

FORSCHUNGSBEIRAT



Expertise des Forschungsbeirats der Plattform Industrie 4.0

Open Source als Innovationstreiber für Industrie 4.0

Empfohlene Zitierweise:

Forschungsbeirat der Plattform Industrie 4.0/acatech – Deutsche Akademie der Technikwissenschaften (Hrsg.): *Open Source als Innovationstreiber für Industrie 4.0*, 2022, DOI: 10.48669/fb40_2022-2

Impressum

Herausgeber

Forschungsbeirat der Plattform Industrie 4.0 /
acatech – Deutsche Akademie der Technikwissenschaften

Projektbüro

acatech – Deutsche Akademie der Technikwissenschaften
Geschäftsstelle
Karolinenplatz 4
80333 München

Autorinnen und Autoren

Fraunhofer-Institut für Materialfluss und Logistik – IML
Prof. Dr. Dr. h. c. Michael ten Hompel
Dr.-Ing. Michael Schmidt
Carina Culotta, M. Sc.
Dr.-Ing. Marius Brehler
Estelle Duparc, M. Sc.
Jens Leveling, M. Sc.
Dipl.-Inform. Andreas Nettsträter
Hendrik van der Valk, M. Sc.

Koordination

Lisa Hubrecht, acatech

Redaktion und Lektorat

Karola Klatt, Berlin

Gestaltung und Produktion

GROOTHUIS. Gesellschaft der Ideen und Passionen mbH
für Kommunikation und Medien, Marketing und Gestaltung;
groothuis.de

Bildnachweis

Getty Images/iStockphoto

Stand

Mai 2022



Plattform Industrie 4.0



acatech – Deutsche Akademie
der Technikwissenschaften

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung

 **acatech**

DEUTSCHE AKADEMIE DER
TECHNIKWISSENSCHAFTEN

 **Fraunhofer**
IML

Der **Forschungsbeirat der Plattform Industrie 4.0** berät als strategisches und unabhängiges Gremium die Plattform Industrie 4.0, ihre Arbeitsgruppen und die beteiligten Bundesministerien, insbesondere das Bundesministerium für Bildung und Forschung (BMBF).

Als **Sensor** von Entwicklungsströmungen beobachtet und bewertet der Forschungsbeirat die Leistungsprofilentwicklung von Industrie 4.0 und versteht sich als **Impulsgeber** für künftige Forschungsthemen und Begleiter beziehungsweise Berater zur Umsetzung von Industrie 4.0. Dabei konzentriert sich der Forschungsbeirat inhaltlich auf folgende **Themenfelder im Kontext von Industrie 4.0**:

- Wertschöpfungsnetzwerke
- Technologische Wegbereiter
- Neue Methoden und Werkzeuge
- Arbeit und Gesellschaft

Hier setzen die **Expertisen des Forschungsbeirats** an. Vor dem Hintergrund der Themenfelder werden klar umrissene Problemstellungen aufgezeigt, Forschungs- und Entwicklungsbedarfe definiert und Handlungsoptionen für eine erfolgreiche Gestaltung von Industrie 4.0 abgeleitet.

Die Expertisen liegen in der inhaltlichen Verantwortung der jeweiligen Autorinnen und Autoren. Alle bisher erschienenen Publikationen des Forschungsbeirats stehen unter www.acatech.de/projekt/forschungsbeirat-industrie-4-0/ zur Verfügung.

Inhalt

Management Summary.....	4
1. Einleitung.....	5
2. Ausgangslage und Zielsetzung.....	6
3. Methodisches Vorgehen und theoretischer Rahmen.....	9
3.1. Methodisches Vorgehen.....	9
3.2. Zugrunde liegendes Verständnis von Industrie 4.0.....	11
4. Allgemeine Grundlagen, Status quo und Ansätze von Open Source.....	13
4.1. Grundlagen der Open Source Software.....	13
4.2. Verbreitung und wirtschaftliche Bedeutung von Open Source Software.....	16
4.2.1. Wirtschaftliche Bedeutung von Open Source für KMU sowie Start-ups.....	17
4.2.2. Geschäftsmodelle und kommerzielle Nutzung von Open Source Software.....	18
4.2.3. Indirekte Mehrwerte von Open Source Software: Open Innovation und Standardisierung.....	19
5. Open Source-Ökosysteme und Open Source-Strukturen.....	21
5.1. Organisationsstrukturen von und Prozesse in Open Source-Projekten.....	21
5.2. Einbettung von Open Source-Projekten in Unternehmen.....	25
5.3. Eigenschaften und Messgrößen gesunder Open Source-Projekte und Open Source-Ökosysteme.....	26
6. Strukturierte Erfassung von Open Source-Initiativen.....	29
6.1. Open Source-Taxonomie.....	29
6.2. Archetypen von Open Source-Projekten in Industrie 4.0.....	32
7. Open Source Software in Industrie 4.0.....	34
7.1. Übersicht relevanter Projekte.....	34
7.1.1. Apache Software Foundation und relevante Projekte.....	34
7.1.2. AWS IoT Device SDKs.....	35
7.1.3. BaSys 4 / Eclipse BaSys.....	36
7.1.4. Catena-X / Eclipse Tractus-X.....	37
7.1.5. Die Eclipse Foundation und das Top Level-Projekt Internet of Things.....	38
7.1.6. FIWARE.....	41
7.1.7. IndustryFusion.....	42
7.1.8. International Data Spaces.....	43
7.1.9. open62541.....	44
7.1.10. Open Manufacturing Platform.....	45
7.1.11. Open Source Automation Development Lab.....	46
7.1.12. Robotic Operating System.....	47
7.2. Bewertung des Stands von Open Source im Kontext von Industrie 4.0.....	48
8. Expertenperspektiven: Chancen und Herausforderungen von Open Source Software im Kontext von Industrie 4.0.....	49
8.1. Bedeutung, aktuelle Trends und Eignungsbereiche von Open Source Software im Kontext von Industrie 4.0.....	49
8.2. Erfolgsfaktoren und Attraktivität von Open Source-Ökosystemen und Entwicklungs-Communities.....	50
8.3. Strategische Gründe für Open Source und ihre Relevanz in Abhängigkeit von der Unternehmensgröße.....	51
8.4. Bedeutung von Lizenzfragen.....	52
8.5. Open Source Software im Kontext von Standardisierung.....	53
8.6. Bedenken gegen den Einsatz von Open Source Software.....	54
8.7. Expertenstimmen: Handlungsoptionen zur Stärkung von Open Source Software.....	55

9. SWOT-Analyse von Open Source Software im Kontext von Industrie 4.0	57
9.1. Stärken: positive Einflussfaktoren für den Ausbau von Open Source Software im industriellen Umfeld.....	57
9.2. Schwächen: Ungenutzte Potenziale von Open Source Software für Industrie 4.0	58
9.3. Chancen: Förderliche Rahmenbedingungen zur Verbreitung von Open Source Software für Industrie 4.0	58
9.4. Risiken: Negative Einflussfaktoren für die Verbreitung von Open Source Software im Bereich Industrie 4.0.....	59
9.5. Zusammenfassung Stärken, Schwächen, Chancen und Risiken von Open Source Software für Industrie 4.0	60
10. Handlungsoptionen zur Stärkung von Open Source Software im Industrie 4.0-Kontext	62
10.1. Aufklärungskampagnen und Wissenstransfer	62
10.2. Forschungsbedarfe und fehlende Bildungsangebote	62
10.3. Nachhaltige Ausgestaltung von Fördermaßnahmen und Leuchtturmprojekten	63
10.4. Stärkung von Organisationsstrukturen und Koordination sowie inhaltliche Ausrichtung von Open Source-Initiativen im Industrie 4.0-Kontext.....	63
10.5. Abschließende Betrachtungen und Implikationen für Unternehmen	64
Anhang	67
Literatur	70
Autorinnen und Autoren	76
Mitglieder des Forschungsbeirats	77

Management Summary

Open Source Software ist zu einem essenziellen Bestandteil der modernen Wirtschaft geworden. Nahezu alle aktuellen Softwarelösungen setzen in kleinerem oder auch größerem Umfang auf quelloffenen Lösungen auf, betten diese ein, ergänzen oder transformieren sie. Open Source-Lösungen sind so zu einem zentralen Baustein technologischen Fortschritts geworden. Sie sind entscheidend für eine digitale Souveränität Europas und die breite Realisierung der Vision Industrie 4.0. Aktuelle Erhebungen zeigen, dass in vielen Unternehmen wie auch im öffentlichen Sektor diese zentrale Bedeutung von Open Source Software für Innovationen wahrgenommen wird. Weit über 60 Millionen Beitragende allein auf GitHub, dem weltweit umfassendsten Dienst zur Verwaltung von Software-Entwicklungsprojekten, sowie die politische Zielsetzung in der *Open Source Software Strategy* der Europäischen Kommission und dem aktuellen Koalitionsvertrag zwischen SPD, Bündnis 90/Die Grünen und FDP bestätigen dies. Trotz dieser Initiativen und der damit verbundenen Aufmerksamkeit in aktuellen Debatten zeigen Erhebungen auch, dass Open Source Software bisher nur von verhältnismäßig wenigen (deutschen) Industrieunternehmen strategisch behandelt wird.

