

**Gottfried Wilhelm  
Leibniz Universität Hannover  
Fakultät für Elektrotechnik und Informatik  
Institut für Praktische Informatik  
Fachgebiet Software Engineering**

# **Analyse der Strukturen und Eigenschaften von agilen Organisationen**

**Analysis of the structures and characteristics of agile  
organisations**

**Masterarbeit**

im Studiengang Informatik

von

**Dorian Mahnke**

**Prüfer: Prof. Dr. Kurt Schneider**

**Zweitprüfer: Dr. Jil Klünder**

**Betreuer: Nils Prenner**

**Hannover, 19.10.2021**



# Erklärung der Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbständig und ohne fremde Hilfe verfasst und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel verwendet habe. Die Arbeit hat in gleicher oder ähnlicher Form noch keinem anderen Prüfungsamt vorgelegen.

Hannover, den 19.10.2021

---

Dorian Mahnke



# Zusammenfassung

Agile Organisation gelten allgemein als eine schlanke und flexible Organisationsstruktur, welche die Innovation des Unternehmens vorantreibt. In der Literatur gibt es bereits Ansätze, wie Unternehmen eine erfolgreiche agile Organisation aufbauen können.

In dieser Arbeit wird mit einer systematischen Literatursuche die Eigenschaften und der Aufbau von verschiedenen agilen Organisationstypen untersucht und überprüft, ob es Forschungslücke in bestimmten Bereichen zu diesem Themengebiet gibt. Es wird auch geprüft, ob es Kritik in der Literatur an den unterschiedlichen Organisationsmodellen gibt und welche Voraussetzungen eine Organisation erfüllen muss, um erfolgreich eine vollständig agile Organisation zu sein. Die verschiedenen Ansätze von agilen Organisationen, welche sich auch in der Praxis bereits bewährt haben, werden auf Gemeinsamkeiten untersucht.

Mit den Ergebnissen aus der Analyse der systematischen Literatursuche wird anschließend ein grundlegendes Modell für agile Organisationen erstellt, welches auf die wichtigsten Aspekte der Organisation eingeht.



# Abstract

Agile organization is generally considered to be a lean and flexible organizational structure that drives business innovation. There are already approaches in the literature on how companies can build a successful agile organization.

In this master thesis, a systematic literature search is conducted to examine the characteristics and structure of different types of agile organizations and to see if there is a research gap in certain areas of this topic. It is also examined whether there are criticisms in the literature of the different organizational models and what requirements an organization must meet in order to successfully be a fully agile organization. The different approaches of agile organizations, which have already proven themselves in practice, are examined for commonalities.

With the results from the analysis of the systematic literature search, a basic model for agile organizations is then created, which addresses the most important aspects of the organization.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Problemstellung . . . . .	2
1.2	Lösungsansatz . . . . .	2
1.3	Struktur der Arbeit . . . . .	2
<b>2</b>	<b>Grundlagen</b>	<b>5</b>
2.1	Systematische Literatursuche . . . . .	5
2.1.1	Systematic Mapping Prozess . . . . .	5
2.1.2	Snowballing Prozess . . . . .	7
2.2	Agile Frameworks . . . . .	7
2.2.1	SAFe . . . . .	7
2.2.2	Spotify Modell . . . . .	10
<b>3</b>	<b>Durchführung der systematischen Literatursuche</b>	<b>13</b>
3.1	Systematic Mapping Prozess . . . . .	13
3.2	Snowballing Prozess . . . . .	16
<b>4</b>	<b>Ergebnisse der Literatursuche</b>	<b>19</b>
4.1	Datenanalyse . . . . .	19
4.2	Forschungsfragen . . . . .	23
4.2.1	RQ1: Voraussetzungen und Eigenschaften einer agilen Organisation . . . . .	23
4.2.2	RQ2: Modelle für eine agilen Organisation . . . . .	25
4.2.3	RQ3: Einfluss der Modelle auf agile Organisation . . . . .	29
<b>5</b>	<b>Modell einer agilen Organisation</b>	<b>31</b>
5.1	Agilen Werte . . . . .	32
5.2	Struktur . . . . .	33
5.3	Strategie . . . . .	34
5.4	Kommunikation . . . . .	35
5.5	Prozess und Methodik . . . . .	35
<b>6</b>	<b>Verwandte Arbeiten</b>	<b>37</b>

<b>7 Zusammenfassung und Ausblick</b>	<b>39</b>
7.1 Zusammenfassung . . . . .	39
7.2 Ausblick . . . . .	39
<b>A Ausgewählten Publikationen der systematischen Literatursuche</b>	<b>45</b>

# Kapitel 1

## Einleitung

Die Globalisierung und Digitalisierung beschleunigen Markt- und Umweltveränderungen. Ereignisse, wie zum Beispiel die Corona Pandemie, können zu Problemen oder auch neuen Möglichkeiten für Unternehmen führen. Um die Probleme zu lösen oder die neuen Möglichkeiten umzusetzen, muss sich ein Unternehmen schnell verändern können. Den Unternehmen, die sich schnell an Unvorhergesehenes anpassen, gelingt es langfristig zu überleben und konkurrenzfähig zu bleiben [56]. Heutige Organisationen agieren in einem sehr wettbewerbsintensiven und turbulenten Marktumfeld. Das zunehmende Tempo von Innovation, technologischer Entwicklung und gestiegenen Kundenerwartungen an kundenspezifische Produkte führen zu turbulenten Veränderungen in der Unternehmensumgebung. Um bei diesem schnellen Tempo mithalten zu können, müssen die Unternehmen sich verändern, sie müssen agiler werden. Für die Unternehmen bedeutet das, dass sie sich in eine agile Organisation transformieren müssen, um das volle agile Potenzial auszuschöpfen. Das Konzept der Agilität wurde zuerst im Bereich der Softwareentwicklung angewendet, später wurde das Konzept auch auf andere Funktionen in einer Organisation übertragen, wodurch der Begriff agile Organisation geprägt wurde [47]. Die verschiedenen Definitionen sind sich einig, dass eine agile Organisation schnell auf überraschende und unvorhersehbare Ereignisse reagieren kann und die Fähigkeit besitzt, sich gut an neue Bedingungen anzupassen [47]. Die Ziele von Agilität fasst Bruckner [9] (S. 383) als „höhere Reagibilität auf (kritische) Entwicklungen in der Umwelt, gesteigerte Fähigkeit zur kontinuierlichen Selbstveränderung, schnellere Entscheidungsfindung, Nutzung verteilten Wissens und Selbstorganisation in netzwerkförmigen Strukturen“ zusammen. Beispiele für agile Organisationen sind Google und Facebook oder auch Wikipedia als Prototyp einer agilen netzwerkförmigen Struktur [9]. Studien, wie die von Worley aus dem Jahr 2014 [59] zeigen, dass agile Organisationen Vorteile gegenüber nicht-agilen Wettbewerbern haben, da sie sich besser an die komplexe Umwelt anpassen und somit ihre globale Wettbewerbsfähigkeit verbessern [43].

## 1.1 Problemstellung

Unternehmen versprechen sich viele Vorteile durch die Transformation zu einer agilen Organisation. Diese Vorteile, wie zum Beispiel schnelle und flexiblere Anpassung an den Markt, treten jedoch nur ein, wenn das Unternehmen das richtige agile Modell für ihrer Organisation auswählt und korrekt umsetzt. Für die Umsetzung ist es wichtig das Modell zu verstehen und welche Voraussetzungen und Eigenschaften für eine agile Organisation relevant sind.

Es gibt bereits viele Ansätze wie z.B. SAFe, die sich mit der Umsetzung und Transformation zu einer agilen Organisation beschäftigen. Einen umfassenden Überblick zu diesem Themengebiet gibt es noch nicht. Das Interesse an agilen Organisationen wächst jedoch stetig in der Forschung, wie auch in der Wirtschaft.

## 1.2 Lösungsansatz

In der Literatur und in der Praxis gibt es bereits viele Frameworks die Unternehmen unterstützen, sich in eine agile Organisation zu transformieren, bzw. neu zu gründen. Die Begriffserklärung einer agilen Organisation, beziehungsweise was diese beinhaltet, ist in der wissenschaftlichen Literatur nicht genau definiert. Mit der Durchführung einer systematischen Literaturrecherche zu den Voraussetzungen und Eigenschaften einer agile Organisation, werden mögliche Lösungen, wie auch Herausforderungen identifiziert, die in der Literatur erwähnt werden. Anschließend wird ein Modell entwickelt, welches den möglichen Aufbau und die Strukturen einer agilen Organisation zeigt. Dieses Modell dient dazu ein besseres Verständnis für agile Organisation zu erlangen, als auch eine grundlegende Idee, wie eine solche Organisation aufgebaut werden kann.

## 1.3 Struktur der Arbeit

Diese Arbeit ist wie folgt strukturiert:

In Kapitel 2 werden die Grundlagen dieser Arbeit beschrieben. Dazu gehört die Methodik der Systematische Literatursuche, sowie das Framework SAFe und das Spotify Modell.

In Kapitel 3 wird die Durchführung der Systematische Literatursuche behandelt.

In Kapitel 4 werden die Ergebnisse der Literatursuche präsentiert und auf die Forschungsfragen eingegangen.

In Kapitel 5 wird mit den Ergebnissen des vorherigen Kapitel ein allgemeines Modell einer agilen Organisation aufgestellt.

In Kapitel 6 werden Organisation betrachtet, die sich nicht mit der Richtung Software Entwicklung beschäftigen, sowie einen Ausblick für das Themengebiet der agile Organisation geben.

In Kapitel 7 werden die Resultate dieser Arbeit zusammengefasst und es gibt einen Ausblick zu weiteren möglichen Forschungsfragen zu agilen Organisationen.



# Kapitel 2

## Grundlagen

Im Grundlagenkapitel wird zuerst die systematische Literatursuche erklärt, anschließend wird auf die beiden agilen Frameworks SAFe und das Spotify Modell eingegangen.

### 2.1 Systematische Literatursuche

Für die systematische Literatursuche gibt es zwei Verfahren: Systematic Review und Mapping Studies. Mapping Studien werden häufig vor ein Systematic Review erstellt, um Ressourcen kosteneffizienter einsetzen zu können. Die Hauptunterschiede zwischen einer Mapping Studie und einem Systematic Review sind nach Keele et al. [28] vor allem, dass Mapping-Studien in der Regel umfassendere Forschungsfragen haben und oft mehrere Forschungsfragen gestellt werden und die Suchbegriffe weniger stark fokussieren, so dass eine größere Anzahl von Studien als Ergebnisse wahrscheinlicher wird. Der Datenextraktionsprozess ist für Mapping-Studien viel breiter angelegt als der Datenextraktionsprozess bei Systematic Reviews. Bei Systematic Reviews ist der Datenextraktionsprozess mehr eine Klassifizierungs- oder Kategorisierungsphase. Insgesamt zielt eine Mapping Studien auf darauf ab, die zukünftige Richtung der Primärforschung zu beeinflussen und ermöglicht die Identifizierung von Evidenzclustern und Evidenzwüsten, um den Fokus zukünftiger Systematic Reviews zu lenken und Bereiche zu identifizieren, in denen weitere Primärstudien durchgeführt werden sollten [28].

#### 2.1.1 Systematic Mapping Prozess

In Abbildung 2.1 ist der Prozess einer Systematic Mapping Studie nach Petersen et al. [41] abgebildet. Der obere Teil der Abbildung stellt einen Prozessschritt dar, dessen Ergebnis sich im unteren Teil der Abbildung befindet. Jeder Prozessschritt liefert ein Ergebnis, welches in den nächsten Prozess mit einfließt, mit Ausnahme des letzten Ergebnisses der *Systematic*

*Map*, welches das Hauptziel einer Systematic Mapping Studie ist.

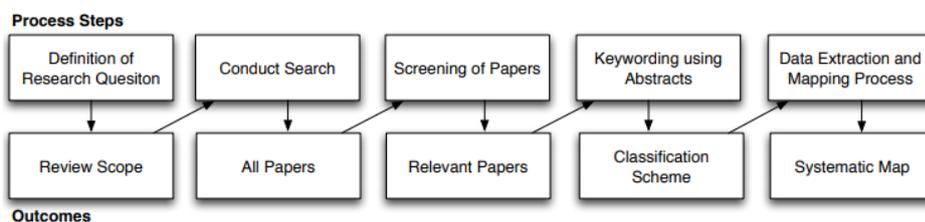


Abbildung 2.1: Der Systematic Mapping Prozess [41]

Der Prozess beginnt mit dem Definieren von Forschungsfragen, diese sollten so gestellt sein, dass ein Überblick über einen Forschungsbereich erlangt wird und die Menge und Art der in diesem Bereich verfügbaren Forschungsarbeiten und Ergebnisse ermitteln wird.

Der nächste Schritt ist die Durchführung der Literatursuche, dazu werden als erstes passende Suchbegriffe ermittelt, um einen Suchstring zu erstellen. Die passenden Suchbegriffe hängen von den zuvor erstellten Forschungsfragen ab. Es ist möglich nur bestimmte Arten von Studien zu berücksichtigen, Petersen et al. [41] rät davon ab, da der Überblick möglicherweise verzerrt und das mapping unvollständig ist. Das Ergebnis der Literatursuche sind die wissenschaftlichen Paper.

