen im verteilten Umfeld eine detaillierte Anforderungsdokumentation, der per se laut dem Agilen Manifest eigentlich nicht viel Beachtung geschenkt werden sollte (Shrivastava und Rathod 2019).

Bei hinsichtlich Erfahrung sowie Skills ungleich verteilten Standorten ist zumeist eine deutlich aufwändigere Dokumentation erforderlich, um die weniger erfahrenen sowie versierten Standorte auf einem ausreichenden Produktivitäts- und Qualitäts-Niveau halten zu können (Korkala und Maurer 2014).

4.10 Team-Fluktuation

Für standortübergreifende Teams ist es verhältnismäßig schwierig, neue Team-Mitglieder integrieren zu können bzw. bestehende geordnet abzugeben. Es gilt, ein neues Team-Mitglied allen anderen Team-Mitgliedern vorzustellen („Onboarding“), was viel Zeit beanspruchen kann, die einzuplanen ist. Auch die Einarbeitungszeit kann hoch ausfallen, wenn Informationen von verschiedenen Standorten benötigt werden (Shrivastava und Rathod 2014). Die Erfordernisse des Onboardings und der Integration neuer Team-Mitglieder implizieren entgegen dem Agilen Manifest einen nicht unerheblichen Dokumentationsaufwand (Bose 2008). Außerdem ist es für die

anderen Standorte schwierig, „remote“ Vertrauen zum neuen Mitglied aufzubauen (Kajko-Mattsson et al. 2010). Dies ist aber für ein produktives Miteinander Grundvoraussetzung, weshalb dieser Problematik durch das Management entsprechende Aufmerksamkeit und Priorität einzuräumen ist. Dem Scrum Master kommt dabei eine Schlüsselrolle zu.

In gewissen Kulturen ist es üblich, nach Erreichen gewisser Erfahrungsstufen befördert bzw. als Führungskraft berufen zu werden. Somit kann sich die Team-Zusammensetzung immer wieder ändern und sich in weiterer Folge auf die Projektgeschwindigkeit und auch auf die mühevoll aufgebaute Team-Chemie negativ auswirken (Cottmeyer 2008).

Ein weiteres Problem bei verteilten Teams ist, dass es Länder mit grundsätzlich hoher Fluktuation in Unternehmen und damit hohen personellen Schwankungen innerhalb der Teams gibt. In Indien und China beispielsweise herrschen hohe Fluktuationsraten von jährlich 30–50 % vor, was in weiterer Folge zu Wissensverlust führt. Dass im agilen Umfeld Wissen oftmals bewusst nicht explizit dokumentiert wird (sofern überhaupt dokumentierbar), weil auf hohe Team-Kohäsion und geringe Fluktuation gesetzt wird, verstärkt diesen negativen Effekt (Sutherland et al. 2008).

5 Ausblick

Im vorliegenden Beitrag wurde eine konzeptorientierte Literaturanalyse vorgestellt, die nach systematischer Recherche angefertigt wurde. Dabei wurden 40 Artikel analysiert, um die zehn häufigsten Problembereiche verteilter agiler Teams herauszuarbeiten. Es wurde eine Rangliste erstellt, mit welcher Häufigkeit die identifizierten Schwierigkeiten in der Fachliteratur vorkommen (vgl. Abb. 1). Die entscheidende Problematik ist Kommunikation (sie wurde in allen 40 Artikeln thematisiert). Im Beitrag wurde jeder der zehn Problembereiche diskutiert, und zwar in einer Art, so dass daraus Implikationen für die Praxis offenkundig werden.

Für die Praxis kann die Arbeit somit als Informationsquelle vor der Transformation in ein agiles Umfeld bei bestehender Verteiltheit oder vor der Distribution bestehender agiler Team-Strukturen an einem Standort auf mehrere Standorte herangezogen werden. Zur Vorbereitung können die Herausforderungen, die sich typischerweise ergeben, dem vorliegenden Beitrag entnommen werden. Darauf aufbauend können bereits im Vorhinein vorbeugende Maßnahmen getroffen und regelmäßiges Leitungshandeln (Daily Management) zur Entschärfung der beschriebenen Problembereiche institutionalisiert werden, auch wenn dies der Dogmatik des Agilen Manifests in Teilen widerspricht. Schließlich können die hier vorgestellten Problembereiche Ausgangspunkt vertiefter empirischer Forschung sein. Qualitative Forschung in Form von Fallstudien oder Interviewstudien könnte etwa jeden der zehn Bereiche vertieft untersuchen. Quantitative Forschung könnte theoriegeleitet die zehn Bereiche als unabhängige Variablen konzeptualisieren, um ihren Einfluss auf abhängige Variablen wie Projektlaufzeit, Projektkosten, Zufriedenheit der Entwickler und Softwarequalität festzustellen (Riedl 2019).