Die vorliegende Expertise wurde von Mitarbeiterinnen und Mitarbeitern des Fraunhofer-Instituts für Materialfluss und Logistik (IML) und des Lehrstuhls Industrielles Informationsmanagement der Technischen Universität Dortmund verfasst. Auf der Basis von Desk Research und den Ergebnissen der Auswertung von 22 Experteninterviews beleuchtet sie aktuelle Fragestellungen und Hemmnisse im Kontext von Open Source für Industrie 4.0. Der Schwerpunkt liegt besonders darauf, positive Erfahrungen und Strategien im Kontext von Open Source zu erfassen und Handlungsoptionen daraus abzuleiten. Die Auswahl der Expertinnen und Experten setzt thematisch wie fachlich auf Vielfalt. Sie kommen aus Start-ups, dem Mittelstand und internationalen Großunternehmen, aus Forschungseinrichtungen, Open Source-Genossenschaften, -Stiftungen und -Vereinen sowie aus Standardisierungsorganisationen, Verbänden und Bundeseinrichtungen. Weiterhin werden Fallanalysen aktuell laufender Open Source-Projekte mit klarem Industrie 4.0-Bezug vorgestellt.

In einem ersten Schritt werden allgemeine Grundlagen und aktuelle Open Source-Ansätze beschrieben. Es zeigen sich dabei sowohl eine Vielzahl an Einsatzszenarien von Open Source-Komponenten im Rahmen der Produktentwicklung als auch verschiedene funktionierende und etablierte Geschäftsmodelle für Industrie 4.0. Open Source und Themen wie Open Innovation und Open Standards teilen Schnittmengen und stehen in Wechselwirkung miteinander. Eine Betrachtung dieser gegenseitigen Einflüsse zeigt, dass die Bereitstellung von Code sowie das Engagement in Open Source Communities nicht nur direkte Mehrwerte im Sinne eines Geschäftsmodells für Unternehmen bieten, sondern vor allem indirekte, sich gegenseitig bedingende Mehrwerte für Unternehmen schaffen.

Die sich anschließende Darstellung von Messgrößen und die Klärung von Erfolgsfaktoren gesunder Open Source-Ökosysteme

fördern eine Vielzahl an Möglichkeiten zutage, Projekte anhand von Kennwerten aktiv zu steuern und den Prozess der Auswahl geeigneter Open Source-Komponenten zu unterstützen. Besonders hervorzuheben ist in diesem Zusammenhang die hohe Bedeutung der Community und von Aktivitäten, die die Community stärken, zum Beispiel Software-Dokumentation und Foren-Kommunikation. Diese Erkenntnis hat sich im Rahmen der Experteninterviews durchgängig bestätigt.

Darüber hinaus wird im Rahmen der Expertise ein Framework zur Einordnung und Beschreibung von Open Source-Projekten in Form einer Taxonomie entlang von drei Metadimensionen (Geschäftsmodell, Ökosystem, Governance) entwickelt. Diese Taxonomie und die hierauf basierende Identifikation sogenannter Open Source-Archetypen für Industrie 4.0 ermöglicht eine kompakte Analyse und Darstellung zentraler Eigenschaften von Open Source-Projekten. Dieser Beschreibungsansatz wird in den Fallanalysen anhand der ausgewählten Projekte demonstriert.

Die dargestellten Grundlagen wurden gemeinsam mit den Expertinnen und Experten diskutiert und von diesen in ihren praktischen und betrieblichen Erfahrungshorizont eingeordnet. In der Expertenperspektive herrscht eine positive Grundhaltung gegenüber Open Source vor, die auf vielschichtigen persönlichen Open Source-Erfolgen gründet. Der im Rahmen des Desk Research gewonnene Eindruck, dass sich der Einsatz von Open Source Software nicht auf bestimmte Technologiebereiche beschränken sollte, beziehungsweise dass per se keine klare Präferenz für spezifische Technologieschwerpunkte existiert, wurde bestätigt. Formuliert wurden Präferenzen für quelloffene Lösungen in den thematisch-technologischen Bereichen Kommunikation und Konnektivität, Prozessautomatisierung und -flexibilisierung sowie Daten und Analytik. Herausfordernd sehen die Expertinnen und Experten nach wie vor rechtliche Fragestellungen, wie Open Source-Compliance und Lizenz(in)kompatibilitäten. Darüber hinaus kann der Einsatz von Open Source-Lösungen Änderungen an Geschäftsstrategien und -modellen sowie Kooperations- und Entwicklungsprozessen erforderlich machen.

Die durch eine SWOT-Analyse ergänzten Expertenperspektiven werden zum Ausgangspunkt von Handlungsoptionen. In vier Schwerpunkten (Aufklärungskampagnen und Informationsbereitstellung, Forschungsbedarfe und Bildungsangebote, Fördermaßnahmen und Leuchtturmprojekte sowie eine fachliche und methodische Stärkung von Industrie 4.0) werden 15 potenzielle Maßnahmen beschrieben, die geeignet sind, drängende Fragen zum Umgang mit Open Source im Industrie 4.0-Umfeld zu klären, Hemmnisse in Bezug auf die Verwendung zu mindern und damit insgesamt die Innovationskraft der Industrie zu steigern. Dazu gehören der Aufbau einer europäischen Anlaufstelle für Patent- und Lizenzfragen, ein genereller „Public money, public code“-Ansatz für öffentlich geförderte Forschungsprojekte und die Stärkung von Vorhaben im Bereich offener Integrationsplattformen und Fabrik-Betriebssystemen.

1. Einleitung

Ein neues Zeitalter hat begonnen: Die Menschheit ist zur bestimmenden Kraft bei der Gestaltung des Planeten Erde geworden. Erstmals übersteigt die durch Menschen geschaffene „anthropogene Masse“, das heißt alle vom Menschen gefertigten Dinge von der Stecknadel bis zur Fabrik, die lebende Biomasse, bestehend aus Pflanzen, Tieren und anderen Lebewesen.¹ Diese Masse allen anthropogenen Materials entsteht zum großen Teil, indem im primären Sektor (Urproduktion) Rohstoffe für die Industrie abgebaut und anschließend von ihr verarbeitet werden. Immer häufiger werden diese Rohstoffe einer zirkulären Rückgewinnung zugeführt und anschlussverwertet. Gleichzeitig stellen globale, in der Rohstoffnutzung begründete Herausforderungen wie der Klimawandel unausweichliche Bedingungen an resiliente Wertschöpfungsnetzwerke und erfordern eine sozial, ökonomisch und ökologisch nachhaltige Entwicklung der Industrie – in Deutschland, Europa und weltweit.

Technologische Innovationen und Digitalisierung im Allgemeinen sowie Industrie 4.0 im Besonderen sollen dazu beitragen, „die Welt wieder in Balance [zu] bekommen“.² Infrage steht nicht mehr, welche Technologien großes Potenzial haben. Sie sind im Prinzip bekannt und werden von strategischen Vorausschau (Foresight Reports) klar benannt. In sogenannten Hype Cycles lässt sich die Aufmerksamkeit, die sie im Zeitverlauf erfahren, gut darstellen. Neben vielen anderen Technologien sind es vor allem:³

- Künstliche Intelligenz und maschinelles Lernen – vertrauensvoll und erklärbar, eingebettet und integriert (embedded und embodied), menschenzentriert und auf dem neuesten Stand
- Digitale Ökosysteme und Datenräume
- Distributed Cloud und Edge Computing
- Mobile Robotik und autonome Systeme
- Augmented und Virtual Reality
- Blockchain (Distributed Ledger)
- Automatisiertes Verhandeln (Smart Contracting)

Diese Themen sind in ein weltweit anerkanntes Ziel-Framework eingebettet und in den Forschungsabteilungen angekommen.⁴ Von höchstem Interesse für Wissenschaft und Wirtschaft ist selbstverständlich die Frage nach der konkreten Ausgestaltung und Implementierung oben genannter Technologien. Doch in den letzten

Monaten treten immer stärker auch andere Fragen in den Vordergrund: Wie können und wollen wir diesen Wandel bewerkstelligen? Mit welcher Form der Zusammenarbeit können Teilhabe und Teilnahme vieler erreicht und Verantwortung geteilt werden?

Zwei Dinge sind dabei offensichtlich: Die Dimension der Herausforderung übersteigt das Maß, das ein einzelnes Unternehmen oder eine einzelne Nation imstande ist zu leisten – und seien sie noch so groß. Daraus folgt: Zusammenarbeit ist notwendig und braucht Orte und Ökosysteme, damit sie geprägt sein kann von neuen Formen: agiler, interdisziplinärer, lernender und langfristiger.⁵

Die Antwort auf dieses WIE lautet immer häufiger, auch in und aus der Industrie: „Open Source“, die freie Verfügbarkeit von Software-Quellcodes. Das Interesse an Open Source-Initiativen ist groß – von Gaia-X und International Data Spaces (IDS) über die Open Manufacturing Platform oder das Automotive Grade Linux bis hin zur Open Logistics Foundation. Gleichzeitig ist Open Source auch die Antwort auf die Frage, wie Europa seine digitale Souveränität, das heißt seine technologische Wahlfreiheit, behalten beziehungsweise in Teilen zurückgewinnen kann. Auch die Europäische Union und die neue Bundesregierung bringen dies in einem Strategiepapier und in ihrem Koalitionsvertrag zum Ausdruck.⁶

Genau hier setzt die vorliegende Expertise an: Auf der Basis von Desk Research und Expertengesprächen werden zentrale Punkte herausgearbeitet und analysiert, die für Open Source im Kontext von Industrie 4.0 zum jetzigen Zeitpunkt charakteristisch sind. Dabei steht im Zentrum der Arbeit die Motivation, einen Einblick in Felder zu verschaffen, in denen große Unklarheiten und Ungewissheiten hemmend wirken – seien es Bedenken gegenüber Open Source im Allgemeinen, Fragen nach einer grundsätzlichen Open Source-Eignung im Speziellen oder fehlendes Wissen, wie Projekte effizient aufgesetzt und zielführend geleitet werden können.