Die gefundene Literatur wird im nächsten Prozessschritt *Screening of Papers* auf Relevanz geprüft. Als Hilfsmittel werden Einschluss- und Ausschlusskriterien verwendet, um Studien auszuschließen, die für die Beantwortung der Forschungsfragen nicht relevant sind.

Die Abstracts der relevante Literatur werden gelesen und nach Schlüsselwörtern und Konzepten, die den Inhalt der Arbeit widerspiegeln, durchsucht. Wenn das Abstract von zu schlechter Qualität ist, um aussagekräftige Schlüsselwörter zu wählen, kann auch die Einleitung oder Schlussfolgerung der Arbeit untersucht werden.

Der letzte Schritt in dem Systematic Mapping Prozess ist der *Data Extraction and Mapping Process*. Für die Daten Extraktion wird das anlegen einer Excel-Tabelle empfohlen, in der jede der ausgewählten Paper in Kategorien einsortiert wird. Abschließend kann aus der Tabelle die Häufigkeit der Veröffentlichungen in jeder Kategorie berechnet werden. Auf diese Weise lässt sich erkennen, welche Kategorien in der bisherigen Forschung hervorgehoben wurden und somit Lücken und Möglichkeiten für die künftige Forschung aufzuzeigen. [41]

### 2.1.2 Snowballing Prozess

Snowballing ist in einer systematischen Literatursuche ein Prozess, um aus einer initialen Menge relevante Publikationen weitere thematisch verwandte Publikationen zu ermitteln. Als erster Schritt muss ein Startset erstellt werden. Das Startset besteht nur aus Publikationen, die in der endgültigen Analyse einbezogen werden sollen. Es ist wichtig, dass alle Publikationen vollständig geprüft wurden, da ein späterer Ausschluss eines Papers ein Rollback des Snowballing Prozesses zur Folge hat.

Der Snowballing Prozess unterteilt sich in *Backward Snowballing* und *Forward Snowballing*.

*Backward Snowballing* bedeutet, dass man die Referenzliste verwendet, um neue Arbeiten zu identifizieren, die aufgenommen werden sollen. Der erste Schritt besteht darin, die Referenzliste durchzugehen und die Arbeiten auszuschließen, die grundlegenden Kriterien nicht erfüllen, wie z. B. Sprache, Erscheinungsjahr und Art der Veröffentlichung (z.B. wenn nur Arbeiten mit Peer-Review berücksichtigt werden). Der nächste Schritt besteht darin, Paper aus der Liste zu entfernen, die bereits in dieser oder einer früheren Iteration durch das backward oder forward snowballing gefunden wurden.

*Forward Snowballing* bezieht sich auf die Identifizierung neuer Arbeiten derjenigen Arbeiten, die die zu untersuchende Arbeit zitieren. Wohlin [58] verwendet dafür die Datenbank Google Scholar, da diese über die Funktion verfügt eine Liste von Publikationen anzuzeigen, welche eine bestimmte Publikation zitiert haben. Anschließend wird, wie beim *Backward Snowballing*, die neue Literatur auf formale Kriterien und Relevanz überprüft.

Der Snowballing Prozess läuft in Iterationen ab, in jeder Iteration wird einmal Backward- und einmal Forward Snowballing durchlaufen. Üblicherweise endet der Prozess erst, wenn in einer Iteration keine neue Literatur gefunden wird. [58]

## 2.2 Agile Frameworks

Agile Frameworks sind Vorgehensmodelle für Unternehmen und dienen der Beschreibung von Organisations- und Prozessstrukturen. In dieser Arbeit werden die Begriffe Framework und Modell gleichwertig betrachtet.

### 2.2.1 SAFe

Das *Scaled Agile Framework (SAFe)* wurde 2012 von Dean Leffingwell für große Unternehmen entwickelt und dient als skalierter Ansatz für eine

agile Organisation. SAFe ist neben der Softwareentwicklung, auch in der Fertigungsindustrie und im Finanzdienstleistungssektor weit verbreitet. Von SAFe gibt es verschiedene Versionen, die aktuellste und in dieser Arbeit betrachtete Version ist SAFe 5.0.[46]

Das Framework verfügt über vier verschiedenen Konfigurationen: *Essential*, *Large*, *Portfolio* und *Full*.

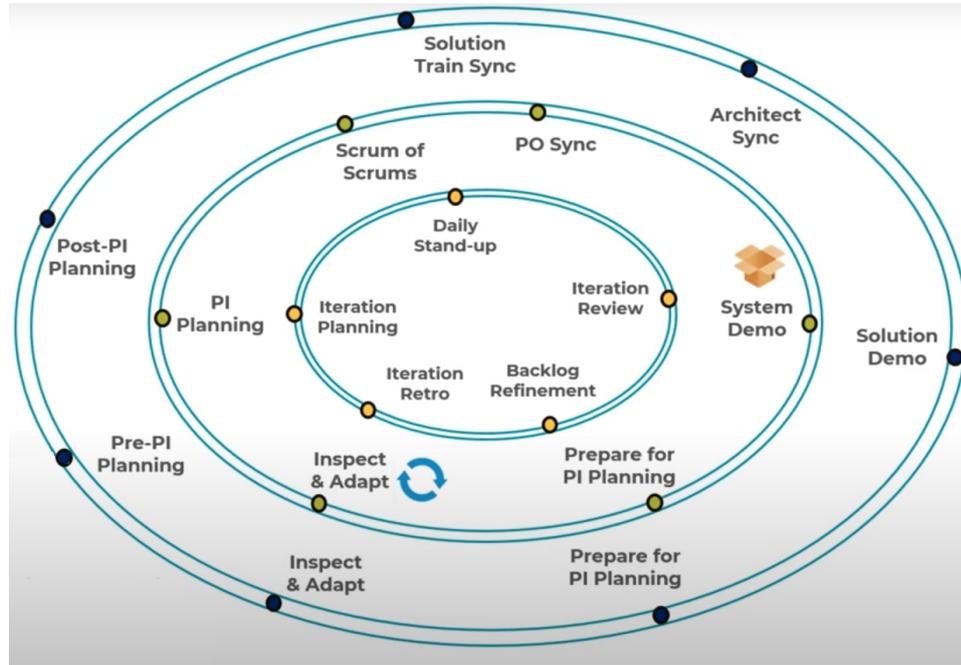


Abbildung 2.2: Aufbau des SAFe Modells [1]

Die innerste Iteration in Abbildung 2.2 ist die *Development Iteration*, welche auf ein einzelnes Team bezogen ist. Jedes Team hat auf der Teamebene eine eigene Iteration. Eine Iteration dauert ca. zwei Wochen und startet mit der Phase *Iteration Planning*. In dieser Phase werden die Ziele für die Iteration festgelegt, dazu wird überlegt, welche *Stories* aus dem *Team Backlog* umgesetzt werden sollen. Die Teammitglieder treffen sich täglich im *Daily Stand-up*, dies ist vergleichbar mit einem *Daily Scrum Meeting*, in dem alle Mitglieder über ihre aktuellen Aufgaben reden und Probleme ansprechen können. Während einer Iteration gibt es ein bis zwei Mal ein *Backlog Refinement*. Dabei können neue *User Stories* aufgenommen und bereits vorhandene aufgeteilt oder zusammengefasst werden, um somit neue Aufwandsschätzungen zu erstellen und Unklarheiten zu beseitigen. Am Ende einer Iteration gibt es ein *Iteration Review*, in diesem zeigt das Team dem *Produkt Owner* den gemachten Fortschritt. Im Anschluss gibt es eine *Iteration Retrospective*, in der Abbildung als *Iteration Retro* bezeichnet.

Die Retrospektive dient dazu, die genutzten Praktiken zu bewerten und zu verbessern. Mit dem Abschluss der Iteration hat das agile Team ein qualitativ hochwertiges, funktionierendes und getestetes Stück Software erstellt. Diese innerste Iteration aus Abbildung 2.2 ist das Kernstück von SAFe und Teil jeder Konfiguration. Unternehmen, die nur diese Iteration anwenden, verwenden *Essential SAFe*.

Die mittlere Iteration in Abbildung 2.2 ist die *Innovation and Planning Iteration* und besteht meistens aus vier *Development Iterationen*. Die Teams werden zu einem *Agile Release Train* (ART) zusammengefasst, auch „*Team-of-Teams*“ genannt. Die Teams eines *Agile Release Trains* sind durch eine gemeinsame Version und das *Program Backlog* verbunden. Die Iteration startet immer mit einem *PI Planning* Event. Die Teams schätzen ab, welche Ziele sie erreichen können und welche Abhängigkeiten zwischen den Teams und dem „*Train*“ entstehen. *Scrum of Scrums* ist ein wöchentliches Meeting, das von der Rolle *Release Train Engineer* geleitet wird. In diesem Meeting treffen sich Vertreter der einzelnen Teams, um die Fortschritte bei den Meilensteinen und PI-Zielen, sowie die Abhängigkeiten zwischen den Teams zu überprüfen. *Product Owner Sync* (PO Sync) hat große Ähnlichkeiten mit dem *Scrum of Scrums Meeting*, daher können beide Meetings auch zu einem *ART Sync* zusammengefasst werden. Im PO Sync wird sich ein Überblick darüber verschafft, wie gut die ART bei der Erfüllung ihrer PI-Ziele vorankommt und um mögliche Probleme zu lösen. Die *System Demo* zeigt den Fortschritt eines Program Increments und beinhaltet alle neuen Funktionen, die in der Iteration erstellt wurden. Es dient als Maß für die Geschwindigkeit und den Fortschritt der vollständig integrierten Arbeit in allen Teams. Das Ereignis *Inspect and Adapt* dient den Teams zum reflektieren und identifizieren von Verbesserungspunkten im *Backlog*. Die *Innovation and Planning Iteration* und die darunter liegende *Development Iteration* bilden zusammen das *Enterprise Level* ab.

Mit der äußersten Iteration wird ein *Solution Train* („*Team-of-Teams-of-Teams*“) eingeführt, dieser besteht aus zwei oder mehr *Agile Release Trains* und bildet die *Large SAFe* Konfiguration. Die neu entstandene Iteration besteht aus fast den gleichen Events, wie in der *Innovation and Planning Iteration*, nur auf mehrere *Agile Release Trains* hoch skaliert, die Bedeutung und die Aufgaben bleiben jedoch gleich.

Die Portfolio Konfiguration beinhaltet *Essential SAFe* und ist die kleinste Konfiguration, die zur Erreichung von *Business Agility* eingesetzt werden kann. *Business Agility* wird durch *Value Streams* erreicht. *Value Streams* stellen die Reihe von Schritten dar, die ein Unternehmen durchführt, um Lösungen zu implementieren, die einen kontinuierlichen Wertfluss für einen Kunden bieten [3].

In SAFe gibt es zwei Arten von *Value Streams*:

- *Operational Value Stream*: Abfolge von Aktivitäten, die erforderlich sind, um ein Produkt oder eine Dienstleistung an einen Kunden zu liefern
- *Development Value Stream*: Abfolge von Aktivitäten, welche die *Operational Value Streams* bei ihrer Wertschöpfung unterstützen.

*Full SAFe* ist eine Kombination aus den drei Konfigurationen *Essential SAFe*, *Large SAFe* und *Portfolio SAFe* und die größtmögliche Umsetzung von SAFe.[1]

### 2.2.2 Spotify Modell

Das Spotify Modell wurde von Henrik Kniberg, Anders Ivarsson und Joakim Sundén entwickelt und 2012 in dem Unternehmen Spotify eingeführt [52].