Funding Open access funding provided by Johannes Kepler University Linz.

Open Access Dieser Artikel wird unter der Creative Commons Namensnennung 4.0 International Lizenz veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Artikel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.

Weitere Details zur Lizenz entnehmen Sie bitte der Lizenzinformation auf <http://creativecommons.org/licenses/by/4.0/deed.de>.

Anhang

1. Alsaqaf W, Daneva M, Wieringa R (2019). Quality requirements challenges in the context of large-scale distributed agile: An empirical study. *Information and Software Technology*, Vol. 110: 39–55.
2. Alzoubi Y I, Gill A Q, Al-Ani A (2016) Empirical studies of geographically distributed agile development communication challenges: A systematic review. *Information & Management*, Vol. 53: 22–37
3. Aslam W, Ijaz F, Lali M I, Mehmood W (2017) Risk Aware and Quality Enriched Effort Estimation for Mobile Applications in Distributed Agile Software Development. *JOURNAL OF INFORMATION SCIENCE AND ENGINEERING*, Vol. 33: 1481–1500.
4. Awar K B, Sameem M S, Hafeez Y (2017). A Model for Applying Agile Practices in Distributed Environment: A Case of Local Software Industry. 2017 International Conference on Communication, Computing and Digital Systems (C-CODE), S 228–232. Islamabad: IEEE.
5. Bose I (2008) Lessons Learned from Distributed Agile Software. *Communications of the Association for Information Systems*, Vol. 23: 619–632.
6. Cichocki P, Maccari A (2008) Empirical Analysis of a Distributed Software Development Project. *IFIP Central and East European Conference on Software Engineering Techniques*, S 169–181. Poznan: Springer.
7. Cohen B, Thias M (2009). The Failure of the Off-shore Experiment: A Case for Collocated Agile Teams. 2009 Agile Conference, S 251–256. Chicago: IEEE.
8. Cottmeyer M (2008) The Good and Bad of Agile Offshore Development. *Agile 2008 Conference*, S 362–367. Toronto: IEEE.
9. Ghani I, Lim A, Hasnain M, Ghani I, Babar M I (2019). Challenges in Distributed Agile Software Development Environment: A Systematic Literature Review. *KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS*, Vol. 13: 4555–4571.
10. Hossain E, Babar M A, Verner J (2009) How Can Agile Practices Minimize Global Software Development Co-ordination Risks? *European Conference on Software Process Improvement*, S 81–92. Alcalá de Henares: Springer.

11. Kahya M D, Seneler Ç (2018) Geographical Distance Challenges in Distributed Agile Software Development: Case Study of a Global Company. 2018 3rd International Conference on Computer Science and Engineering (UBMK), S 78–83. Sarajevo: IEEE.
12. Kajko-Mattsson M, Azizyan G, Magarian M K (2010) Classes of Distributed Agile Development Problems. 2010 Agile Conference, S 51–58. Orlando: IEEE.
13. Korkala M, Maurer F (2014) Waste identification as the means for improving communication in globally distributed agile software development. *The Journal of Systems and Software*, Vol. 95: 122–140.
14. Lee S, Yong H-S (2010). Distributed agile: project management in a global environment. *Empirical Software Engineering*, Vol. 15: 204–217.
15. Lous P, Kuhrmann M, Tell P (2017). Is Scrum Fit for Global Software Engineering? 2017 IEEE 12th International Conference on Global Software Engineering (ICGSE), S 1–10. Buenos Aires: IEEE.
16. Lous P, Tell P, Michelsen C B, Dittrich Y, Allan E (2018) From Scrum to Agile: a Journey to Tackle the Challenges of Distributed Development in an Agile Team. International Conference on the Software and Systems Process 2018, S 11–20. Gothenburg: ACM.
17. Lous P, Tell P, Michelsen C B, Dittrich Y, Kuhrmann M, Ebdrup A (2018). Virtual by Design: How a Work Environment can Support Agile Distributed Software Development. 2018 IEEE/ACM 13th International Conference on Global Software Engineering (ICGSE), S 97–106. Gothenburg: IEEE.
18. Matalonga S, Solari M, Maturro G (2013). Factors affecting distributed agile projects: A systematic review. *International Journal of Software Engineering and Knowledge Engineering*, Vol. 23: 1289–1301.
19. Ozawa H, Zhang L (2013). Adapting Agile Methodology to Overcome Social Differences in Project Members. 2013 Agile Conference, S 82–87. Nashville: IEEE.
20. Paasivaara M, Durasiewicz S, Lassenius C (2008) Distributed Agile Development: Using Scrum in a Large Project. 2008 IEEE International Conference on Global Software Engineering, S 87–95. Bangalore: IEEE.
21. Paasivaara M, Durasiewicz S, Lassenius C (2009) Using Scrum in Distributed Agile Development: A Multiple Case Study. 2009 Fourth IEEE International Conference on Global Software Engineering, S 195–204. Limerick: IEEE.
22. Pries-Heje L, Pries-Heje J (2011) Why Scrum works: A case study from an agile distributed project in Denmark and India. 2011 Agile Conference, S 20–28. Salt Lake City: IEEE.
23. Ramesh B, Cao L, Mohan K, Xu P (2006) CAN DISTRIBUTED SOFTWARE DEVELOPMENT BE AGILE? *Communications of the ACM*, Vol. 49: 41–46.
24. Ramesh B, Mohan K, Cao L (2012). Ambidexterity in Agile Distributed Development: An Empirical Investigation. *Information Systems Research*, Vol. 23: 323–339.
25. Rizvi B, Bagheri E, Gasevic D (2015) A systematic review of distributed Agile software engineering. *Journal of Software: Evolution and Process*, Vol. 27: 723–762.