Diese Expertise liefert keine zahlenmäßig exakte oder statistisch repräsentative Erhebung, sondern vielmehr ein breit gefächertes, vielfältiges Stimmungsbild darüber, welche Chancen beziehungsweise Potenziale Open Source für Industrie 4.0 bietet. Außerdem werden Open Source-Projekte vorgestellt, die eine hohe Relevanz für Industrie 4.0 haben. Hieraus werden Handlungsoptionen und Maßnahmen für Industrieunternehmen, Politik und Wissenschaft für die Gestaltung von Rahmenbedingungen, aber auch für das Handeln in Unternehmen abgeleitet. Diese gilt es in Zukunft aufzugreifen und umzusetzen, um das zweifellos vorhandene und noch weitgehend nicht gehobene Potenzial von Open Source für Industrie 4.0 nutzbar zu machen.

1 Vgl. Elhacham et al. 2020.

2 Vgl. Laguna de la Vera/Ramge 2021.

3 Vgl. European Commission 2019, Gartner 2020a.

4 Vgl. United Nations 2015.

5 Vgl. Forschungsbeirat/acatech 2019b.

6 Vgl. Europäische Kommission 2020b, Koalitionsvertrag 2021.

2. Ausgangslage und Zielsetzung

Das Internet ist wohl die größte Innovation der letzten Jahrzehnte. Es basiert zu großen Teilen auf quelloffener Infrastruktur. Quelloffenheit hat auch seine Entwicklung stark beeinflusst. Heute laufen Cloud Server basierend auf freier und quelloffener Software (Free and Open Source Software, kurz: FOSS) wie Linux, Apache, Docker oder Kubernetes (siehe Abbildung 1).⁷ Open Source Software (OSS) ist zu einem essenziellen Bestandteil der modernen Wirtschaft geworden. Schätzungen gehen davon aus, dass bis zu 90 Prozent der modernen (Unternehmens-)Software aus OSS besteht.⁸ Diese starke Abhängigkeit von OSS ist nicht nur im öffentlichen, sondern auch im privaten Sektor und bei sowohl technischen als auch nicht technischen Unternehmen gleichermaßen verbreitet.⁹

Der Open Source-Ansatz hat beispiellos dazu beigetragen, dass schnell und effizient neue Standards geschaffen und an die Bedürfnisse der Nutzenden angepasst werden. Moderne Technologien, wie Quellcodeverwaltung (zum Beispiel git), Continuous Integration (CI) oder Continuous Deployment und Delivery (CD), ermöglichen es, IT- und Softwareprodukte basierend auf OSS schnell zu entwickeln und aktuell zu halten.

Open Source Communities dokumentieren in ihren Foren Entwicklungsschritte, Probleme und Lösungen in nie dagewesenen

Maßstäben. Auch in aktuellen Trendbereichen wie Künstliche Intelligenz (KI) setzen sich quelloffene Lösungen, wie TensorFlow, PyTorch oder Keras, durch. Der Open Source-Ansatz beweist somit, dass er echte Innovationskraft besitzt.

Die gemeinschaftliche Entwicklung und Nutzung von OSS ist gekennzeichnet durch eine hohe Effizienz sowie die Teilhabe unterschiedlicher Gruppen und Unternehmen. Es entstehen gemeinsame Standards, Tools und Services, die eine erfolgreiche kommerzielle Nutzung in Unternehmen ermöglichen. Dies trifft insbesondere auch auf sich entwickelnde Technologien (Emerging Technologies), wie das Internet der Dinge (Internet of Things, kurz: IoT) und KI zu, die zugleich Grundlage und Treiber des Wandels sind, den wir mit „Industrie 4.0“ bezeichnen.¹⁰

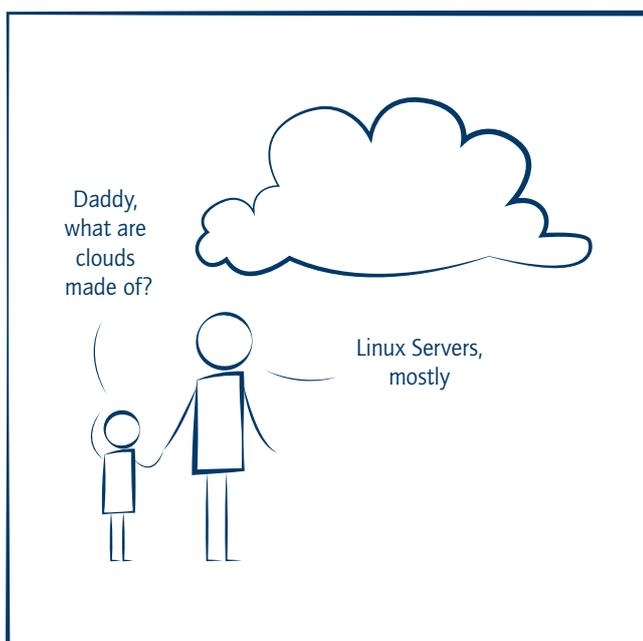
Technologischer Wandel – die zeitliche Koinzidenz der Entwicklungen ist entscheidend

KI-Algorithmen werden am Ende der 2020er Jahre im industriellen Bereich – und nicht nur dort – automatisiert entscheiden, regeln und kontrollieren. Um sie herum erobern plattformbasierte Geschäftsmodelle den Business-to-Consumer(B2C)-Markt. Den Vorbildern aus dem Silicon Valley folgend, etablieren sie sich inzwischen zunehmend auch im Business-to-Business(B2B)-Sektor. Plattformen werden in Zukunft (industrielle) Daten horten und Wissen generieren. Schwärme autonomer Roboter werden ihre Umgebung erkunden. Maschinen werden miteinander verhandeln und sich selbst organisieren. KI-Algorithmen und ihr maschinelles Lernen werden den Wettbewerb bestimmen.¹¹

Autonom interagierende Entitäten sind das Zielbild der industriellen Entwicklung. Seit jeher getrieben durch die Hardware-Entwicklung digitaler Halbleiter, wie Speicher, Low Power-Sensoren und Prozessoren, tritt nun die Automatisierung vollständiger Prozesse und Wertschöpfungsnetzwerke auf Basis autonomer Entitäten im Soft- und Hardwarebereich in den Vordergrund. Entscheidend ist dabei die zeitliche Koinzidenz einer Fülle technischer Entwicklungen:¹²

- IoT in Verbindung mit KI-tauglichen Geräten (Edge AI Devices)
- echtzeitfähige Vernetzung (5G, Wi-Fi 6)
- KI-basierte Plattformen als Service (AI Platform as a Service)
- Blockchain (Distributed Ledger) und automatisiertes Verhandeln (Smart Contracting)

Abbildung 1: Open Source und Cloud Computing



Quelle: eigene Darstellung

7 Vgl. Open Source 2022.

8 Vgl. Sonatype 2016, Synopsis 2020, Gartner 2019.

9 Vgl. The Linux Foundation 2018, Finley 2016.

10 Vgl. Bitkom 2019, Synopsis 2020.

11 Vgl. ten Hompel et al. 2022.

12 Vgl. ten Hompel/Henke 2022.

- Schwärme autonomer Roboter¹³
- Virtualisierung und Simulation mithilfe von KI-Methoden
- immersive Technik, die Menschen mit KI verbindet und in virtuelle Umgebungen eintauchen lässt, wie Augmented Reality (AR) und Virtual Reality (VR)
- Cognitive-Computing-Technologien und -Methoden
- Quantum Computing

In der industriellen Wertschöpfung und insbesondere in der eng mit ihr verwobenen (industriellen) Logistik scheinen all diese Technologien nun gleichzeitig durchzubrechen.

Open Source als Lösungsansatz

Der Siegeszug der oben skizzierten technischen Entwicklungen sowie der Plattformökonomie fördert die Erkenntnis, dass für den wirtschaftlichen Erfolg die Teilnahme und Teilhabe Vieler zu sichern sind. Die universellen Herausforderungen dieser digitalen Transformation übersteigen jedoch in ihren Dimensionen das Maß, das ein einzelnes Unternehmen zu leisten imstande ist – und sei es noch so groß.

Zugleich ist „Sharing“ das Leitmotiv einer neuen Generation von Entwicklerinnen und Entwicklern, die mit dem Prinzip des Tauschens und Teilens im Internet aufgewachsen sind und eine andere Logik des Gebens und Nehmens verinnerlicht haben: „Nutzen statt Besitzen“ – das ist ihr Motto. Das Prinzip hat sich auf große Bereiche der Wirtschaft ausgedehnt und ist zur Grundlage neuer Wertschöpfungsmodelle geworden.¹⁴ So ist es auch nur folgerichtig, dass im Zuge der politischen Debatte zur digitalen Souveränität, vereinfacht verstanden als technologische Wahlfreiheit, Open Source zunehmend in den Fokus rückt.¹⁵

Die Open Source-Bewegung folgt dem Sharing-Leitmotiv. Ihr Hauptziel ist eine frei zugängliche Bereitstellung von Quellcode, damit Menschen und Unternehmen die Möglichkeit haben, diesen zu nutzen, anzupassen und zu verbreiten. Auch die Veröffentlichung von Bauplänen als Open Hardware oder die Bereitstellung und Nutzung von Daten als Open Data sind Ausdruck des Sharing Mindsets, ebenso wie offene Innovationsprozesse mit firmeninternen und externen Kräften (Open Innovation). Allen diesen Strömungen gemein ist das begründete Vertrauen darauf, in intakten und offenen Ökosystemen Geschäftspotenziale gemeinsam besser zu heben, zum Beispiel durch größere Innovationskraft, bessere Stabilität

und IT-Sicherheit oder durch die Vermeidung von Lizenzkosten etc.¹⁶

OSS ist mittlerweile ein fester Bestandteil der digitalen Wirtschaft und nahezu sämtlicher Innovationsprozesse – und zwar grenzüberschreitend und unter Beteiligung zahlreicher und vielfältiger Unternehmen. Sie ist aus der heutigen Welt nicht mehr wegzudenken. Es scheint offensichtlich zu sein, dass die digitale Transformation ohne den Einsatz von Open Source nicht gelingen kann.