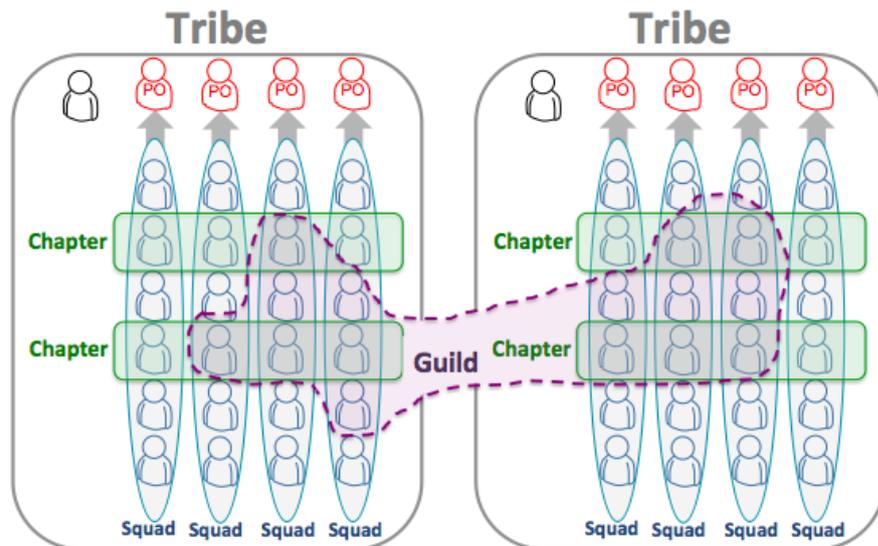


Abbildung 2.3: Aufbau des Spotify Modells

Als kleinste organisatorische Einheit gibt es in dem Spotify Modell das **Squad**. Dieses trägt die gesamte Verantwortung für ein zu entwickelndes Feature. Es ist ein kleines selbstorganisiertes Team, welches über alle nötigen Ressourcen verfügt, um Features zu entwerfen, zu entwickeln und zu testen. Jedes Team darf frei entscheiden, welche Methode sie verwenden möchten, häufig wird Scrum angewendet, andere verwenden Kanban und

oder eine Kombination aus beidem [5]. Die Mitglieder arbeiten normalerweise zusammen an einem Ort und sind in verschiedene Rollen eingeteilt. Nach Gerster et al. [21] gibt es sechs verschiedene Rollen innerhalb eines Squads:

- **Product owner:** Vertritt den Kunden, setzt Prioritäten und definiert und akzeptiert die Features des Produktes
- **Technical product owner:** Stellt den geschäftlichen Nutzen des Produktes sicher und hat die technische Gesamtverantwortung
- **Scrum master:** Ist für die Umsetzung von Scrum verantwortlich und unterstützt das Team
- **Agile coach:** Berät das Team beim Erlernen von agilen Praktiken
- **Operative team members:** Sind Verantwortlich für die Umsetzung des Designs, Entwicklung, Testens, Integration, Wartung und den Betrieb des Produkts
- **Experts:** Sind keine ständigen Mitglieder des Squads und werden als Unterstützung für eine spezifische Aufgabe hinzugezogen

Von den Rollen *Product Owner*, *Technical Product Owner* und *Scrum Master* gibt es einen Vertreter pro Squad, ein *Agile Coach* betreut meist mehrere Squads in Abhängigkeit von ihrer Größe und Erfahrung in agilen Praktiken. *Operative Team Members* gibt es mehrere pro Squad und die Rolle des Experts kann es beliebig oft oder auch keinen geben. Ein **Tribe** besteht aus acht bis zwölf Squads und umfasst ein oder mehrere Produkte aus einem Produktbereich. Geleitet wird ein Tribe von dem *Area Product Owner*, dieser ist für konsistente Produktmerkmale in allen Produkten des entsprechenden Produktbereichs verantwortlich. Dazu gibt es regelmäßige Abstimmung zwischen den *Product Ownern* der Squads. Ein **Chapter** besteht aus den Mitgliedern der Squads mit den gleichen Aufgaben, unterschieden wird in funktionale Fähigkeiten (z.B. Testen) oder technische Fähigkeiten (z. B. maschinelles Lernen). Es gibt regelmäßige Treffen, um über ihr Fachgebiet oder spezifische Probleme zu diskutieren. Chapters sind immer lokal an einen Tribe gebunden, während **Gilden** sich in der Regel über die gesamte Organisation erstrecken [5]. Gilden oder auch *Communities of Practice* (CoPs) genannt, sind Gemeinschaften von Mitgliedern mit gemeinsamen Interessen. Sie dienen dem informellen Austausch in der gesamten Organisation [21]. Für die Mitarbeiter ist die Teilnahme freiwillig, sie können jederzeit wechseln oder auch Mitglied in mehreren Gilden gleichzeitig sein. Die Kommunikation erfolgt über GMail, Slack und regelmäßige Treffen, wobei einige Gilden sich häufiger treffen und andere nur einmal pro Jahr [49].



## Kapitel 3

# Durchführung der systematischen Literatursuche

In dieser Arbeit wurde das Verfahren der Systematischen Literatursuche angewendet, welches dazu dient einen umfassenden Überblick über das Forschungsthema *agile Organisation* zu erlangen und um festzustellen, ob es Forschungsergebnisse gibt. Um relevante Literatur zu identifizieren wurde eine *Systematic Mapping Study*, nach Vorlage von Petersen et al., angewandt [41]. Da das Themengebiet der agilen Organisation noch nicht weit erforscht ist, wird eine Mapping Study empfohlen, ansonsten ein systematic Review [28].

### 3.1 Systematic Mapping Prozess

Der erste Schritt im Systematic Mapping Prozess nach Petersen et al. (vgl. Grundlagen Kapitel 2) ist das Definieren von Forschungsfragen, welche so gestellt werden, dass ein Überblick über den Forschungsbereich geschaffen wird.

**RQ1:** *Welche Voraussetzungen und Eigenschaften benötigt eine agile Organisation?*

Die erste Forschungsfrage bieten einen Einblick in das Themengebiet der agile Organisationen und zeigt die in der Literatur vorkommenden grundlegenden Voraussetzungen und Eigenschaften einer agilen Organisation. Mit den Ergebnissen dieser Forschungsfrage gibt es ein besseres Verständnis für die Grundlagen einer agilen Organisation.

**RQ2:** *Welche Modelle für eine agile Organisation gibt es?*

Die zweite Forschungsfrage fasst die Menge der bereits vorhandenen Modelle zusammen und macht diese vergleichbar. Es können Ähnlichkeiten zwischen

den Modellen erkannt werden, wie z.B. eine ähnlicher Aufbau oder die Verwendung gleicher Methodiken.

**RQ3:** *Zu welchem Grad führen die Modelle zu einer agilen Organisation?*

Die letzte Forschungsfrage hinterfragt die einzelnen Modelle und zeigt warum ein Modell besser oder schlechter geeignet ist, um eine agile Organisation zu erstellen. Für die Beantwortung dieser Frage wird auch geprüft, ob es Kritik an den Modellen aus der wissenschaftlichen Literatur gibt.

Der nächste Schritt im *Systematic Mapping Prozess* ist die Durchführung der Literatursuche. Die systematische Literatursuche benötigt einen Suchstring, dieser wird anhand abgeleiteter Keywords aus den Forschungsfragen generiert. Die abgeleiteten Keywords sind: *agile, organization, structure*. Diese Keywords wurden durch weitere Synonyme ergänzt, um auch Literatur zu erfassen, welche eine andere Terminologie verwenden. Das Keyword *organization* wurde durch die Synonyme *company, enterprise* erweitert und das Keyword *structure* mit dem Wort *construction*. Für die Erstellung des Suchstrings wurden die Keywords mit boolesche Operatoren verbunden, wobei die Operatoren AND und OR verwendet wurden. AND dient für Keywords, die in der Suche vorkommen müssen und OR wurde zwischen die Synonyme gesetzt, da mindestens eins dieser Keywords vorkommen soll.

Der Finale Suchstring lautet *agile AND (organization OR company OR enterprise) AND (construction OR structure)*.

Mit diesem Suchstring wurde in den Datenbanken Google Scholar, IEEE, ACM, Science Direct und Springer Link nach geeigneter Literatur gesucht. Im nächsten Prozessschritt wurden die Publikationen der Datenbanksuche auf Relevanz überprüft, wozu Inklusions- und Exklusionskriterien erstellt wurden. Folgende Kriterien wurden bei der systematischen Literatursuche angewendet:

Inklusion:

- der Aufbau einer agilen Organisation wird beschrieben
- es wird zwischen agiler und nicht agiler Organisation verglichen
- es wird eine „case study“ oder „experience support“ einer agilen Organisation beschrieben

Exklusion:

- es handelt sich um keine „Software“ Organisation
- das Paper bezieht sich ausschließlich auf agile Teams oder Projekte (bzw. nicht agil auf allen Ebenen)

- die Publikation ist weder auf englisch noch auf deutsch geschrieben
- die Publikation wurde nicht durch ein peer-review überprüft

Im nächsten Schritt „*Keywording using Abstracts*“ wurden die relevanten Publikationen aus dem vorherigem Prozessschritt weiter gefiltert. Dafür wurden die *Abstracts* gelesen und nach Schlüsselwörtern und Konzepten gesucht, die bei der Beantwortung der Forschungsfragen helfen. Eignet sich die Zusammenfassung der Publikation nicht zur Auswahl aussagekräftiger Schlüsselwörter, wurden ebenfalls Einleitung oder Schlussfolgerung der Arbeit untersucht. Im letzten Schritt findet der Daten-Extraktionsprozess statt. In einer Exel-Tabelle wurden alle relevanten Publikationen aufgelistet und wichtige Informationen in verschiedenen Kategorien gesammelt. Zu den zunächst gesammelten Informationen zählen: Titel der Publikation, Autoren, Veröffentlichungsjahr sowie die Datenbank aus der die Quelle stammt. Die anschließend inhaltlichen gesammelten Informationen umfassen: Methoden der Datenerfassung, verwendetes Framework, Unternehmensgröße, als auch Eigenschaften und Informationen zum Aufbau einer agilen Organisation.

Datenbank	Gefundene Paper	Aussortiert nach Abstract	Aussortiert nach Inhalt	Ausgewählte Paper
Google Scholar	48	25	10	13
IEEE	5	2	2	1
ACM	12	6	2	4
Science Direct	12	7	3	2
Springer Link	13	5	3	5
Gesamt	90	44	20	26

Tabelle 3.1: Aufbau des Startset der Literatursuche

### 3.2 Snowballing Prozess

In Tabelle 3.1 ist der erste Teil der systematischen Literatur abgebildet. Um die Anzahl der Publikationen in der Literatursuche zu vergrößern wurde ein *Snowballing* Prozess nach Vorlage von Wohlin [58], wie in den Grundlagen (Kapitel 2) beschrieben, durchgeführt. Die bereits gefundene Literatur diente dabei als Startset (vgl. Tabelle 3.1). Für das *Backward Snowballing* wurde von allen Publikationen des Startsets die Referenzliste durchgegangen und nach den Inclusion- und Exclusionskriterien, die auch schon für das Startset verwendet wurden, Literatur ausgewählt.

Bei dem *Forward Snowballing* wurden die Publikationen des Startsets in Google Scholar gesucht und mit der von der Datenbank bereitgestellten Funktion, sich die Literatur anzeigen zulassen, die das gesuchte Paper zitieren, genutzt. Wie auch beim *Backward Snowballing* wurde mit den Inclusion- und Exclusionskriterien vorsortiert. Anschließend wurde wie im Systematic Mapping Prozess die Zusammenfassung der relevanten Literatur gelesen und nach Schlüsselwörtern und Konzepten durchsucht.

1. Iteration	Gefundene Paper	Aussortiert nach Abstract	Aussortiert nach Inhalt	Ausgewählte Paper
Rückwärtssuche	25	8	5	12
Vorwärtssuche	9	2	2	5
Gesamt	34	10	7	17

Tabelle 3.2: Übersicht der weiteren Iteration der systematischen Literatursuche

Die Tabelle 3.2 zeigt die neue Iteration des *Snowballings*. Bei der *Snowballing* Suche wurde ausschließlich mit der Datenbank Google Scholar gesucht, da weitere Datenbanken keinen Mehrwert bieten bzw. nicht über die geeigneten Funktionen verfügen.

Nach der Vorlage von Wohlin endet der Snowballingprozess erst, wenn in einer Iteration keine neuen Publikationen hinzugefügt werden. Aufgrund der zeitlichen Begrenzung dieser Arbeit wurde nach der ersten Iteration der Prozess gestoppt.

In Abbildung 3.1 ist der gesamte Ablauf der systematischen Literatursuche abgebildet. Das Startset hat einen Gesamtanteil von 26 Publikationen und das Snowballing zusätzlich 17 Publikation, so dass zum Abschluss der Suche 43 Quellen vorliegen.

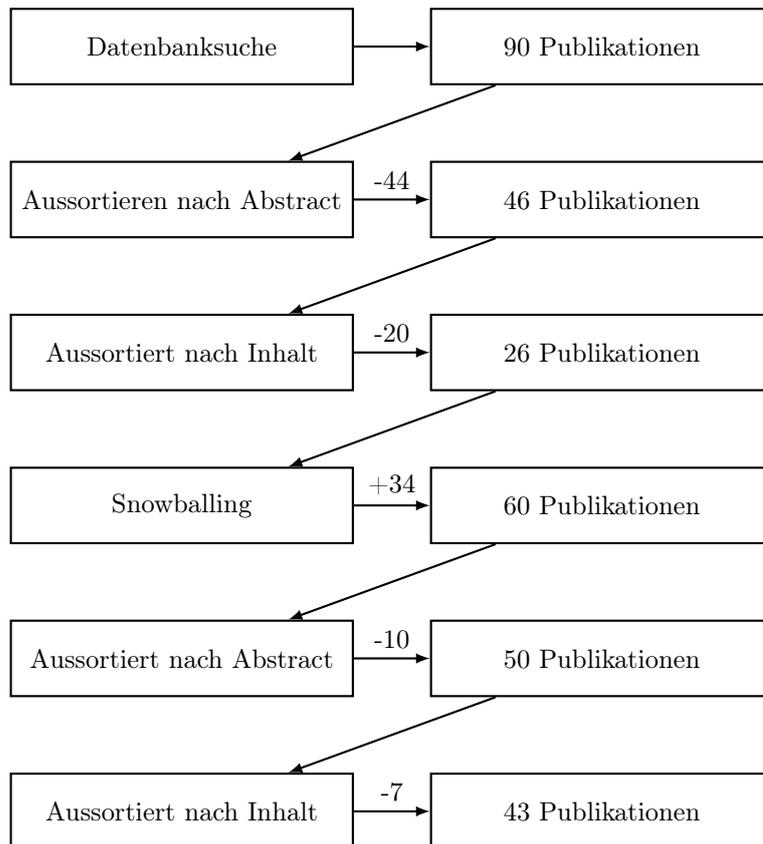


Abbildung 3.1: Ablauf der gesamten Literatursuche



## Kapitel 4

# Ergebnisse der Literatursuche

Dieses Kapitel beinhaltet die Auswertung der Systematischen Literatursuche. Zunächst eine generelle Analyse der ausgewählten Literatur, danach folgt die Analyse bezogen auf die einzelnen Forschungsfragen.