26. Rothman J, Hastie S (2013) Lessons Learned from Leading Workshops about Geographically Distributed Agile Teams. *IEEE Software*, Vol. 30: 7–10.
27. Shameem M, Kumar R R, Kumar C, Chandra B, Khan A A (2018). Prioritizing challenges of agile process in distributed software development environment using analytic hierarchy process. *Journal of Software: Evolution and Process*, Vol. 30: 1–19.
28. Shrivastava S V, Rathod U (2014) Categorization of risk factors for distributed agile projects. *Information and Software Technology*, Vol. 58: 373–387.
29. Shrivastava S V, Rathod U (2017) A risk management framework for distributed agile projects. *Information and Software Technology*, Vol. 85: 1–15.
30. Shrivastava S V, Rathod U (2019) A Goal-driven Risk Management Approach for Distributed Agile Development Projects. *Australasian Journal of Information Systems*, Vol. 23: 1–30.
31. Šmite D, Moe N B, Ågerfalk P J (2010) *Agility Across Time and Space*. Berlin Heidelberg: Springer.
32. Summers M (2008) Insights into an Agile Adventure with Offshore Partners. *Agile 2008 Conference*, S 333–338. Toronto: IEEE.
33. Sutherland J, Schoonheim G, Rustenburg E, Rijk M (2008) Fully Distributed Scrum: The Secret Sauce for Hyperproductive Offshored Development Teams. *Agile 2008 Conference*, S 339–344. Toronto: IEEE.
34. Therrien E (2008) Overcoming the Challenges of Building a Distributed Agile Organization. *Agile 2008 Conference*, S 368–372. Toronto: IEEE.
35. Vallon R, Dräger C, Zapletal A, Grechenig T (2014). Adapting to Changes in a Project's DNA: A Descriptive Case Study on the Effects of Transforming Agile Single-site to Distributed Software Development. *2014 Agile Conference*, S 52–60. Kissimmee: IEEE.
36. Vallon R, Strobl S, Bernhart M, Grechenig T (2013) Inter-organizational Co-development with Scrum: Experiences and Lessons Learned from a Distributed Corporate Development Environment. *International Conference on Agile Software Development*, S 150–164. Vienna: Springer.
37. Yadav V (2016). A Flexible Management Approach for Globally Distributed Software Projects. *Global Journal of Flexible Systems Management*, Vol. 17: 29–40.
38. Yap M (2005) Follow the Sun: Distributed Extreme Programming Development. *Agile Development Conference (ADC'05)*, S 218–224. Denver: IEEE.
39. Young C, Terashima H (2008) How Did We Adapt Agile Processes to Our Distributed Development? *Agile 2008 Conference*, S 304–309. Toronto: IEEE.
40. Zaghoul B, Riehle D, Zhou M (2015) Communication in Firm-Internal Global Software Development with China. *International Conference of Software Business*, S 132–138. Braga: Springer.