Dabei ist OSS alles andere als ein neues Prinzip. Von Anbeginn der industriellen beziehungsweise gewerblichen Software-Entwicklung in den 1950er bis in die 1980er Jahre war es üblich, dass Käuferinnen und Käufer den Quellcode aller von ihnen verwendeten Programme besaßen und die Erlaubnis und die Möglichkeit hatten, ihn für den eigenen Gebrauch zu verändern. Erst ab etwa Mitte der 1970er Jahre kamen zunehmend geschlossene beziehungsweise proprietäre Softwareprodukte (Closed Source Software) auf den Markt und setzten sich ab dem Jahr 1983, als IBM sich im Zuge seiner *Object Code Only Policy* entschloss, Quellcode nicht mehr verfügbar zu machen, durch.¹⁷ Gleichzeitig entstand mit der Gründung des GNU-Projekts, das ein freies unix-ähnliches Betriebssystem schaffen wollte, eine Gegenbewegung, die zum Startpunkt von Open Source Software wurde, wie sie hier betrachtet wird.¹⁸

Trotz eines grundsätzlich hohen Potenzials von Open Source für die Entwicklung von Software- und Hardwarelösungen, spielt Open Source in Diskussionen zu Industrie 4.0 noch keine große Rolle. Dies geht einher mit einem Mangel an präskriptivem Wissen zum Einsatz von Open Source in Industrie 4.0. Open Source für Industrie 4.0 soll daher in dieser Expertise anhand folgender Fragen untersucht werden:

1) Status quo von Open Source und OSS-Projekten in Industrie 4.0

- Wie sieht die Projektlandschaft für Industrie 4.0 aus? Welche Projekte beziehungsweise welche Art von Projekten gibt es?
- Wie wird Open Source von Expertinnen und Experten aus der Industrie bewertet? Welche Hindernisse und Potenziale werden gesehen?

2) Wodurch zeichnen sich erfolgreiche und lebendige Open Source-Ökosysteme aus?

- Organisationen und Projekte: Was ist State of the Art in den Bereichen Governance, Community Management und Code of Conducts?

13 Das Fraunhofer IML entwickelte beispielsweise in Zusammenarbeit mit der KION Group im Rahmen eines vom Bundesministerium für Verkehr und digitale Infrastruktur geförderten Projekts den LoadRunner®, einen Prototypen eines autonomen Schwarmroboters, der Pakete sortiert.

14 Vgl. ten Hompel/Henke 2022.

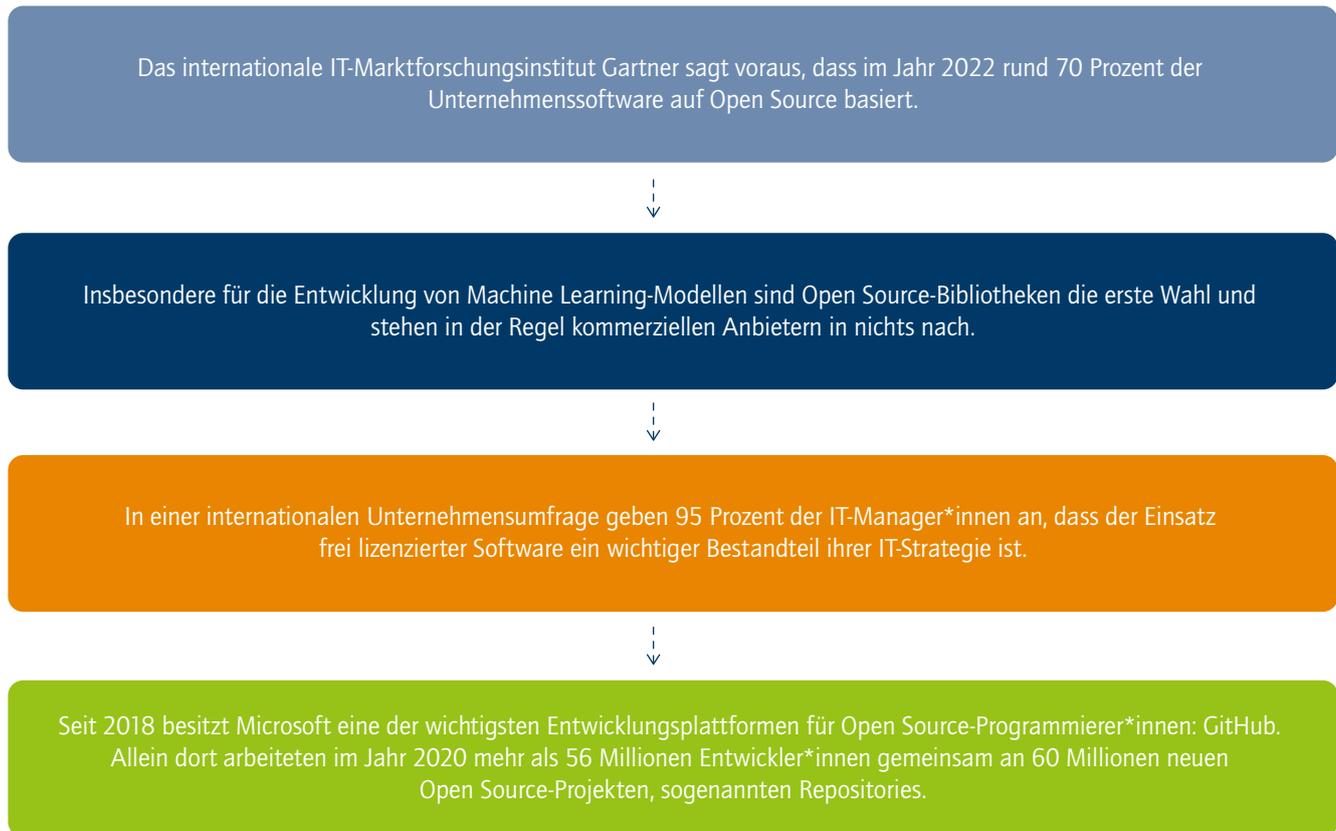
15 Vgl. acatech 2021.

16 Vgl. Al Marzouq et al. 2005, Schreieck et al. 2016.

17 Vgl. IBM 1983.

18 Vgl. Stallman 1983.

Abbildung 2: Open Source durchdringt immer weitere Bereiche



Quelle: eigene Darstellung in Anlehnung an Sommer 2021

- Open Source in Unternehmen: Wie organisieren Unternehmen intern ihre Open Source-Entwicklungen?
- Messgrößen: Anhand welcher Kennzahlen können Projekte bewertet und gesteuert werden?

3) Welche Themen im Umfeld von Industrie 4.0 sind geeignet für gemeinsame Open Source-Entwicklungen? Bei welchen Themen lohnt sich die Entwicklung quelloffener Software dagegen nicht?

- Wirtschaftlich/unternehmerisch: Welche (unternehmerischen) Strategien zum Umgang mit OSS haben sich etabliert und sind für Industrie 4.0 relevant? Welche Grenzen von Open Source existieren in Industrie 4.0?
- Inhaltliche Schwerpunkte: Welche (Technologie-)Bereiche profitieren besonders von Open Source als Lizenzmodell und Kooperationsform?

- Open Source als Innovationstreiber: Welche Rolle spielt OSS bei der Definition und Etablierung von Industriestandards, der Etablierung und Diffusion neuer Technologien?

Zur Beantwortung dieser Fragen werden zunächst allgemeine Grundlagen und aktuelle Open Source-Ansätze beschrieben. Es werden Geschäftsmodelle sowie Mehrwerte von Open Source, Open Innovation und Open Standards erläutert. Hieran schließen sich eine Darstellung von Messgrößen gesunder Ökosysteme, die Erstellung eines Rahmenwerks zur Einordnung von Open Source-Initiativen sowie die Beschreibung Industrie 4.0-relevanter Open Source-Projekte an. Zur Vertiefung des Wissenstands wurden 22 Interviews mit Expertinnen und Experten aus dem Open Source-Umfeld geführt. Die Ergebnisse dieser Gespräche werden durch eine SWOT-Analyse ergänzt. Sie bildet den Ausgangspunkt für die Formulierung von Handlungsoptionen in den vier Schwerpunktfeldern Aufklärungskampagnen und Informationsbereitstellung, Forschungsbedarfe und Bildungsangebote, Fördermaßnahmen und Leuchtturmprojekte sowie eine fachliche und methodische Stärkung von Industrie 4.0.