### 4.1 Datenanalyse

Wie aus dem Durchführungskapitel (Kapitel 3) zu entnehmen ist, wurden aus dem Startset und einer weiteren Iteration der *Systematic Mapping Study* insgesamt 43 Paper ausgewählt. Der weitaus größte Teil der Literatur stammt aus der Datenbank von Google Scholar. Aus den Datenbanken ACM und Springer Link wurden gleichmäßig viele Quellen entnommen, die Datenbanken von IEEE und Science Direct hat den geringsten Anteil (vgl. Kapitel 3 Tabelle 3.1).

In der wissenschaftlichen Literatur wurden verschiedenen Verfahren der Datenerfassung angewandt. Es gibt insgesamt acht Veröffentlichungen mit *Case Studys*, drei davon mit *Multiple-Case Study* (vgl. Tabelle 4.1). In allen *Case Study* wurde die Transformation von Unternehmen zu einer agilen Organisation untersucht. Häufig verbunden mit einer *Case Study* ist ein Interview mit den Unternehmen, um die Beweggründe für die Umwandlung zu erfassen und über die Probleme und den Erfolg der Transformation zu sprechen. Nicht alle Publikationen mit *Case Studys* haben Angaben zu den untersuchten Unternehmen gemacht, so wurde teilweise mit Pseudonymen gearbeitet und die Größe des Unternehmens nicht angegeben.

Name des Unternehmens	Größe des Unternehmens <sup>1</sup>	Framework	Branche
SIMCorp [42]	500 Mitarbeiter in 55 Teams (8ART)	SAFe 4.0 min. Portfolio SAFe	IT
SunGard [37]	groß	Model Driven Development (MDD)	Finanzen
CarCo-Drive [21]	100.000+	Spotify Modell (2, 4)	Automobilindustrie
CarCo-IT [21]	100.000+	Spotify Modell (2, 3)	Automobilindustrie
Chemco [21]	100.000+	Spotify Modell (3)	Chemie
EnergyCo [21]	40.000+	Spotify Modell (1, 3)	Energie
FinCo [21]	65.000+	Spotify Modell (2)	Finanzen
InsureCo [21]	4.000+	Spotify Modell (2)	Versicherung
MedtechCo [21]	100.000+	Spotify Modell (2)	Medizin
RailCo [21]	100.000+	Spotify Modell (1, 2)	Transport
RetailCo [21]	50.000+	Spotify Modell (2)	Einzelhandel
ReinsureCo [21]	40.000+	Spotify Modell (1, 3)	Rückversicherung
ServicesCo [21]	16.000+	Spotify Modell (2)	Services
SoftwareCo [21]	100.000+	Spotify Modell (2)	IT
TelCo [21]	17.000+	Spotify Modell (2)	Telekommunikation
ToolsCo [21]	70.000+	Spotify Modell (2)	Herstellung
VehicleCo [21]	100.000+	Spotify Modell (1)	Automobilindustrie
TechCo [13]	340+	SAFe	IT
Dell [13]	180+	Scrum@Scale	IT
FinanceCo [13]	1.250+	Spotify Modell	Finanzen
SemiCo [13]	1.500+	LeSS	IT
Ericsson [13]	200+	Scrum@Scale	IT
RevenueCo [13]	200+	Spotify, SAFe	IT
BankCo [13]	150+	LeSS	Finanzen

Tabelle 4.1: Übersicht von Unternehmen und dem verwendeten Framework

Das Unternehmen SIMCorp (1. Zeile in Tabelle 4.1) hat nicht die Konfiguration von dem verwendeten SAFe Framework angeben. Es wird aber mitgeteilt, dass bei der Transformation zu SAFe, die ehemaligen Manager sich für die SAFe-Stellen als „Train Conductor“, „Portfolio Manager“ oder „Scrum Master“ bewerben mussten, wodurch anzunehmen ist, dass mindestens Portfolio SAFe, vielleicht auch Full-SAFe benutzt wurde [42].

<sup>1</sup>Keine genaue Angaben zur Größe des Unternehmens ([37]), Gesamtanzahl von Mitarbeitern ([21]), Gesamtanzahl von Entwicklern ([13], [42]).

Bei der Spotify *Multiple-Case Study* wurden in der Studie das Modell in vier verschiedene Varianten aufgeteilt. In Tabelle 4.1) wird die Zugehörigkeit der verschiedenen Spotify Modelle durch die Zahl hinter dem Frameworknamen ausgedrückt. Auf die verschiedenen Varianten wird in der dritten Forschungsfrage weiter eingegangen.

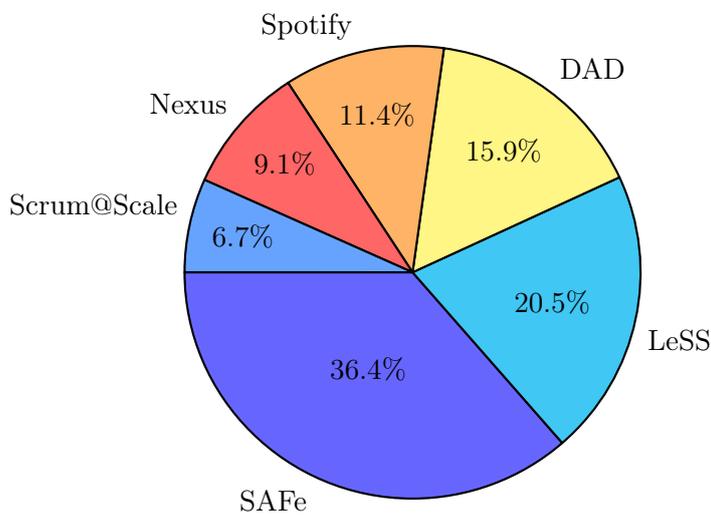


Abbildung 4.1: Anteile agiler Frameworks in der Literatur

Die Verteilung der vorkommenden agilen Frameworks ist in Abbildung 4.1 zu sehen. Neben den SAFe- und Spotify Modellen, die meist den Hauptteil einer Publikation ausmachen, gibt es die Frameworks LeSS, DAD, Nexus und Scrum@Scale, welche meist nur Teil eines Systematic Reviews waren und insgesamt nur wenig angesprochen wurden. Beide Fälle sind gleichermaßen für die Abbildung erfasst worden. Wurde ein Framework nur erwähnt und nicht weiter darauf eingegangen, so ist es nicht gewertet worden.

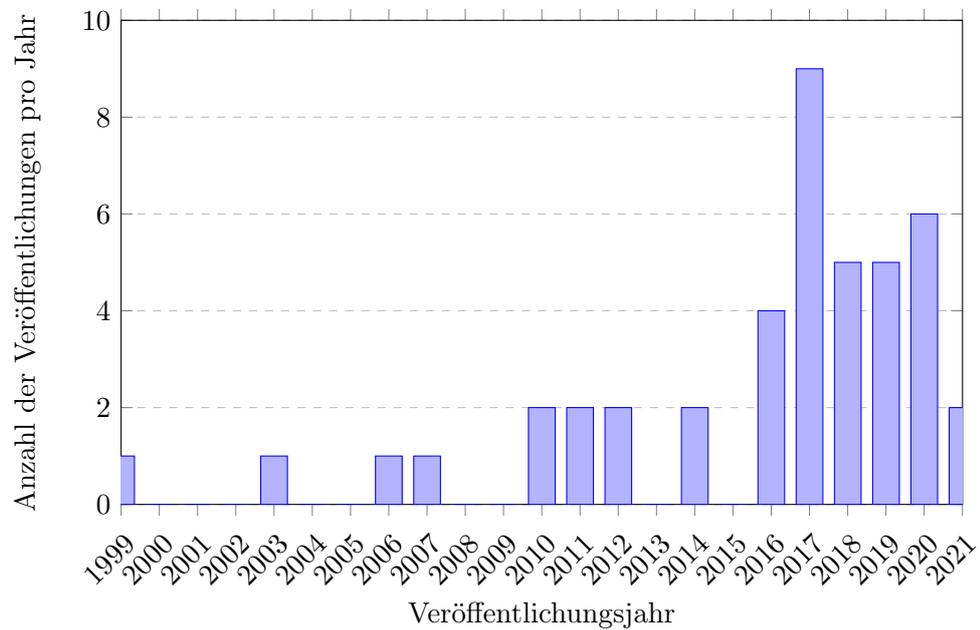


Abbildung 4.2: Übersicht über die Veröffentlichungsjahre der Literatur

Die Abbildung Abbildung 4.2 zeigt die Anzahl der Veröffentlichungen pro Jahr. Die älteste Veröffentlichung erschien 1999 mit dem Titel „*KWM: Knowledge-based Workflow Model for Agile Organization*“ und beschreibt einen Mechanismus zur Verwaltung von Änderungen innerhalb der Organisationsstruktur und von Unternehmensregeln [31]. In dieser Veröffentlichung wurde auf kein agiles Framework eingegangen, da es zu diesem Zeitpunkt noch kaum relevante Frameworks gab. Die Anzahl der Veröffentlichungen stieg erst ab dem Jahr 2010, als die Veröffentlichung neuer Frameworks deutlich zunahm [52]. Ab 2016, nachdem die Frameworks auch in den vielen Unternehmen angewendet wurden, stieg die Veröffentlichungsrate deutlich an. Bemerkbar ist dies auch daran, dass die Menge an Studien mit *Case Study* in diesem Zeitraum zunahm. Der Rückgang auf zwei Veröffentlichungen pro Jahr im Jahre 2021 hängt damit zusammen, dass die *Systematic Mapping Study* in der ersten Hälfte dieses Jahres durchgeführt wurde. Von einem plötzlichen Desinteresse an diesem Themengebiet ist vorläufig nicht auszugehen.

## 4.2 Forschungsfragen

In diesem Abschnitt werden die Ergebnisse der ausgewählten Literatur aus der Systematischen Literatursuche ausgewertet. Die zu Beginn der Arbeit vorgestellten Forschungsfragen werden in den folgenden Unterkapiteln behandelt.

### 4.2.1 RQ1: Voraussetzungen und Eigenschaften einer agilen Organisation

Die wichtigste Voraussetzung für eine erfolgreiche agile Organisation ist der Wille des Unternehmens, sowohl im Managerbereich, wie auch auf Teamebene, sich auf agile Arbeitsweisen einzulassen [8][42]. Ohne die Entscheidung und den Willen eine agile Organisation aufbauen zu wollen wird auch nicht mit der Umsetzung begonnen.

Bei den folgenden Voraussetzungen und Eigenschaften einer agilen Organisation gibt es keine Priorisierung.

#### Agiles Mindset

Ein agiles Mindset bedeutet, dass die Mitarbeiter ein Verständnis für die agilen Werte des Unternehmens entwickeln und das Konzept des Frameworks verstanden haben und anwenden können. Conboy und Carroll [13] haben bei der Auswertung ihrer *Case Study* herausgefunden, dass Unstimmigkeiten auf Teamebene schnell behoben werden können, aber Unterschiede in einer großen Anzahl von Teams sich verfestigen. Eine weitere Feststellung war, dass die Entwickler ohne klare Vorgaben zu einer agilen Organisation, nach einer Transformation weiter arbeiteten wie bisher. [13].

In einer weiteren *Case Study* wird das Aufbauen eines agilen Mindset als eine der drei größten Herausforderungen angesehen, denn nur auf der Grundlage von Experimenten und Fehlschlägen sei eine kontinuierliche Verbesserung möglich [42].

Paasivaara et al. stellten in ihrer Studie fest, dass Organisationen sich vor der Einführung eines Frameworks auf die Entwicklung eines agilen Mindsets und das Verständnis agiler Prinzipien konzentrieren müssen [39].