Literatur

- Alzoubi YI, Gill AQ, Al-Ani A (2016) Empirical studies of geographically distributed agile development communication challenges: a systematic review. *Inf Manag* 53:22–37

- Ames H (2020) What is video call fatigue? <https://www.medicalnewstoday.com/articles/video-call-fatigue>. Zugegriffen: 19. Juli 2020
- Aslam W, Ijaz F, Lali MI, Mehmood W (2017) Risk aware and quality enriched effort estimation for mobile applications in distributed agile software development. *J Inf Sci Eng* 33:1481–1500
- Bose I (2008) Lessons learned from distributed agile software. *Commun Assoc Inf Syst* 23:619–632
- Cichocki P, Maccari A (2008) Empirical analysis of a distributed software development project. IFIP Central and East European Conference on Software Engineering Techniques, S 169–181
- Cottmeyer M (2008) The good and bad of agile offshore development. *Agile 2008 Conference*, S 362–367
- Fischer T, Riedl R (2017) Stress durch E-Mail: Forschungsbefunde und Praxisimplikationen. *Wirtsch Inform Manag* 9:22–31
- Hossain E, Babar MA, Verner J (2009) How Can agile practices minimize global software development co-ordination risks? *European Conference on Software Process Improvement*, S 81–92
- Kahya MD, Seneler Ç (2018) Geographical distance challenges in distributed agile software development: case study of a global company. 2018 3rd International Conference on Computer Science and Engineering (UBMK), S 78–83
- Kajko-Mattsson M, Azizyan G, Magarian MK (2010) Classes of distributed agile development problems. 2010 Agile Conference, S 51–58
- Korkala M, Maurer F (2014) Waste identification as the means for improving communication in globally distributed agile software development. *J Syst Softw* 95:122–140
- Lous P, Tell P, Michelsen CB, Dittrich Y, Allan E (2018) From scrum to agile: a journey to tackle the challenges of distributed development in an agile team. *International Conference on the Software and Systems Process 2018*, S 11–20
- Paasivaara M, Durasiewicz S, Lassenius C (2008) Distributed agile development: using scrum in a large project. 2008 IEEE International Conference on Global Software Engineering, S 87–95
- Paasivaara M, Durasiewicz S, Lassenius C (2009) Using scrum in distributed agile development: a multiple case study. 2009 Fourth IEEE International Conference on Global Software Engineering, S 195–204
- Pries-Heje L, Pries-Heje J (2011) Why scrum works: a case study from an agile distributed project in Denmark and India. 2011 Agile Conference, S 20–28
- Ramesh B, Cao L, Mohan K, Xu P (2006) Can distributed software development be agile? *Commun ACM* 49:41–46
- Riedl R (2019) *Management von Informatik-Projekten: Digitale Transformation erfolgreich gestalten*, 2. Aufl. Oldenbourg, München
- Rizvi B, Bagheri E, Gasevic D (2015) A systematic review of distributed Agile software engineering. *J Softw Evol Proc* 27:723–762
- Rothman J, Hastie S (2013) Lessons learned from leading workshops about geographically distributed agile teams. *IEEE Softw* 30:7–10
- Shrivastava SV, Rathod U (2014) Categorization of risk factors for distributed agile projects. *Inf Softw Technol* 58:373–387
- Shrivastava SV, Rathod U (2017) A risk management framework for distributed agile projects. *Inf Softw Technol* 85:1–15
- Shrivastava SV, Rathod U (2019) A goal-driven risk management approach for distributed agile development projects. *Australas J Inf Syst* 23:1–30
- Šmite D, Moe NB, Ågerfalk PJ (2010) *Agility across time and space*. Springer, Berlin, Heidelberg, New York
- Summers M (2008) Insights into an agile adventure with offshore partners. *Agile 2008 Conference*, S 333–338
- Sutherland J, Schoonheim G, Rustenburg E, Rijk M (2008) Fully distributed scrum: the secret sauce for hyperproductive offshored development teams. *Agile 2008 Conference*, S 339–344
- Therrien E (2008) Overcoming the challenges of building a distributed agile organization. *Agile 2008 Conference*, S 368–372
- Vallon R, Strobl S, Bernhart M, Grechenig T (2013) Inter-organizational co-development with scrum: experiences and lessons learned from a distributed corporate development environment. *International Conference on Agile Software Development*, S 150–164
- Webster J, Watson RT (2002) Analyzing the past to prepare for the future: writing a literature review. *MIS Q* 26:13–23
- Wiederhold BK (2020) Connecting through technology during the coronavirus disease 2019 pandemic: avoiding “Zoom fatigue”. *Cyberpsychol Behav Soc Netw* 23:437–438
- Yap M (2005) Follow the sun: distributed extreme programming development. *Agile Development Conference (ADC’05)*, S 218–224

- Young C, Terashima H (2008) How did we adapt agile processes to our distributed development? Agile 2008 Conference, S 304–309
- Zaghloul B, Riehle D, Zhou M (2015) Communication in firm-internal global software development with China. International Conference of Software Business, S 132–138