3. Methodisches Vorgehen und theoretischer Rahmen

3.1. Methodisches Vorgehen

Die grundlegende Motivation zur Erstellung dieser Expertise bildet der Mangel an richtungsweisendem Wissen zum Einsatz von Open Source in Industrie 4.0. Hieraus lässt sich die Forderung nach Vorgaben, Hilfsmitteln und Handlungsempfehlungen für den Einsatz des Open Source-Prinzips und für die Durchführung von standardisierten und strukturierten Open Source-Initiativen ableiten. Eine Möglichkeit zur Ermittlung solcher Empfehlungen bietet die gestaltungsorientierte Forschung (Design Science Research, kurz: DSR).¹⁹ Der Forschungsansatz umfasst sechs Schritte, die in der Synthese das Artefakt, hier Handlungsempfehlungen, gestalten.²⁰ Die sechs Schritte dieser Expertise werden im Folgenden in Tabelle 1 dargestellt. Sie werden durch ergänzende Forschungsmethoden wissenschaftlich fundiert und teilweise wiederholt durchlaufen, um sich dem Ergebnis schrittweise anzunähern.

Zunächst wird im ersten Schritt eine strukturierte Literaturanalyse durchgeführt, deren Ziel die Identifikation der Forschungslücke ist.²¹ Das Wissen im Themenfeld wird in einer Taxonomie gegliedert, die nach dem weit verbreiteten und etablierten Vorgehen von Nickerson et al. erstellt wird.²² Zudem werden auf der Basis einer strukturierten Recherche Archetypen von Open Source-Initiativen identifiziert und Industrie 4.0-spezifisch aufbereitet. Diese Analyse und Konzeptionierung des Wissensfelds ermöglicht eine

ganzheitliche Betrachtung des Status quo von Open Source in Industrie 4.0. Getestet wird die Taxonomie in der Beschreibung von zwölf Fallbeispielen von Open Source-Projekten und -Initiativen. Dieses Wissen wird durch Interviews von Expertinnen und Experten erweitert. Dies erlaubt das Zeichnen eines sehr detaillierten und aktuellen Bilds über den Stand von Open Source in Industrie 4.0.

Die Entwicklung der Gestaltungsprinzipien basiert auf insgesamt 22 im Rahmen dieser Expertise durchgeführten Interviews mit Expertinnen und Experten aus Industrie, angewandter Forschung und öffentlichen Einrichtungen. Die Interviews werden halbstrukturiert durchgeführt, das heißt sie orientieren sich an einem Frageskript, lassen jedoch auch ausreichend Freiraum für situationsbedingte Abweichungen im Gespräch.²³ Das Skript ist in sieben thematische Blöcke unterteilt. Nach ersten erfolgten Interviews wird der Fragebogen im Team evaluiert und gegebenenfalls werden Anpassungen vorgenommen. Der Empfehlung von Myers und Newman folgend beginnt das Skript mit persönlichen Fragen, um die Interviewten an das Format zu gewöhnen.²⁴ Die weiteren Abschnitte beschäftigen sich mit Einstiegsfragen zum Thema Open Source, Fragen zu Communities und Ökosystemen, strategischen Aspekten, zum Beispiel Lizenzen, Firmenstrukturen im Zusammenhang mit Open Source, Aspekte der Standardisierung, sowie Abschlussfragen. Zudem werden die Leitlinien für qualitative Experteninterviews angewandt.²⁵ Bei den Interviewten handelt es sich um ausgewiesene Expertinnen

Tabelle 1: Sechsschrittiges Vorgehen der Expertise

Schritt	DSR-Gegenstand	Umsetzung
1	Identifikation des Forschungsproblems	Strukturierte Literaturanalyse / Bildung Taxonomie / Synthese Archetypen / Erhebung Status quo / Erprobung der Taxonomie / Beschreibung von Fallbeispielen
2	Definition des Lösungsansatzes	Festlegung auf Gestaltungsprinzipien als zentrales Ergebnis der Studie
3	Gestaltung und Entwicklung	Iterative Entwicklung des Fragebogens Folgend Entwicklung der Gestaltungsprinzipien
4	Demonstration	Anwendung des Fragebogens / Feedback von Expertinnen und Experten
5	Evaluierung	Laufende interne Evaluierung der Ergebnisse
6	Kommunikation	Dissemination der Ergebnisse in Wirtschaft und Wissenschaft

Quelle: eigene Darstellung in Anlehnung an Sommer 2021

19 Vgl. Peffers et al. 2007.

20 Vgl. Rhyh/Blohm 2017.

21 Vgl. Vom Brocke et al. 2009.

22 Vgl. Nickerson et al. 2013, Glass/Vessey 1995.

23 Vgl. Fontana/Frey 2000.

24 Vgl. Myers/Newman 2007.

25 Vgl. Myers/Newman 2007.

und Experten des Themengebiets Open Source. Eine anonymisierte Charakterisierung der Teilnehmenden hinsichtlich ihrer Position, ihres Unternehmens oder ihrer Organisation und ihrem Bezug zu Industrie 4.0 zeigt Tabelle 2.

Für die Interviews wird eine Dauer von 60 Minuten angesetzt. Jedes Interview wird von zwei Personen aus der Forschungsgruppe geführt. Eine Person stellt die Fragen und moderiert das Interview, während die zweite Person auf einem strukturierten Notizzettel das Interview protokolliert. Durch das Vier-Ohren-Prinzip, Audioaufzeichnung und schriftliche Dokumentation können auch im Nachgang der Gespräche Ergebnisse erarbeitet werden.

Die Gestaltungsprinzipien werden in Anlehnung an die von Möller et al. vorgestellte Methode entwickelt.²⁶ Dazu werden nach dem sogenannte Reflective Approach die Erkenntnisse und Erfahrungen

der verschiedensten Initiativen im Open Source-Kontext durch die Interviews erfasst und daraus Herausforderungen und Best Practices synthetisiert. Die Interviewten werden zu Chancen und Hemmnissen von Open Source-Initiativen sowie zu den konkreten Projekten, die sie begleiten, befragt. Sie haben die Möglichkeit, mit einem sehr hohen Grad an Autonomie ihre Erfahrungen darzulegen. Die Fragenden greifen nur moderierend ein, um das Gespräch in groben Korridoren des Themenbezugs zu halten. Die Erkenntnisse und Erfahrungen können auf diese Weise uneingeschränkt und tiefgehend erfasst werden.

Ausgehend von den Einzelinterviews werden die Erkenntnisse synthetisiert. Darauf aufbauend werden die Handlungsempfehlungen entwickelt und mittels einer SWOT-Analyse, das heißt einer Betrachtung von Stärken und Schwächen sowie Chancen und Risiken, unter Berücksichtigung von politischen, wirtschaftlichen, sozialen, technologischen, ökologischen und gesetzlichen Faktoren bewertet.

Tabelle 2: Befragte Expertinnen und Experten (anonymisiert)

#	Position	Unternehmen/Organisation	Bezug zu Industrie 4.0 / Inhaltliche Schwerpunkte
1	Bereichsleitung Engineering	Forschungseinrichtung	Daten und Analytik
2	Gründer*in	Start-up	Daten und Analytik
3	Gründer*in	Start-up	Daten und Analytik
4	Open Source Programm Management	Konzern (Software)	Enterprise Resource Planning
5	Projektmanagement Industrie 4.0 und Open Source	Standardisierungsorganisation	Industrie 4.0 allgemein
6	Bereichsleitung Software	Verband	Informations-/ Kommunikationstechnologien
7	Director IT	Konzern (Automotive)	Informations-/ Kommunikationstechnologien
8	Bereichsleitung Vertrieb	Open Source-Stiftung	Konnektivität und Kommunikation
9	Geschäftsführung	Open Source-Stiftung	Konnektivität und Kommunikation
10	Software Architect	Konzern (Logistikdienstleistung)	Logistik, Softwareentwicklung
11	Leiter*in Forschung und Entwicklung	Mittelstand	Maschinenbau, Prozessautomation und -flexibilisierung
12	Partner*in	Kanzlei	Open Source-Lizenzen, Open Source Compliance
13	Open Source Maintainer	Forschungseinrichtung	Prozessautomation und -flexibilisierung
14	Technical Steering Committee OS Projekt	Forschungseinrichtung	Prozessautomation und -flexibilisierung
15	Bereichsleitung Business Development	Mittelstand	Prozessautomation und -flexibilisierung / Mensch-Technik-Interaktion
16	Gründer*in	Start-up	Prozessautomation und -flexibilisierung / Mensch-Technik-Interaktion
17	Geschäftsführung	Open Source-Genossenschaft	Prozessautomation und -flexibilisierung, Konnektivität und Kommunikation
18	CTO / Technische Leitung	Verband	Sicherheit und Vertrauen
19	Geschäftsführung	Mittelstand	Sicherheit und Vertrauen
20	Gründer*in	Start-up	Sicherheit und Vertrauen
21	Senior Software Engineer	Konzern (Soft- und Hardware)	Softwareentwicklung allgemein
22	Geschäftsführung	Bundesagentur	Technologie allgemein

26 Vgl. Möller et al. 2021.

3.2. Zugrunde liegendes Verständnis von Industrie 4.0

Ermöglicht durch die rasante Entwicklung unterschiedlicher Technologiebereiche ist eine allgegenwärtige unternehmensinterne und -übergreifende digitale Vernetzung unterschiedlicher Objekte, Produktionssysteme und ganzer Industriezweige zu beobachten. Im industriellen Bereich zeichnet sich der Wandel durch eine steigende *Prozessautomation und -flexibilisierung* aus. Im Gegensatz zu visionären Vorstellungen der 1970er-Jahre, die eine vollautomatisierte Fabrik mit minimalem menschlichem Einsatz vorhersahen, sollen in der Vision von Industrie 4.0 jedoch menschliche Akteure optimal in die intelligente Fabrik eingebunden werden. Technologie soll dazu dienen, dem Menschen zu assistieren (*Mensch-Technik-Interaktion*).²⁷ Vereinheitlichende (Kommunikations-)Standards, das IoT und gemeinsame Datenräume stellen die Grundlage für die unternehmensübergreifende *Konnektivität und Kommunikation* bereit. Durch die einhergehende allgegenwärtige Datenerfassung können große Datenmengen analysiert und weiterverarbeitet werden, um wertvolles Wissen im Rahmen von Big Data und mit KI-Methoden generieren zu können (*Daten und Analytik*). Gleichzeitig werden aufgrund der veränderten Rahmenbedingungen in Unternehmensnetzwerken die Erhöhung von *IT-Sicherheit und Vertrauen* durch innovative Konzepte (zum Beispiel Blockchain-Technologie, Security by Design oder sichere Datenräume) erforderlich. Diese fünf Technologiefelder bilden das hier zugrunde gelegte Verständnis von Industrie 4.0 (siehe Abbildung 3).