#### Selbstorganisierend

Tyszkiewicz und Pawlak-Wolanin definieren selbstorganisierend als „spontanen, nicht choreografierten, auf Feedback basierenden Austausch, der in agilen Unternehmen häufig anzutreffen ist“ [51]. Die Eigenschaft „selbstorganisierend“ ist Teil des agilen Mindsets und für

agile Organisationen von wichtiger Bedeutung für die Trennung zu traditionellen Modellen. In agilen Organisationen treffen alle Beteiligten gemeinsam Entscheidungen, ohne dass sie von außen geleitet werden [51]. Selbstorganisierende Teams haben die Eigenschaft schneller und effizienter zu sein, als Teams mit einer „Command-and-Control-Philosophie“ [50].

### **Katalytische Führung**

Bei einer katalytischen Führung hat die Führungskraft keine Weisungsbefugnis und Ausführungskontrolle über die Gruppe, sondern stellt sicher, dass die Bedingungen für eine zielführende Selbstorganisation sichergestellt ist [12]. Der Transformationsprozess in der Führungsabteilung wird von Vyver et al. [55] als *Devolution of Authority* bezeichnet, dabei wird beobachtet, dass es den Managern nach einer Transformation zu einer agilen Organisation häufig schwer fällt, sich nicht mehr wie ein „Chef“ zu verhalten [15].

### **Koevolution**

Mit Koevolution ist die ständige Weiterentwicklung der Organisation auf Grund wechselnder Umwelteinflüsse gemeint, wobei dazu gehört, dass aus Erfahrungen gelernt und Anpassungen vorgenommen werden. Häufige Ursachen für Anpassungen sind die Reaktion auf externe Umweltfaktoren, wie z.B. Produkte von Wettbewerbern, Regulierungsbehörden, Lieferanten und Kundenreaktionen [51].

### **Offene Kommunikation**

Nach Coldewey [12] muss eine Kommunikationsstruktur errichtet werden, die agile Ansätze und Selbstorganisation ermöglicht. Sie muss Informationen über das operative Geschäft, wie auch unerwartete und gelegentlich mal schmerzliche Informationen transportieren können. Dazu ist es notwendig, dass die Entwickler ein einheitliches Verständnis für die Prozesse innerhalb des Agilen Frameworks besitzen [11].

Je größer eine Organisation ist, desto komplexer wird die Kommunikation, so dass es in größeren Softwareunternehmen zu Verzögerungen bei der Entscheidungsfindung kommt. Diese führen zu weniger Flexibilität und Agilität. Kettunen [29] verweist auf das SAFe Framework, in dem es eine Reihe von Rollen gibt, welche die Kommunikation unterstützen. So ist die Aufgabe des *Product Owner* und *Product Manager*, dass mindestens zweimal pro Woche über die Inhalte des *Backlogs*

kommuniziert wird (vgl. Grundlagen Kapitel 2). Ein weiteres Beispiel ist die Produktintegrationsplanung, bei der sich die Mitarbeiter beraten, welche Aufgaben in den nächsten Wochen umgesetzt werden.[29]

### **Langfristiges, ergebnisorientiertes Controlling**

Coldwey [12] ist der Meinung, dass budgetorientierte Controlling-Ansätze sich wenig eignen, da nur kurzfristige Kennzahlen im Fokus sind, während ein ergebnisorientierter Ansatz auf langfristige Gewinne ausgelegt ist und auch kurz- wie mittelfristige Durststrecken aushalten kann. Tyszkiewicz und Pawlak-Wolanin widersprechen dieser Aussage teilweise und bewerten Organisationen, die eine kurzfristige Chance nach der anderen verfolgen, als genauso agil wie Organisationen, die den langfristigen Nutzen kurzfristiger Chancen bewerten und eine fundierte Entscheidung getroffen haben, ob sie die Idee verfolgen oder fallen lassen [51].

Doz und Kosonen [16] (S.371) definieren strategische Agilität als ein „durchdachtes und zielgerichtetes Zusammenspiel“ des Top-Managements zwischen den drei „Meta-Fähigkeiten“: *Strategische Sensibilität*, *Einheitlichkeit der Führung* und *Fluidität der Ressourcen*.

*Strategische Sensibilität* bedeutet, dass die Manager aufmerksam die strategische Entwicklungen des Unternehmens beobachten und nach neuen Chancen und Möglichkeiten suchen. *Einheitlichkeit der Führung* ist die Fähigkeit des Managements mutige und schnelle Entscheidungen zu treffen, ohne sich in der „Win-Lose“-Politik auf höchster Ebene zu verstricken. *Fluidität der Ressourcen* beschreibt die Fähigkeit, Ressourcen schnell umzuverteilen und Fähigkeiten neu zu konfigurieren [16]

Zusammengefasst ist es die Aufgabe des Managements neue strategische Ausrichtungen zu entdecken, entschlossen anzugehen und die Ressourcen bereitzustellen.

Außerdem stellen Doz und Kosonen [16] fest, dass Geschäftsmodelle dazu neigen, von Natur aus stabil zu sein und sich nur schwer ändern lassen. Unternehmen entwickeln im Laufe der Zeit ganz natürlich Geschäftsmodelle, an denen sie häufig festhalten und fallen der Starrheit ihres Geschäftsmodells zum Opfer [16].

#### **4.2.2 RQ2: Modelle für eine agilen Organisation**

In der wissenschaftlichen Literatur existieren viele Modelle von Organisationen. Ein Vorläufer der ersten skalierbaren Frameworks ist die *Crystal Family* und wurde 1994 entwickelt [52].

Häufig sind es Weiterentwicklungen von bereits existierenden Modellen

oder eine Erweiterung einer agilen Methode. Agile Frameworks wie Scrum sind nicht geeignet große oder dezentrale Teams zu verwalten, daher wurde mit der Entwicklung von *Scaled Agile Frameworks* begonnen. Auch auf die Entwicklung weiterer Frameworks hat Scrum einen großen Einfluss.

Die in der heutigen Zeit geläufigen agilen Frameworks, welche in dieser Arbeit verglichen wurden, sind: SAFe, Spotify, LeSS, DAD, Scrum@Scale und das Nexus Framework. Zunächst ein kleiner Einblick in die jeweiligen Frameworks:

*SAFe:*

Das *Scaled Agile Framework* (SAFe) ist ein skalierbares und konfigurierbares Framework, dessen Kernstück der *Agile Release Train* ist. Der *Agile Release Train* ist der Zusammenschluss mehrere Teams, die inkrementell Komponenten entwickeln, welche wiederum in den *Solution Train* einfließen. Die Portfolioebene ist für die Ausrichtung der Strategie des Unternehmens verantwortlich (vgl. Grundlagen Kapitel 2).

*DAD:*

*Disciplined Agile Delivery* (DAD) ist eine Erweiterung von Scrum, dessen Hauptziel es ist, den gesamten Lebenszyklus der Entwicklung abzudecken und gleichzeitig Praktiken aus anderen agilen Methoden wie Lean und Kanban zu übernehmen. [5]

*LeSS:*

*Large Scale Scrum* (LeSS) unterteilt sich in zwei verschiedenen groß angelegte Scrum-Frameworks. Die kleinere Variante besteht aus bis zu acht Teams und wird meist nur LeSS genannt. Für die bessere Unterscheidung wird diese Variante, nach Vorlage von Alqudah et al. [5] *Basic LeSS* genannt. Die große LeSS Variante kann aus mehreren tausend Mitarbeitern bestehen und wird *LeSS Huge* genannt. [57]

Für die Umsetzung von Projekten gibt es organisatorische Änderungen zu Scrum durch Bildung funktionsübergreifender Teams und durch die Abschaffung traditioneller Rollen (z. B. Projektmanager, Teamleiter). Zusätzlich verwendet LeSS weitere Praktiken und Grundsätze anderer agiler Methoden. [4]

*Nexus Framework:*

Das *Nexus Framework* (Nexus) hat als wesentliches Ziel die Koordination von mehreren Scrum-Teams. Der größte Unterschied zu Scrum ist der Fokus, der auf die Erforschung von Abhängigkeiten und Synchronismen

zwischen den verschiedenen Scrum-Teams gelegt wird. Dafür wurde die Rolle des Nexus-Integrationsteams erschaffen, welches für die Anwendungsintegrationsarchitektur und die Betreuung der Scrum-Teams verantwortlich ist. [4]

*Scrum@Scale:*

Scrum@Scale wurde von Dr. Jeff Sutherland auf der Grundlage von Scrum entwickelt. Die offiziellen Richtlinien von Scrum@Scale [2] wollen die neuartigen Netzwerke von Teams effizienter koordinieren, mit Hilfe der Einrichtung einer „minimal lebensfähigen Bürokratie“ über eine „skalierungsfreie“ Architektur.

Scrum at Scale hat zwei Zyklen: den Scrum Master Zyklus und den Product Owner Zyklus, dessen Kombination einen Rahmen schaffen soll, der die Bemühungen mehrerer Teams auf ein einziges Ziel ausrichtet.[4]

*Spotify:*

Das Spotify Modell wurde nach dem Unternehmen Spotify benannt, in dem es das erste Mal eingesetzt wurde und auch weiter entwickelt wird. Die Entwickler organisieren sich in kleinen Teams, *Squad* genannt, welche sich wiederum zu einem *Tribe* zusammenschließen. Innerhalb eines *Tribes* gibt es *Chapter*, in denen sich die Experten eines bestimmten Fachgebietes gegenseitig austauschen. Zusätzlich können die Entwickler auf freiwilliger Basis einer oder mehrerer Gilden beitreten, um Erfahrungen im gesamten Unternehmen weiter zugeben (vgl. Grundlagen Kapitel 2).

Criteria	DAD	SAFe	LeSS 1	LeSS 2	Spotify	Nexus
Team size	200 people or more. It also supports small and medium teams.	Large Enterprise includes more than 1 release trains (50 to 124 people in each release trains)	Up to 70 people or 10 SCRUM teams, 7 stakeholders in each team	Any large projects, More than thousand people on one product	Any large projects, Normally 250 to 300 people at Spotify (30 teams)	Three to nine SCRUM teams
Methods and practices adopted	Kanban Practices, SCRUM (almost all SCRUM practices), Agile Modeling which is the source for DAD's modeling and documentation practices, the Unified Process, XP, TDD and Agile Data.	SCRUM, Lean, Kanban, SCRUMBan, DevOps and some practices of XP	SCRUM was fully adopted including additional practices for large projects	SCRUM was fully adopted including additional practices for large projects	Allow Kanban, SCRUM, DevOps and Lean Startup	SCRUM with additional practices in solving the dependency-related issues in multiple teams
Organization Type	Multiple Organization and Enterprise practicality	Enterprises and portfolio level	Large Traditional organization	Enterprises	Enterprises specifically similar to Spotify	Portfolio level for medium project

Tabelle 4.2: Unterschiede und Gemeinsamkeiten verschiedener Frameworks (nach Alqudah et al. [5])

In Tabelle 4.2 werden verschiedene Frameworks miteinander verglichen. Das Framework LeSS (*Large-Scale Scrum*) wird in der Studie [5] in *Basic LeSS* (LeSS 1) und in *LeSS huge* (LeSS) aufgeteilt. Die „kleinsten“ in der Tabelle gezeigten Frameworks sind Nexus und LeSS 1, an deren Projekten maximal 70 bis 80 Entwickler mitarbeiten können. Die Frameworks LeSS 2 und Spotify sind ausschließlich für große Projekte gedacht, während DAD und SAFe sowohl kleine als auch große Projekte umsetzen können. In der Literatur gibt es zu „large“ bzw. „large scale“ verschiedene Definitionen, die sich auf die Anzahl der Teams, die Teamgröße, die Komplexität, die Codezeilen, die Zeitdauer und die Kosten des Projekts beziehen können [5]. Die Teamgröße ist bei den Frameworks meistens sehr ähnlich, da alle

Frameworks mit der Verwendung von Scrum in den Teams kompatibel sind. Die Tabelle 4.2 gibt bei der Teamgröße zuerst die Größe des Projektes/Unternehmens und anschließend die Anzahl der Teams oder direkt die Teamgröße an. Dabei ist zu beachten, dass ein Standard Scrumteam aus fünf bis neun Entwicklern besteht.

Bei den angewendeten Methoden und Praktiken fällt auf, dass in allen Frameworks Scrum üblich ist, aber nicht immer notwendigerweise verwendet werden muss [5][19]. Die Frameworks LeSS und Nexus sind Weiterentwicklungen von Scrum, wodurch auf Scrum nicht verzichtet werden kann [5].

### 4.2.3 RQ3: Einfluss der Modelle auf agile Organisation

Die dritte Forschungsfrage soll zeigen wie agil die Frameworks wirklich sind, welche Anpassungen von den Unternehmen an den Frameworks erstellt wurden und welche Kritik es an den Modellen in der wissenschaftlichen Literatur gibt.

Das Spotify Modell dient als weit verbreitete Vorlage für agile Organisationen. In einer *Multiple Case Study* haben Gerster et al. [21] festgestellt, dass die Unternehmen mit Spotify-Modell, vom Original Modell wie es das Unternehmen Spotify vorlebt, abweichen. Aus den Veränderungen wurden in der Studie vier Modelle erstellt. Das erste Modell (*Spotify-Template/Fully Agile Unit*) ist das Original Spotify Modell. In dem Original Modell wurden zwei Nachteile erkannt. Erstens, es wird immer angenommen, dass die Squads über alle erforderlichen Ressourcen für die Produktlieferung verfügen, sind in der Regel jedoch mit begrenzten Ressourcen konfrontiert und haben oft nicht alle erforderlichen Ressourcen zur Verfügung. Squads sind für bestimmte Anforderungen auf Spezialisten angewiesen, die nicht in allen Teams vorhanden sind. Der zweite Nachteil ist, dass Squads völlig autonom produktbezogene Entscheidungen treffen können, welches zu Mangel an Standardisierung und Synergien zwischen den Produkten führt.

Um diesen Nachteilen entgegen zu wirken wurde im zweiten Modell (*Fully Agile Unit with Cross-product Support*) ein produktübergreifendes Team zu einem *Tribe* hinzugefügt. Diese Einheit stellt sicher, dass grundlegende Architekturstandards von allen Squads befolgt werden. Um das begrenzte Ressourcenproblem zu beheben, wurde ein *Shared-Services-Team* zum *Tribe* hinzugefügt, das allen Squads des Stammes zur Verfügung steht und spezielles Fachwissen liefert.

In der *Case Study* viel auf, dass das funktionsübergreifende *Shared-Services-Team* die Entwicklung verlangsamt, da es zu ständiger Feinabstimmung über die Dienstleistungen zwischen dem *Shared-Services-Team* und dem Squad kommt. Das dritten Modell (*Fully Agile Unit with Cross-product Projects*) spezifiziert das zweite Modell weiter und berücksichtigt die Existenz von

Projekten in agilen Organisationsformen. Dies hat zur Folge, dass es Ressourcenpriorisierungskonflikte zwischen Produkten und Projekten gibt. Die Unternehmen CarCo-IT und ChemCO haben das Problem gelöst, indem sie den Projekten spezielle Squad-Kapazitäten zuwies und Projekte wie Backlog-Elemente behandelten. Häufige Änderungen des Bedarfs an Projektressourcen führen sowohl zu einer Verlangsamung der Projekte als auch der Squads. Das vierte Modell (*Fully Agile Unit in a Multiple-partner Setting*) ist eine spezielle Erweiterung des zweiten Modells für mehrere Organisationen, die gemeinsam an einem Projekt beteiligt sind, dessen Nachteil die Aufteilung der Verantwortung ist.

Kritik an dem SAFE Framework gibt es von verschiedenen Autoren. Salikhov et al. [48] meinen, dass SAFe möglicherweise zu kompliziert und schwer zu verstehen sei und ein Rahmenwerk mit einer organisatorischen Top-Down-Kontrolle geschaffen werde, die zu einer strengen Organisationsstruktur mit geringerer Flexibilität führt. Bei einer Umfrage von Entwickler aus Organisationen, die SAFe verwenden, haben mehrere Teilnehmer angegeben, dass bei einer falschen Implementierung SAFe nicht agil wird. [48] Weitere Kritik, die sich nicht direkt auf das Framework bezieht, ist dass es in der Literatur zu wenig Fallstudien zu SAFe existieren [40][45]. Putta et al. [45](2018) bemängelten, dass nur fünf von Fachleuten geprüfte Fallstudien zu SAFe gefunden wurden und somit eine systematischen Beschreibungen der Transformation und Implementierungen fehlen. Außerdem seien viele nicht von Fachleuten begutachteten Fallstudien von SAFe-Berater veröffentlicht worden, welche vermutlich eine „inhärente Verzerrung bezüglich der Herausforderungen und negativen Eigenschaften von SAFe in großen Unternehmen aufweisen“ ([45] S.4)

## Kapitel 5

# Modell einer agilen Organisation

In der Analyse der Literatursuche wurden zunächst die Eigenschaften und Voraussetzungen für agile Organisationen untersucht, anschließend wurden die Frameworks, welche in der Praxis bereits verwendet werden mit einander verglichen. Im letzten Teil der Analyse ist auf die Kritik an verschiedenen Modellen eingegangen worden und auch welche Änderungen vorgenommen wurden, um das Modell zu verbessern. Aus den Ergebnissen dieser Analyse wird in diesem Kapitel ein grundlegendes Modell einer agilen Organisation erstellt, welches die wichtigsten Eigenschaften einer agilen Organisation in sich vereint.

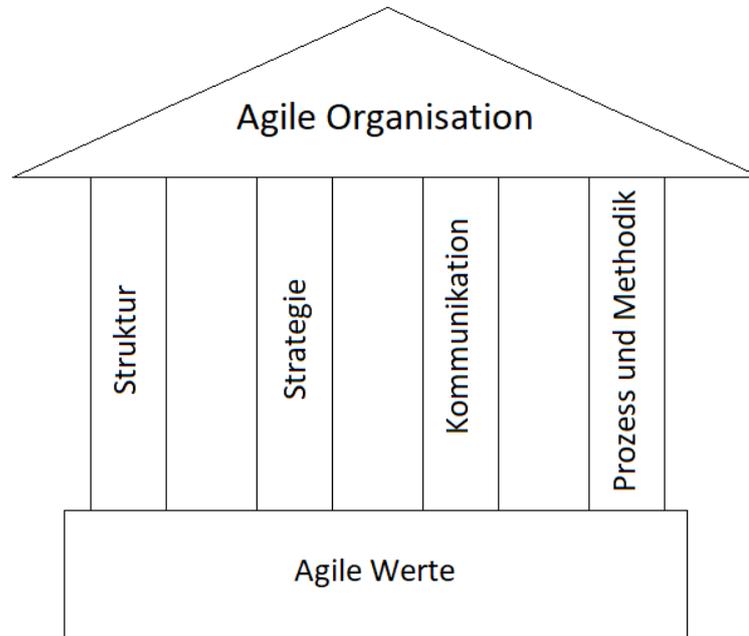


Abbildung 5.1: Grundlegendes Modell einer agilen Organisation

Die Grundebene des Modells bilden die *Agilen Werte*, diese Werte sind in allen Teilen der Organisation zu finden, ohne sie bricht die agile Organisation zusammen. Darauf stehen vier Säulen, welche die agile Organisation tragen:

- **Struktur:** Über welche grundlegenden Aufbaustrukturen die Organisation verfügen muss
- **Strategie:** Die langfristige Unternehmensstrategie
- **Kommunikation:** Wie die Mitarbeiter untereinander kommunizieren
- **Prozess und Methodik:** Welche Prozesse und Methoden notwendig sind

## 5.1 Agilen Werte

Die Agilen Werte bilden das Grundgerüst der Organisation. Ohne die agilen Werte ist der Aufbau einer agilen Organisation nicht möglich. Auf dem Grundgerüst stehen die vier Säulen, ist eine nicht voll einsatzfähig, schadet es der Organisation, zerstört sie jedoch nicht wenn der Schaden behoben wird. Probleme könnten z.B. sein, wenn nicht ausreichend oder zu langsam

kommuniziert wird.

Zu den agilen Werten gehören die Eigenschaften einer agilen Organisation, wie sie in der Analyse der ersten Forschungsfrage herausgearbeitet wurden. Dazu gehört, dass alle Mitarbeiter ein *agiles Mindset* besitzen und die Vision des Unternehmens, sowie die Abläufe und Konzepte des Frameworks verstehen.

Ein weitere Eigenschaft ist die *Selbstorganisation* der Entwickler. Die Entwickler bzw. Teams müssen eigene Entscheidungen treffen z.B. welche agilen Methoden sie in ihrem Team verwenden wollen. Damit die Teams sich selbst organisieren können, muss die Managementebene in Form der *katalytischen Führung* auf ihrer Weisungsbefugnis und Ausführungskontrolle über die Teams verzichten. Teams die nicht auf Entscheidungen vom Management warten müssen und selbst ständig entscheiden können sind agiler, da sie schnell und flexibel auf Änderungen reagieren können.

## 5.2 Struktur

Der strukturelle Aufbau von agilen Organisationen ist bei den verschiedenen Frameworks meistens sehr ähnlich. Es existieren drei hierarchische Ebenen innerhalb der Organisationen:

- Teamebene
- Projektebene
- Managmentebene

### *Teamebene*

Die kleinste agile Einheit ist das Team. Die Teams in den Frameworks haben verschiedene Bezeichnungen z.B. Squads (Spotify-Modell) oder *Agile Teams* (SAFe), ihre Aufgaben sind jedoch die gleichen (vgl. Kapitel 2). Die Größe der Teams ist üblicherweise bis zu zehn Personen (vgl. RQ2 Kapitel 4). Diese Größe eignet sich gut für das Anwenden von Scrum, welches wie in zweiten Forschungsfrage gezeigt, ein beliebtes Verfahren in verschieden Frameworks auf der Teamebene ist. Die Teams arbeiten an kleinen Aufgaben, welche zusammen mit weiteren Teams auf der Projektebene untereinander aufgeteilt werden. Die Anzahl der Teams variiert in Abhängigkeit von dem Umfang und der Komplexität des Projektes.

### *Projektebene*

Die Projektebene besteht aus dem Zusammenschluss der Teams, es sind üblicherweise zwischen zwei Teams und maximal neun Teams, die sich zusammenschließen. In dem Safe Framework entspricht dieser Zusammenschluss einem *Agile Release Train* und im Spotify Modell einem *Tribe* (vgl. Grundlagen Kapitel 2). Auf der Projektebene steht die Organisation der Teams im Vordergrund, dazu gehört die Planung des nächsten Inkrements und die Aufteilung der Aufgaben auf die verschiedenen Teams. Zusätzlich gibt es in dieser Ebene neue Rollen. Meistens gibt es eine Art *Scrum Master*, welcher die Treffen für die Planung leiten und einen *Product Owner*, welcher für das *Backlog*, also die zu verteilenden Aufgaben zuständig ist.

### *Managementebene*

Die Managementebene hat in einer agilen Organisation einen Teil seiner „Macht“ an die anderen Ebene abgeben, indem es ihnen mehr Selbständigkeit erlaubt. Dennoch ist diese Ebene wichtig für das Unternehmen. Die Führungskräfte in der Managementebene sind für die Entwicklung einer Unternehmensstrategie verantwortlich und für die Weitergabe dieser Vision an die unteren Ebenen.

Neben diesen drei Ebenen kann es noch weitere Strukturen in einer agilen Organisation geben. In dem Spotify Modell gibt es noch das *Chapter* und die *Guilds*, beide dienen dazu, dass das Fachwissen sich in der Organisation verbreitet. Das *Chapter* umfasst nur die Experten eines Fachgebietes eines Projektes, während in einer Gilde alle Entwickler teilnehmen können und das Wissen in der gesamten Organisation sich verbreiten kann.

## 5.3 Strategie

Die „richtige“ Strategie, wenn es denn eine solche gibt, ist immer unternehmensabhängig, somit gibt es keine absolut sichere Unternehmensstrategie mit der jedes Unternehmen erfolgreich ist.

In Kapitel 4 zur ersten Forschungsfrage wurde gezeigt, dass es in der wissenschaftlichen Literatur Uneinigkeiten existieren, ob Unternehmen agiler sind, wenn sie auf langfristige Gewinne ausgelegt sind oder zusätzlich eine kurzfristige Chance nach der anderen zu verfolgen. Kurzfristige Chance zu verfolgen würde eindeutig der agilen Mentalität, flexibel auf Änderungen zu reagieren, entsprechen.

Die Unternehmensstrategie zu definieren und anzupassen an Umweltfaktoren ist die Aufgabe der Manager der Organisation. Ein aktuelles Beispiel für

den Einfluss von Umweltfaktoren ist die Corona Pandemie. Die Entwickler arbeiten grundsätzlich zusammen an einem Ort, die Pandemie kann dazu führen, dass die Mitarbeiter im *Home Office* arbeiten müssen. Die Manager müssen mit Absprache der Entwickler eine Lösung finden, wie die Kommunikation und Zusammenarbeit weiterhin gewährleistet werden kann und die Softwareentwicklung möglichst wenig beeinträchtigt wird.

## 5.4 Kommunikation

Kommunikation ist ein wichtiger Bestandteil in der agilen Organisation, um sich untereinander Abzusprechen und flexibel agieren zu können. Kommunikation findet innerhalb der Ebenen, zwischen den Ebenen und nach außen zur Umwelt, wie zum Beispiel dem Kunden statt.

Durch die Selbständigkeit der Teams und dessen Entwickler können leicht Entscheidungen gefällt werden, die sich auch auf andere Personen bzw. Teams auswirken. Um Chaos zu vermeiden und damit die Entwicklung des Projektes zu verlangsamen, müssen die Entwickler viel kommunizieren. Je größer das Team, das Projekt oder die Organisation, desto komplexer wird auch die Kommunikation.