Der Begriff „Industrie 4.0“ wird gemäß der Plattform Industrie 4.0 wie folgt verstanden:

„Der Begriff Industrie 4.0 steht für die vierte industrielle Revolution, eine neue Stufe der Organisation und Steuerung der gesamten Wertschöpfungskette über den Lebenszyklus von Produkten. Dieser Zyklus orientiert sich an den zunehmend individualisierten Kundenwünschen und erstreckt sich von der Idee, dem Auftrag über die Entwicklung und Fertigung, die Auslieferung eines Produkts an den Endkunden bis hin zum Recycling, einschließlich der damit verbundenen Dienstleistungen.“

*Basis ist die Verfügbarkeit aller relevanten Informationen in Echtzeit durch Vernetzung aller an der Wertschöpfung beteiligten Instanzen sowie die Fähigkeit, aus den Daten zu jedem Zeitpunkt optimalen Wertschöpfungsfluss abzuleiten. Durch die Verbindung von Menschen, Objekten und Systemen entstehen dynamische, echtzeitoptimierte und selbst organisierende, unternehmensübergreifende Wertschöpfungsnetzwerke, die sich nach unterschiedlichen Kriterien wie beispielsweise Kosten, Verfügbarkeit und Ressourcenverbrauch optimieren lassen.“*²⁸

Abbildung 3: Technologiebereiche von Industrie 4.0



Quelle: eigene Darstellung

27 Vgl. Soder 2014.

28 Siehe Bitkom et al. 2015, S. 8.

Im Fokus von Industrie 4.0 stehen die horizontale Integration über Wertschöpfungsnetzwerke hinweg, die Vernetzung von Produktionssystemen (vertikale Integration) sowie die Durchgängigkeit des Engineerings entlang des gesamten Lebenszyklus.²⁹

Die **horizontale Integration** beschreibt die flexible Integration von IT-Systemen und Akteuren in ein Wertschöpfungsnetzwerk über unterschiedliche Prozessschritte hinweg. Hierfür soll eine durchgängige Kommunikationsbasis bestehend aus Integrations- und Kollaborationsplattformen sowie entsprechenden Standards geschaffen werden. Durch den unternehmensübergreifenden Daten- und Informationsaustausch entstehen flexible und anpassbare Wertschöpfungsnetzwerke, die schnell auf unvorhergesehene Ereignisse reagieren können.³⁰

In vielen Unternehmen werden IT-Systeme nach den Bedürfnissen einzelner Unternehmensebenen und -abteilungen isoliert voneinander strukturiert, sodass ein Datenaustausch über die einzelnen Unternehmensebenen hinweg erschwert wird. Die **vertikale Integration sowie die Vernetzung von Produktionssystemen** zielen

auf eine Datendurchgängigkeit zwischen IT-Systemen ab. Dadurch entsteht eine ganzheitliche Vernetzung von der Produktionsebene bis hin zu höheren Ebenen. Auf Produktionsebene werden durch einheitliche Standardschnittstellen ganze Produktionsanlagen miteinander verbunden, die eine herstellerunabhängige Kommunikation und Integration erlauben. In Echtzeit können hierdurch Daten gesammelt werden, die zur Optimierung von Produktions- und Geschäftsprozessen eingesetzt werden können und zu einer Verschmelzung unterschiedlicher IT-Ebenen führen.³¹

Durch den Einsatz von digitalen Zwillingen soll das Ziel der **Durchgängigkeit des Engineerings über den gesamten Lebenszyklus** eines Produkts erreicht werden. Von der Entwicklung über die Produktplanung bis hin zur Entsorgung sollen alle Prozessschritte miteinander verbunden und visualisiert werden. Die zumeist fragmentierten Informationsflüsse werden über den gesamten Lebenszyklus hinweg gebündelt, um die Entwicklungszeiten zu verkürzen und kurzfristige Änderungen in das Produkt einarbeiten zu können. Bereits heute ist Software für das Product Lifecycle Management weit verbreitet und ein wichtiges Tool, um Industrie 4.0 voranzutreiben.³²

29 Vgl. Plattform Industrie 4.0 2015.

30 Vgl. Huber 2016, Siepmann/Graef 2016, Plattform Industrie 4.0 2015.

31 Vgl. Plattform Industrie 4.0 2015.

32 Vgl. ebd.

4. Allgemeine Grundlagen, Status quo und Ansätze von Open Source

Die Entwicklung von Software und die damit verbundenen Innovationen sind eng mit der Geschichte der OSS verwoben. Auch die Entstehung der Technologien, die Industrie 4.0 zugrunde liegen, und deren Vernetzung fußen auf OSS und wären ohne diese nicht möglich gewesen. Beispielsweise bietet das freie Betriebssystem GNU/Linux die Basis für kleine eingebettete Rechnersysteme, sogenannte Embedded Systems, aber auch Servermaschinen oder Großrechenanlagen. Ebenso ermöglicht OSS den Erfolg von großen Plattformanbietern und Technologiekonzernen wie Facebook, Amazon und Google, die den freien Linux-Kernel in ihren eigenen Systemen verwenden.³³ OSS kann nicht nur die technologische Grundlage für den Unternehmenserfolg bieten, sondern damit verbunden auch verschiedene Geschäftsmodelle und strategische Vorteile erschließen. Auch wenn für die Nutzung von OSS keine Lizenzgebühren anfallen, bedeutet dies nicht, dass OSS keine Grundlage für Einnahmen darstellt. Geschäftsmodelle von Unternehmen wie Red Hat basieren allein auf dem Support und der Weiterentwicklung von OSS-Komponenten für Geschäftskunden.³⁴

Das wirtschaftliche Potenzial von OSS ist immens. Das hat auch die Europäische Kommission erkannt und 2020 unter dem Motto „Think Open“ eine neue Strategie zur Förderung und Stärkung von OSS in Europa verabschiedet. Diese Strategie zielt unter anderem darauf ab, die europäische Software-Entwicklung im Bereich der Informationstechnologien, länderübergreifende Innovationsbestrebungen und die Digitalisierung des öffentlichen Sektors zu stärken.³⁵ Auf Basis dieser Strategie hat die Europäische Kommission eine Studie in Auftrag gegeben, die gemeinsam vom Open Forum Europe und dem Fraunhofer-Institut für System- und Innovationsforschung (ISI) durchgeführt wurde. Diese Studie zeigt das enorme wirtschaftliche Potenzial der OSS-Entwicklung für Europa auf. Die Autoren kommen zu der Einschätzung, dass eine 10-prozentige Steigerung der Kontributionen zu OSS zu einer Steigerung des europäischen Bruttoinlandsprodukts (BIP) von 0,6 Prozent jährlich führen könnte.³⁶ Sie gehen davon aus, dass in der Europäischen Union gut 260.000 individuelle Kontributoren zur Entwicklung von Open Source beitragen – dies entspricht gut 8 Prozent der europäischen Erwerbstätigen im Bereich der Software-Entwicklung. Die Studie zeigt zudem, dass Unternehmen, die dezidiert Fachkräfte für die Entwicklung von OSS bereitstellen, einen wirtschaftlichen Vorteil aus der gemeinschaftlichen Entwicklung von Software ziehen. Warum genau Open Source eine solche wirtschaftliche Kraft hat und wie Vorteile aus der gemeinschaftlichen und offenen Entwicklung von Software generiert und in Geschäftsmodelle überführt werden können, wird

diese Expertise im Folgenden klären. Zunächst werden jedoch die Grundlagen und Grundprinzipien der OSS definiert.

4.1. Grundlagen der Open Source Software

Die Grundidee einer jeden OSS ist die Freiheit, diese Software auszuführen, sie zu analysieren, an die eigenen Bedürfnisse anzupassen und weiterzugeben, auch in veränderter Form. Die offene Bereitstellung des Quellcodes ist elementare Voraussetzung dafür und entscheidet darüber, ob eine Software als OSS definiert werden kann.³⁷ Die 1998 gegründete Open Source Initiative (OSI) ergänzt diese vier Grundfreiheiten um weitere sechs Eigenschaften und definiert damit einen Standard für OSS. Konkret lauten die zehn Prinzipien der OSS:³⁸

1. Freie Weitergabe: Die Lizenz der Software darf den Verkauf oder die Weitergabe von Software auch in Kombination mit anderer Software zusammen nicht behindern. Es dürfen keine Lizenzgebühren verlangt werden.

2. Verfügbare Quellcode: Der Quellcode muss frei verfügbar und zugänglich sein und die Weitergabe als Quellcode sowie in kompilierter Form ermöglichen.

3. Abgeleitete Arbeiten: Die Lizenz muss abgeleitete Versionen der Software sowie deren Verteilung unter derselben Lizenz wie auch die ursprüngliche Software erlauben.