Um die Kommunikation innerhalb des Teams zu erhöhen und zu fördern arbeitet das Team lokal am gleichen Ort und es gibt tägliche Treffen für den internen Austausch z.B in Form eines *Daily Scrum*. Für die Team-zu-Team Kommunikation ist gerade die Planungsphase zu Beginn eines neuen Increments gedacht. Natürlich ist der Austausch zwischen den Teams innerhalb des Increments möglich und auch erwünscht. Für die Kommunikation mit dem Kunden gibt es regelmäßige Treffen, in denen der Kunde den derzeitigen Stand der Entwicklung sieht und auch mitentscheiden kann, welche Aufgaben als nächstes umgesetzt werden sollen. Der Kunde kann auch mit Absprache der Entwickler Änderungen an dem Projekt vornehmen lassen.

## 5.5 Prozess und Methodik

In jeder der betrachteten Modelle für agile Organisationen ist die Anwendung von Scrum ein beliebtes Mittel auf der Teamebene. In vielen Frameworks ist es sogar verpflichtend, da sie auf Scrum aufgebaut sind. Zu beachten ist, dass Scrum verändert werden kann, um das jeweilige Framework zu ermöglichen. So können neue Rollen und Artefakte hinzugefügt werden. Die Analyse der Literatursuche hat ergeben, dass neben Scrum auch weitere Prozesse und Methoden auf der Teamebene beliebt sind. Die Methode *Kanban* kann den Scrum-Prozess optimieren, während die DevOps den gesamten Entwicklungsprozess verbessern [18].

In agilen Organisationen wird in Zyklen gearbeitet. In der Teamebene gibt es ca. zweiwöchige Sprints, wie sie auch in Scrum vorkommen, in denen die Teams ihrer *Stories* umsetzen, die sie vorher untereinander aufgeteilt haben. Die Zyklen der Teams sind aufeinander abgestimmt, so dass sie gleichzeitig anfangen und auch beendet werden. Mehrere Teams werden auf der Projektebene zu einem *Team-of-Team* Konstrukt zusammengefasst, welches in einem eigenen Zyklus arbeitet. Diese Zyklen bieten den Vorteil, dass der *Product Owner* am Anfang jedes Zyklus mitentscheiden kann, welche Aufgaben als nächstes bearbeitet werden und am Ende des Zyklus werden die Ergebnisse präsentiert und es ist möglich direkt Feedback zugeben. Auf diese Weise kann flexibel auf Kundenwünsche reagiert werden und die Agilität des Unternehmens gesteigert werden.

## Kapitel 6

# Verwandte Arbeiten

Während in dieser Arbeit „Software“-Organisationen im Vordergrund standen, gibt es agile Organisation auch in anderen Branchen, wie z.B. in der Fertigungsindustrie. Die Gründe für eine Transformation zu einer agilen Organisation sind in der Fertigungsindustrie sehr ähnlich, zu denen aus dem Softwarebereich. Der Schwerpunkt liegt bei der Kostenreduzierung und einer höhere Flexibilität für den Kunden [23]. Außerdem stehen der Prozesse mit Kunden und Lieferanten stärker im Mittelpunkt als bei der Entwicklung von Software [17].

Teil der agilen Strategie ist die Bildung von „virtuellen Organisationen“. Virtuelle Organisationen sind ein vorübergehender Zusammenschluss verschiedener unabhängiger Organisationen, die anderen Organisationen gegenüber wie ein einheitliches Unternehmen auftreten [6][14]. In der Fertigungsindustrie werden virtuelle Organisation auch *Virtual Manufacturing* genannt. Gunasekaran definiert *Virtual Manufacturing* als „eine integrierte, synthetische Fertigungsumgebung, die zur Verbesserung aller Entscheidungs- und Steuerungsebenen in einem Fertigungsunternehmen eingesetzt wird“ [23]. Damit unterscheiden sich Fertigungsorganisationen stark zu den „Software“-Organisationen, in den nur die Teams eines Unternehmens gemeinsam Entscheidungen treffen müssen. Beispiele für virtuelle Organisation gibt es in der Schwerindustrie, in Schiffbauunternehmen oder im Baugewerbe [23].



# Kapitel 7

## Zusammenfassung und Ausblick

### 7.1 Zusammenfassung

In dieser Arbeit wurde mit einer *Systematic Mapping Study*, der Forschungsbereich der agilen Organisation untersucht. Die agilen Frameworks, die in der Literatur vorkommen, sind miteinander verglichen worden und die grundlegenden Eigenschaften einer agilen Organisation wurden gesammelt und bewertet. Aus der Literatur ging auch hervor, dass über das Framework SAFe besonders häufig wissenschaftliche Publikationen verfasst werden. Es wurde festgestellt, dass es noch Lücken in der wissenschaftlichen Literatur zu den Problemen von agilen Modellen existieren und dass es zu wenig Fallstudien gibt, die von unabhängigen Experten geprüft werden. Es wurde anhand von einem Beispiel aus einer Fallstudie gezeigt, wie Unternehmen die Modelle anpassen, um mögliche Probleme zu beheben.

Mit den Ergebnissen der Analyse wurde ein grundlegendes Modell einer agilen Organisation aufgebaut, welches die wichtigsten Eigenschaften und Strukturen anderer agiler Modelle vereinigt.

### 7.2 Ausblick

Die Forschung zu agiler Organisation ist noch am Anfang, das Interesse in der Wissenschaft stieg in den letzten Jahren stark an, wie man gut anhand der Anzahl von veröffentlichten Publikationen zum Thema agile Organisation in den letzten Jahren sehen kann. In einigen Publikationen wird bemängelt, dass es zu wenig von Experten geprüfte Fallstudien gibt (vgl. Kapitel 4, RQ3). Mit der ansteigenden Zahl von Veröffentlichungen wird auch die Anzahl der Fallstudien weiter steigen und diese Forschungslücke schließen, dies könnte dazu führen, dass noch mehr Unternehmen die Transformation zu einer agilen Organisation wagen.

Eine weitere Forschungslücke ist, ob hybride Organisationen, die sowohl agile Anteile z.B. in der Entwicklung haben, wie auch auch traditionelle Strukturen besitzen möglicherweise effizienter sind als agile Organisationen. Auch gibt es bereits die Idee von *Agile Government*, eine Art agile Regierungen/Behörden, die ebenfalls von der Flexibilität, die Agilität ihnen verspricht profitieren wollen. [34].

# Abbildungsverzeichnis

2.1	Der Systematic Mapping Prozess [41] . . . . .	6
2.2	Aufbau des SAFe Modells [1] . . . . .	8
2.3	Aufbau des Spotify Modells . . . . .	10
3.1	Ablauf der gesamten Literatursuche . . . . .	17
4.1	Anteile agiler Frameworks in der Literatur . . . . .	21
4.2	Übersicht über die Veröffentlichungsjahre der Literatur . . . . .	22
5.1	Grundlegendes Modell einer agilen Organisation . . . . .	32



# Tabellenverzeichnis

3.1	Aufbau des Startset der Literatursuche . . . . .	15
3.2	Übersicht der weiteren Iteration der systematischen Literatursuche . . . . .	16
4.1	Übersicht von Unternehmen und dem verwendeten Framework	20
4.2	Unterschiede und Gemeinsamkeiten verschiedener Frameworks (nach Alqudah et al. [5]) . . . . .	28
A.1	Verwendete Literatur aus der systematischen Literatursuche .	48



## Anhang A

# Ausgewählten Publikationen der systematischen Literatursuche

<b>Titel</b>	<b>Autoren</b>	<b>Jahr</b>	<b>Datenbank</b>
KWM: Knowledge-based Workflow Model for Agile Organization [31]	Ha Bin Lee, Jong Woo Kim, Sung Joo Park	1999	Google Scholar
The journey to an agile organization [8]	Daniel Brosseau, Sherina Ebrahim, Christopher Handscomb, and Shail Thaker	2019	Google Scholar
An AHP Model towards an Agile Enterprise [33]	Mohamed Amine Marhraoui, Abdellah El Manouar	2017	Google Scholar
Unpacking Agile Enterprise Architecture Innovation work practices: A Qualitative Case Study of a Railroad Company [50]	B Veeresh Thummadi, Vishal Khapre, Rosalie Ocker	2017	Google Scholar
Agile methodologies and the emergence of the agile organization: A software development approach waiting for its time? [55]	Glen van der Vyver, Andy Koronios, Michael Lane	2003	Google Scholar
Everyone's Going to be an Architect [27]	Bettina Horlach, Andreas Drechsler, Ingrid Schirmer, Paul Drews	2020	Google Scholar
Modeling an Agile Enterprise: Reconciling Systems and Process Thinking [7]	Ilia Bider, Gene Bellinger, Erik Perjons	2011	Google Scholar
Fostering and sustaining innovation in a fast growing agile company [35]	Nils B. Moe, Sebastian Barney, Claes Wohlin, Hamish T. Barney, Tony Gorschek, Martha Winata	2012	Google Scholar

<b>Titel</b>	<b>Autoren</b>	<b>Jahr</b>	<b>Datenbank</b>
The SAFe way to the Agile Organization [42]	Jan Pries-Heje, Malene M. Krohn	2017	Google Scholar
Agile Enterprise Architecture Management: An Analysis on the Application of Agile Principles [25]	Matheus Hauder, Sascha Roth, Christopher Schulz and Florian Matthes	2014	Google Scholar
Spotify Guilds: How to Succeed With Knowledge Sharing in Large-Scale Agile Organizations [49]	Darja Šmite, Nils Brede Moe, Georgiana Levinta and Marcin Floryan	2019	Google Scholar
Agile Organization as a concept of production adjustment in the face of the crisis [51]	Rafał Tyszkiewicz, Agnieszka Pawlak-Wolanin	2017	Google Scholar
Software Development Artifacts in Large Agile Organizations: A Comparison of Scaling Agile Methods [57]	Ewelina Winska, Włodzisław Dąbrowski	2020	Google Scholar
Designing An Agile Enterprise Architecture For Mining Company By Using TOGAF Framework [60]	Yuliana, Rika and Rahardjo, Budi	2016	Google Scholar
How To Make The Whole Organization Agile [15]	Denning, Stephen	2015	Google Scholar
ISRUP E-Service Framework for agile Enterprise Architecting [24]	Hashemi, S. M.; Razzazi, M.; Bahrami, A.	2006	Google Scholar
Policies for Self-Managing Communities in Agile Organizations [20]	Kevin C. Feeney, David Lewis, Vincent Wade	2007	IEEE
A Pragmatic Approach for Message Modeling - A Case study in an Agile Organization [37]	Mina Boström Nakićenović	2016	ACM
Adopting Scaled Agile Framework (SAFe): A Multivocal Literature Review [45]	Abheeshta Putta, Maria Paasivaara, Casper Lassenius	2018	ACM
Scaling Agile across the Global Organization: An Early Stage Industrial SAFe Self-Assessment [46]	Mohammad Abdur Razzak, Ita Richardson, John Noll, Clodagh Nic Cana, Sarah Beecham	2018	ACM
How Enterprises Adopt Agile Forms of Organizational Design: A Multiple-Case Study [21]	Daniel Gerster, Christian Dremel, Walter Brenner, Prashant Kelker	2020	ACM
Large-Scale Agile Development Patterns for Enterprise and Solution Architects [53]	Ömer Uludağ, Florian Matthes	2020	Science Direct
Embedding Strategic Agility: A leadership agenda for accelerating business model renewal [16]	Yves L. Doz and Mikko Kosonen	2010	Science Direct
Enterprise architecture for high flexible and agile company in automotive industry [22]	Duarte Gonçalves, Luís Ferreira, Nuno Campos	2021	Springer Link

<b>Titel</b>	<b>Autoren</b>	<b>Jahr</b>	<b>Datenbank</b>
Future software organizations – agile goals and roles [29]	Petri Kettunen, Maarit Laanti	2017	Springer Link
Large-scale agile transformation at Ericsson: a case study [40]	Maria Paasivaara, Benjamin Behm, Casper Lassenius, Minna Hallikainen	2018	Google Scholar
A Review of Scaling Agile Methods in Large Software Development [5]	Alqudah, Mashal; Razali, Rozilawati	2016	Google Scholar
Towards an Agile Design of the Enterprise Architecture Management Function [10]	Sabine Buckl, Florian Matthes, Ivan Monahov, Sascha Roth, Christopher Schulz, Christian M. Schweda	2011	Google Scholar
Critical Issues in Enterprise Architecting – A Literature Review [32]	Carsten Lucke, Sascha Krell, Ulrike Lechner	2010	Google Scholar
Scaling Agile [19]	Christof Ebert and Maria Paasivaara	2017	Google Scholar
Implementing Large-Scale Agile Frameworks: Challenges and Recommendations [13]	Kieran Conboy and Noel Carroll	2019	Google Scholar
IT Governance in Scaling Agile Frameworks [26]	Horlach, Bettina; Böhmann, Tilo; Schirmer, Ingrid; Drews, Paul	2018	Google Scholar
Characteristics and Principles of Scaled Agile [30]	Laanti Maarit	2014	Google Scholar
Was heisst hier eigentlich „agil“? kennzeichen agiler Organisationen [12]	Jens Coldewey	2012	Google Scholar
Establishing Architecture Guidelines in Large-Scale Agile Development Through Institutional Pressures [54]	Ömer Uludag, Matheus Hauder	2019	Google Scholar
What to Expect from Enterprise Architects in Large-Scale Agile Development? A Multiple-Case Study [?]	Ömer Uludag, Martin Kleehaus	2019	Google Scholar
Adopting SAFe to Scale Agile in a Globally Distributed Organization [45]	Maria Paasivaara	2017	Google Scholar
Investigating the Role of Architects in Scaling Agile Frameworks [52]	Omer Uludag, Martin Kleehaus, Xian Xu, Florian Matthes	2017	Google Scholar
An Empirical Analysis of Success Factors in the Adoption of the Scaled Agile Framework – First Outcomes from an Empirical Study [48]	Dilshat Salikhov, Giancarlo Succi, Alexander Tormasov	2020	Google Scholar

## 48 ANHANG A. AUSGEWÄHLTEN PUBLIKATIONEN DER SYSTEMATISCHEN LITERATUR

<b>Titel</b>	<b>Autoren</b>	<b>Jahr</b>	<b>Datenbank</b>
Assessing Requirements for Agile Enterprise Architecture Management: A Multiple-Case Study [11]	Philip Cammin, Leonard Heilig, Stefan Voß	2021	Google Scholar
Designing Enterprise Architecture in Koperasi Karyawan using TOGAF Architecture Development [36]	Noora Qotrun Nada et al	2020	Google Scholar
TOGAF for Agile SOA Modelling [38]	Feng Ni, Runye Li	2017	Google Scholar
Scaling Agile Software Development to Large and Globally Distributed Large-scale Organizations [44]	Putta, Abheeshta	2018	Google Scholar

Tabelle A.1: Verwendete Literatur aus der systematischen Literatursuche

---

# Literaturverzeichnis

- [1] Scaled Agile Framework (SAFe) Tutorial | SAFe Agile Framework Tutorial | Introduction to SAFe Agile. <https://www.youtube.com/watch?v=prlze0QhHmg>, Eingesehen September 2021.
- [2] The Official Scrum@Scale Guide. <https://www.scrumatscale.com/scrum-at-scale-guide/>, Eingesehen Oktober 2021.
- [3] Wertströme – Value Streams – Value Stream Mapping. <https://www.gotscharek-company.com/blog-1/139-value-streams-und-die-bedeutung-im-scaled-agile/-framework-safe-5>, Eingesehen Oktober 2021.
- [4] F. Almeida and E. Espinheira. Large-scale agile frameworks: A comparative review. *Journal of Applied Sciences, Management and Engineering Technology*, 2(1):16–29, 2021.
- [5] M. Alqudah and R. Razali. A review of scaling agile methods in large software development. *International Journal on Advanced Science, Engineering and Information Technology*, 6(6):828–837, 2016.
- [6] R. Alt, C. Legner, and H. Österle. Virtuelle organisation-konzept, realität und umsetzung. *HMD-Praxis der Wirtschaftsinformatik*, 42(242):7–20, 2005.
- [7] I. Bider, G. Bellinger, and E. Perjons. Modeling an agile enterprise: reconciling systems and process thinking. In *IFIP Working Conference on The Practice of Enterprise Modeling*, pages 238–252. Springer, 2011.
- [8] D. Brosseau, S. Ebrahim, C. Handscomb, and S. Thaker. The journey to an agile organization. *McKinsey & Company, May*, 10, 2019.
- [9] F. Brückner and F. von Ameln. Agilität. *Gruppe. Interaktion. Organisation. Zeitschrift für Angewandte Organisationspsychologie (GIO)*, 47(4):383–386, 2016.
- [10] S. Buckl, F. Matthes, I. Monahov, S. Roth, C. Schulz, and C. M. Schweda. Towards an agile design of the enterprise architecture

- management function. In *2011 IEEE 15th International Enterprise Distributed Object Computing Conference Workshops*, pages 322–329. IEEE, 2011.
- [11] P. Cammin, L. Heilig, and S. Voß. Assessing requirements for agile enterprise architecture management: A multiple-case study. In *Proceedings of the 54th Hawaii International Conference on System Sciences*, page 6007, 2021.
- [12] J. Coldewey. Was heisst hier eigentlich, „agil“? kennzeichen agiler organisationen. *Objekt Spektrum*, (5):14, 2012.
- [13] K. Conboy and N. Carroll. Implementing large-scale agile frameworks: challenges and recommendations. *IEEE Software*, 36(2):44–50, 2019.
- [14] J. A. Crispim and J. P. de Sousa. Partner selection in virtual enterprises. *International Journal of Production Research*, 48(3):683–707, 2010.
- [15] S. Denning. How to make the whole organization “agile”. *Strategy & Leadership*, 2016.
- [16] Y. L. Doz and M. Kosonen. Embedding strategic agility: A leadership agenda for accelerating business model renewal. *Long range planning*, 43(2-3):370–382, 2010.
- [17] R. Dubey and A. Gunasekaran. Agile manufacturing: framework and its empirical validation. *The International Journal of Advanced Manufacturing Technology*, 76(9-12):2147–2157, 2015.
- [18] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano. Devops. *Ieee Software*, 33(3):94–100, 2016.
- [19] C. Ebert and M. Paasivaara. Scaling agile. *Ieee Software*, 34(6):98–103, 2017.
- [20] K. C. Feeney, D. Lewis, and V. Wade. Policies for self-managing communities in agile organizations. In *2007 2nd IEEE/IFIP International Workshop on Business-Driven IT Management*, pages 104–105. IEEE, 2007.
- [21] D. Gerster, C. Dremel, W. Brenner, and P. Kelker. How enterprises adopt agile forms of organizational design: a multiple-case study. *ACM SIGMIS Database: the DATABASE for Advances in Information Systems*, 51(1):84–103, 2020.
- [22] D. Gonçalves, L. Ferreira, and N. Campos. Enterprise architecture for high flexible and agile company in automotive industry. *Procedia Computer Science*, 181:1077–1082, 2021.

- [23] A. Gunasekaran. Agile manufacturing: a framework for research and development. *International journal of production economics*, 62(1-2):87–105, 1999.
- [24] S. M. Hashemi, M. Razzazi, and A. Bahrami. Isrup e-service framework for agile enterprise architecting. In *Third International Conference on Information Technology: New Generations (ITNG'06)*, pages 701–706. IEEE, 2006.
- [25] M. Hauder, S. Roth, C. Schulz, and F. Matthes. Agile enterprise architecture management: an analysis on the application of agile principles. In *4th International Symposium on Business Modeling and Software Design*, pages 38–46, 2014.
- [26] B. Horlach, T. Böhmman, I. Schirmer, and P. Drews. It governance in scaling agile frameworks. *Proceedings of the Multikonferenz Wirtschaftsinformatik, Lüneburg*, pages 1789–1800, 2018.
- [27] B. Horlach, A. Drechsler, I. Schirmer, and P. Drews. Everyone’s going to be an architect: Design principles for architectural thinking in agile organizations. In *HICSS*, pages 1–10, 2020.
- [28] S. Keele et al. Guidelines for performing systematic literature reviews in software engineering. Technical report, Citeseer, 2007.
- [29] P. Kettunen and M. Laanti. Future software organizations—agile goals and roles. *European Journal of Futures Research*, 5(1):1–15, 2017.
- [30] M. Laanti. Characteristics and principles of scaled agile. In *International Conference on Agile Software Development*, pages 9–20. Springer, 2014.
- [31] H. B. Lee, J. W. Kim, and S. J. Park. Kwm: Knowledge-based workflow model for agile organization. *Journal of Intelligent Information Systems*, 13(3):261–278, 1999.
- [32] C. Lucke, S. Krell, and U. Lechner. Critical issues in enterprise architecting—a literature review. 2010.
- [33] M. A. Marhraoui and A. El Manouar. An ahp model towards an agile enterprise. *vol*, 8:151–156, 2017.
- [34] I. Mergel, Y. Gong, and J. Bertot. Agile government: Systematic literature review and future research. 2018.
- [35] N. B. Moe, S. Barney, A. Aurum, M. Khurum, C. Wohlin, H. T. Barney, T. Gorschek, and M. Winata. Fostering and sustaining innovation in a fast growing agile company. In *International Conference on Product Focused Software Process Improvement*, pages 160–174. Springer, 2012.

- [36] N. Q. Nada, S. Wibowo, and M. Novita. Designing enterprise architecture in koperasi karyawan using togaf architecture development. In *IOP Conference Series: Materials Science and Engineering*, volume 835, page 012049. IOP Publishing, 2020.
- [37] M. B. Nakićenović. A pragmatic approach for message modeling—a case study in an agile organization. In *2010 6th Central and Eastern European Software Engineering Conference (CEE-SECR)*, pages 12–18. IEEE, 2010.
- [38] F. Ni and R. Li. Togaf for agile soa modelling. In *CloudCom 2018*, 2018.
- [39] M. Paasivaara. Adopting safe to scale agile in a globally distributed organization. In *2017 IEEE 12th International Conference on Global Software Engineering (ICGSE)*, pages 36–40. IEEE, 2017.
- [40] M. Paasivaara, B. Behm, C. Lassenius, and M. Hallikainen. Large-scale agile transformation at ericsson: a case study. *Empirical Software Engineering*, 23(5):2550–2596, 2018.
- [41] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson. Systematic mapping studies in software engineering. In *12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12*, pages 1–10, 2008.
- [42] J. Pries-Heje and M. M. Krohn. The safe way to the agile organization. In *Proceedings of the XP2017 scientific workshops*, pages 1–3, 2017.
- [43] H. G. Prodoehl. Das agile unternehmen. In *Überlebenselixier Agilität*, pages 11–59. Springer, 2019.
- [44] A. Putta. Scaling agile software development to large and globally distributed large-scale organizations. In *Proceedings of the 13th International Conference on Global Software Engineering*, pages 141–144, 2018.
- [45] A. Putta, M. Paasivaara, and C. Lassenius. Adopting scaled agile framework (safe) a multivocal literature review. In *Proceedings of the 19th International Conference on Agile Software Development: Companion*, pages 1–4, 2018.
- [46] M. A. Razzak, I. Richardson, J. Noll, C. N. Canna, and S. Beecham. Scaling agile across the global organization: an early stage industrial safe self-assessment. In *2018 IEEE/ACM 13th International Conference on Global Software Engineering (ICGSE)*, pages 116–125. IEEE, 2018.
- [47] A. Rzepka and E. Bojar. Leadership as one of the factors shaping the development of an agile organization. *Review of Integrative Business and Economics Research*, 9:383–393, 2020.

- [48] D. Salikhov, G. Succi, and A. Tormasov. An empirical analysis of success factors in the adaption of the scaled agile framework—first outcomes from an empirical study. *arXiv preprint arXiv:2012.11144*, 2020.
- [49] D. Smite, N. B. Moe, G. Levinta, and M. Floryan. Spotify guilds: how to succeed with knowledge sharing in large-scale agile organizations. *IEEE Software*, 36(2):51–57, 2019.
- [50] B. Thummadi, V. D. Khapre, and R. Ocker. Unpacking agile enterprise architecture innovation work practices: A qualitative case study of a railroad company. 2017.
- [51] R. Tyszkiewicz and A. Pawlak-Wolanin. Agile organization as a concept of production adjustment in the face of the crisis. *Production Engineering Archives*, 15, 2017.
- [52] Ö. Uludağ, M. Kleehaus, X. Xu, and F. Matthes. Investigating the role of architects in scaling agile frameworks. In *2017 IEEE 21st International Enterprise Distributed Object Computing Conference (EDOC)*, pages 123–132. IEEE, 2017.
- [53] Ö. Uludağ and F. Matthes. Large-scale agile development patterns for enterprise and solution architects. In *Proceedings of the European Conference on Pattern Languages of Programs 2020*, pages 1–22, 2020.
- [54] Ö. Uludag, S. Nägele, and M. Hauder. Establishing architecture guidelines in large-scale agile development through institutional pressures: A single-case study. 2019.
- [55] G. Van Der Vyver, A. Koronios, and M. S. Lane. Agile methodologies and the emergence of the agile organisation: a software development approach waiting for its time? In *Proceedings of the 7th Pacific Asia Conference on Information Systems (PACIS 2003)*, pages 1344–1358. University of South Australia, 2003.
- [56] A. Weissman and A. Zink-Kunnert. Organisation 4.0. In *Management 4.0—Unternehmensführung im digitalen Zeitalter*, pages 171–205. Springer, 2019.
- [57] E. Wińska and W. Dąbrowski. Software development artifacts in large agile organizations: a comparison of scaling agile methods. In *Data-Centric Business and Applications*, pages 101–116. Springer, 2020.
- [58] C. Wohlin. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, pages 1–10, 2014.

- [59] C. G. Worley, T. D. Williams, and E. E. Lawler III. *The agility factor: Building adaptable organizations for superior performance*. John Wiley & Sons, 2014.
- [60] R. Yuliana and B. Rahardjo. Designing an agile enterprise architecture for mining company by using togap framework. In *2016 4th International Conference on Cyber and IT Service Management*, pages 1–6. IEEE, 2016.