4. Integrität des Autoren-Quellcodes: Die Lizenz darf die Weitergabe von Quellcode in modifizierter Form nur unter bestimmten Bedingungen einschränken. Die Weitergabe von Software, die aus modifiziertem Quellcode erstellt wurde, muss explizit erlaubt werden.

5. Keine Diskriminierung von Gruppierungen oder Personen: Die Software muss allen Gruppierungen oder Personen diskriminierungsfrei zur Verfügung stehen, unabhängig von Lokalität, Ethnie, Religion oder politischer Orientierung.

6. Keine Nutzungseinschränkung: Die Lizenz kann den Verwendungszweck oder das Anwendungsfeld der Software nicht einschränken.

33 Vgl. Bitkom 2016.

34 Vgl. Riehle 2019.

35 Vgl. Europäische Kommission 2020b.

36 Vgl. Blind et al. 2021.

37 Vgl. Bitkom 2016.

38 Vgl. Open Source Initiative 2021.

7. Lizenzerteilung: Das Programm, das unter einer Lizenz verteilt wird, muss für alle nutzbar sein, ohne dass eine Registrierung oder der Erwerb von weiteren Lizenzen notwendig sind.

8. Produktneutralität: Die erteilten Rechte dürfen nicht Teil einer bestimmten Softwareverteilung sein und müssen produktneutral gestaltet werden.

9. Keine Einschränkung anderer Software: Die Lizenz darf keinerlei Einschränkungen für andere Software darstellen und beispielsweise verlangen, dass die Software nur mit bestimmter anderer Software vertrieben werden darf.

10. Technologieneutralität: Die Verteilung der Lizenz muss technologieneutral erfolgen und darf sich nicht auf ein bestimmtes Medium beschränken.

Die genaue Ausgestaltung und Spezifikation der Open Source-Lizenz unterliegt dabei der Person, die die Urheberschaft über die Software innehat. Die Lizenz stellt folglich die Nutzungsbedingungen dar, die von der Software-schaffenden Person explizit eingeräumt werden. Stellt sie ihr Werk unter eine als Open Source-Lizenz anerkannte Lizenz, so erteilt sie anderen Nutzern die oben genannten Rechte. Auch wenn es viele verschiedene Software-Lizenzen für OSS gibt, lassen sich die Lizenzen in zwei unterschiedliche Typen einteilen, nämlich in Copyleft- und Permissive-Lizenzen.³⁹

Historische Einordnung von Open Source Software und Free Software

Die Entstehung der beiden Lizenztypen Copyleft- und Permissive-Software-Lizenz steht im Zusammenhang mit der Debatte um die bessere Begrifflichkeit. Der Begriff „Open Source“ ist eine relativ junge Bezeichnung und geht auf die Gründung der Open Source Initiative (OSI) 1998 zurück. In den 1980er Jahren hatte sich hauptsächlich der Begriff „Free Software“ etabliert.⁴⁰

Grundsätzlich war zu Beginn der Software-Entwicklung jede Software „frei“. In den ersten Jahrzehnten der Software-Entwicklung stellte die Software an sich kein eigenes Geschäftsmodell und keine eigene Ertragsquelle dar. Die kommerzielle Informatik konzentrierte sich primär auf den Vertrieb von Hardware. Software wurde zusammen mit der Hardware ausgegeben und so war es möglich, dass Kundinnen und Kunden die Software weiterentwickeln und auch untereinander austauschen konnten. Erste Abnehmer früher Computersysteme waren vor allem Universitäten im amerikanischen Raum. Ende der 1960er Jahre wurde IBM jedoch durch eine Kartellrechtsklage gezwungen, Software und Hardware zu entkoppeln. Dies hatte eine erhebliche Markttransformation zur Folge,

denn Software wurde nun zu einem separat handelbaren und somit kommerziellen Produkt. Zusätzlich wurde 1981 im amerikanischen Raum auch die Anmeldung von Patenten auf Software möglich. Im Zuge der zunehmenden Kommerzialisierung und Lizenzierung von Software formierte sich eine entsprechende Gegenbewegung. Als Antwort auf proprietäre Lizenzen entwickelte Richard Stallman das freie Betriebssystem GNU. Er wurde damit zum Vorreiter der Open Source-Bewegung, die sich 1985 in der Free Software Foundation (FSF) formierte.⁴¹ Aus der Idee heraus, dass Software frei zugänglich sein sollte, entstand die GNU General Public License (GPL). Unter Version 2 dieser Lizenz steht auch der Anfang der 1990er Jahre von Linus Torvalds initiierte Linux-Kernel, der zusammen mit den GNU-Programmen ein komplett freies Betriebssystem bilden kann.

Der eigentliche Begriff „Open Source Software“ entstand im Zuge der Offenlegung des Webbrowsers Netscape Navigator (heute „Firefox“) Ende der 1990er Jahre. Beim Netscape Navigator handelte es sich um ein proprietäres Produkt, das sich jedoch nicht gegen den Microsoft Internet Explorer durchsetzen konnte. Infolgedessen stellte das Unternehmen Netscape Communications das Projekt mit dem Ziel der erhöhten Diffusion einschließlich des Quellcodes zur freien Verfügung.⁴² Die Bezeichnung von „Free Software“ für ein vormals proprietäres Produkt erschien jedoch in der Kommunikation unpassend, sodass erstmalig der Begriff „Open Source Software“ verwendet wurde.⁴³ Dabei unterscheidet sich OSS lizenzrechtlich nicht von freier Software, denn alle Lizenzen freier Software sind auch als Open Source-Lizenzen genannt. Lediglich die Konnotation ist unterschiedlich: Der Begriff „Open Source Software“ stellt den praktischen Nutzen und die gemeinschaftliche Entwicklungsmethode in den Vordergrund. Die Bezeichnung „Free Software“ wiederum möchte vor allem den gesellschaftlichen Mehrwert betonen.⁴⁴

Open Source-Lizenzen

Richard Stallman entwickelte mit der GNU General Public License den Begriff „Copyleft“ und somit einen speziellen Typus von OSS-Lizenzen. Die Copyleft-Klausel in einer Open Source-Lizenz besagt, dass Änderungen an dem Werk wieder unter dieselbe Lizenz gestellt werden müssen, wie das ursprüngliche Werk.⁴⁵ Damit wollte Richard Stallman vermeiden, dass vormals „freie“ Software „unfrei“ wird.

Alternativ zu Copyleft-Lizenzen existieren sogenannte Permissive-Lizenzen. Ihr Grundprinzip ist die Möglichkeit, abgeleitete oder distribuierte Werke unter eine andere Lizenz zu stellen, die nicht die ursprüngliche Lizenz sein muss. Infolgedessen kann OSS, die unter einer permissiven Lizenz steht, auch unter einer proprietären Lizenz vertrieben werden.⁴⁶

39 In seiner Übersicht listet das Institut für Rechtsfragen der Freien und Open Source Software (IfrOSS) derzeit mehr als 250 verschiedene Open Source-Lizenzen auf, vgl. IfrOSS 2021.

40 Vgl. Bitkom 2016.

41 Vgl. ebd.

42 Vgl. Appleyard/Chesbrough 2017.

43 Vgl. Bitkom 2016.

44 Vgl. West 2003.

45 Vgl. Free Software Foundation 2021.

46 Vgl. Bitkom 2016.

Im Falle einer Copyleft-Lizenz kann zudem zwischen einem starken und einem schwachen Copyleft unterschieden werden. Die Stärke des Copyleft-Effekts beschreibt dabei, inwieweit das veränderte, ursprüngliche und in eine andere Software integrierte Werk Einfluss auf andere Software nehmen kann. Die Ausprägung des Copylefts unterscheidet sich dabei von Lizenz zu Lizenz. Entscheidend aus Praxissicht ist dabei vor allem die Lizenzkompatibilität. So sind permissive Lizenzen untereinander üblicherweise kompatibel. Jedoch ist nicht jede Copyleft-Lizenz mit einer permissiven Lizenz oder anderen Copyleft-Lizenzen nutzbar.⁴⁷ Dies gilt es im Vorfeld zu prüfen, wenn Distribution und Veröffentlichung des eigenen Werks geplant sind. Das Institut für Rechtsfragen der Freien und Open Source Software (IfrOSS) listet mehr als 250 verschiedene Open Source-Lizenzen auf seiner Homepage auf.⁴⁸ Die wohl bekanntesten Copyleft-Lizenzen sind die GNU General Public-Lizenzen mit strengem Copyleft-Effekt. Bekannte Lizenzen mit schwachem beziehungsweise beschränktem Copyleft sind zum Beispiel die Common Development and Distribution-Lizenzen oder die Mozilla Public-Lizenzen. Zu den bekanntesten Permissive-Lizenzen gehören vor allem die Apache 2.0-Lizenz, die MIT-Lizenzen und die BSD-Lizenzen. Dabei ist zu beobachten, dass sich Permissive-Lizenzen größerer Beliebtheit erfreuen als Copyleft-Lizenzen. Standen 2012 lediglich 41 Prozent der bereitgestellten Open Source-Komponenten unter einer Permissive-Lizenz, so sind es 2021 76 Prozent. Die beliebtesten Lizenzen sind mit 28 Prozent die Apache 2.0, gefolgt von der MIT-Lizenz mit 26 Prozent und der GPL v3 mit 10 Prozent.⁴⁹

Neben den formalen Anforderungen hat die Auswahl der Lizenz auch einen unmittelbaren Einfluss auf das Geschäftsmodell und die Möglichkeiten, mit OSS Erlöse zu erzielen.⁵⁰ Ein beliebtes Geschäftsmodell ist die duale Lizenzierung, bei der die Software oder ein Teil davon sowohl unter einer Open Source-Lizenz steht wie auch unter einer proprietären Lizenz. Das bietet Firmen den Anreiz, eine Software, die unter einer Copyleft-Lizenz steht, proprietär zu erwerben, um den Copyleft-Effekt zu vermeiden.⁵¹ Die duale Lizenzierung ist jedoch nur eine Möglichkeit, wie mit OSS ein nachhaltiges Geschäftsmodell realisiert werden kann: Beratung, Hosting und Support, individuelle Anpassungen, Vertrieb, Preisdiskriminierungsstrategien, kostenpflichtige Erweiterungen oder Vollprodukte, Abos, Plattformbeiträge und Spenden von der Community sind andere gängige Einnahmequellen im Open Source-Bereich (siehe Kapitel 4.2.2).⁵² Folglich stiftet Open Source nicht nur einen gesellschaftlichen Mehrwert, sondern schafft für Unternehmen auch wirtschaftliche Mehrwerte und die Möglichkeit neuer Geschäftsmodelle.

Open Source-Anwendungen

Das Geschäftsmodell und die Lizenz geben dabei per se nicht die Nutzungsart der OSS vor. Oftmals handelt es sich bei bereitgestellter

OSS um Programmmodule und weniger um vollständige Programme. OSS lässt sich dabei auf verschiedene Art und Weise nutzen. Generell können Open Source-Komponenten

- in andere Softwarekomponenten eingegliedert werden (Incorporation)
- mit anderen Softwarekomponenten verbunden werden (Linking)
- modifiziert werden (Modification)
- umgewandelt werden (Translation)

Unter Eingliederung (*Incorporation*) versteht man die Möglichkeit, Teile von OSS in eigene Software zu integrieren. Hierzu zählt beispielsweise das Einfügen von Quelltextteilen in den eigenen Programmcode. Beim Verbinden (*Linking*) wird eine Open Source-Komponente mit einer eigenen Komponente verknüpft. Diese Verbindung kann statisch sein, dann ist das Programmmodul Teil eines fertigen Programms, oder dynamisch erfolgen, das bedeutet, die Verbindung zum Programmmodul entsteht jedes Mal neu, während ein Programm ausgeführt wird. Auch durch Kapselung (Packaging) kann ein Verbinden erfolgen. Bei der Modifikation (*Modification*) wird der ursprüngliche Quellcode einer Open Source-Komponente angepasst beziehungsweise verändert. Hierunter fällt auch das Einfügen von eigenen Programmmodulen oder das Entfernen von Quelltextteilen. Solche Anpassungen dienen zum Beispiel der Optimierung der Komponenten oder der Fehlerbehebung. Zu guter Letzt dürfen Open Source-Komponenten umgewandelt (*Translation*) werden. Dies umfasst beispielsweise das Übersetzen in andere Programmiersprachen und das Kompilieren in eine Binärdatei. Zu beachten ist bei den verschiedenen Nutzungsmöglichkeiten die Einhaltung der jeweiligen Lizenzpflichten.

Diese Möglichkeiten, Open Source-Komponenten zu verwenden, führen dazu, dass OSS oftmals nicht nur aus einer einzelnen Komponente besteht. Im Gegenteil ist eine Open Source-Anwendung oft die Komposition mehrerer Komponenten. Manche Open Source-Projekte entwickeln Komponenten, die einzeln betrachtet nicht nutzbar sind, sondern explizit dafür entworfen werden, in Kombination mit anderen Komponenten verwendet zu werden. Beispiele hierfür sind Bibliotheken, deren Module alleinstehend (standalone) nicht verwendet werden können. Ebenso kann Software so modular aufgebaut sein, dass sich einzelne Bestandteile (zum Beispiel die Anbindung an eine Datenbank) möglichst einfach austauschen lassen. Der Quelltext der verschiedenen Bestandteile, die erst durch Kombination eine nutzbare Software ergeben, kann dabei auf verschiedene Archive, sogenannte Source Code Repositories, verteilt sein.

47 Vgl. Bitkom 2016.

48 Vgl. IfrOSS 2021.

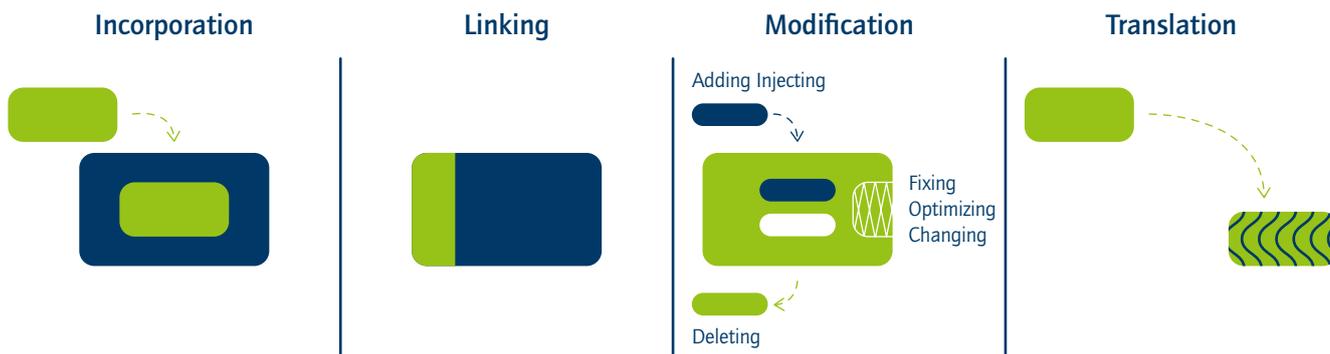
49 Vgl. Johnson 2021.

50 Vgl. Kort/Zaccour 2011, Spijkerman/Jansen 2018.

51 Vgl. ebd.

52 Vgl. Bitkom 2016, Spijkerman/Jansen 2018, Kort/Zaccour 2011.

Abbildung 4: Möglichkeiten der Verwendung von Open Source-Komponenten



Quelle: eigene Darstellung nach OpenChain 2022

4.2. Verbreitung und wirtschaftliche Bedeutung von Open Source Software

OSS ist ein elementarer Bestandteil in der Software-Entwicklung: Viele Anwendungen basieren auf dem offenen Linux-Betriebssystem. Auch abseits Linux-basierter Entwicklungen existieren weitverbreitete Applikationen und Programme.⁵³

- **Web Development**, wie: PHP, Angular, Node.js, Eclipse Che
- **Infrastructure Software**, wie: Xen, KVM, Ceph, OpenStack
- **Big Data**, wie: Apache Hadoop, Apache Spark, Apache Cassandra
- **Mobile**, wie: Android, Apache Cordoba
- **Machine Learning**, wie: TensorFlow, Apache MXNet, PyTorch, Keras
- **DevOps**, wie: Jenkins, Chef, Puppet, Ansible, Terraform, OpenShift, Docker Engine

Folglich ist es nicht verwunderlich, dass die Mehrheit der deutschen Unternehmen auf OSS setzt. In einer Umfrage des Digitalverbands Bitkom aus dem Jahr 2019 geben rund 69 Prozent der befragten Unternehmen an, Open Source einzusetzen.⁵⁴ Bei Großunternehmen mit mehr als 2.000 Beschäftigten sind es sogar 86 Prozent. 58 Prozent der befragten Unternehmen setzen OSS innerhalb des eigenen Betriebs ein, ohne Änderungen am Quellcode vorzunehmen. Weitere 32 Prozent der befragten Unternehmen nehmen Änderungen am Quellcode vor – jedoch nur für unternehmensinterne Zwecke. Besonders beliebt sind dabei die Nutzung von quellcode-offener Container-Technologie (37 Prozent), gefolgt von Big Data und Analytics (34 Prozent), Cloud Computing-Anwendungen (31 Prozent) und IoT-Anwendungen (28 Prozent). Den Abschluss

bilden Software im Bereich des maschinellen Lernens (8 Prozent) und Blockchain-Anwendungen (1 Prozent). Die jährlich durchgeführte Umfrage „The State of Enterprise Software“ des Unternehmens Red Hat ergänzt das oben gezeichnete Bild. Im Jahr 2020 wurden 1.250 IT-Unternehmen, davon 450 aus dem europäischen Raum, zu ihrer Nutzung von OSS befragt. Nach den Ergebnissen wird vor allem die Bedeutung der OSS im Bereich Edge Computing und maschinelles Lernen steigen.⁵⁵

Der sogenannte Hype Cycle des Marktforschungsunternehmens Gartner Inc. für den Bereich der Open Source-Entwicklung im Jahr 2020 zeigt zudem, dass vor allem „Artificial General Intelligence“, „Augmented Intelligence“, „Swarming Robotics“, „AI Augmented Development“ und „AI Developer and Teaching Kits“ aktuelle Trends sind. Folglich bestätigt der Hype Cycle, dass vor allem Anwendungen und Entwicklungen im Bereich der KI große Beachtung finden.⁵⁶ Ebendiese Bereiche werden gleichzeitig als Themen mit einem besonders hohen Sprunginnovationspotenzial identifiziert und bedürfen allein deshalb einer besonderen Betrachtung.

Als Vorteile von OSS sehen die befragten Unternehmen der Red Hat-Umfrage vor allem den Zugang zu neuen Innovationen (82 Prozent) wie auch die Möglichkeit der Flexibilisierung und Anpassung der Software auf die eigenen Unternehmensbedürfnisse (81 Prozent).⁵⁷ Die Befragten der oben erwähnten Bitkom-Studie sehen die Vorteile von Open Source in der Kosteneinsparung durch die Vermeidung von Lizenzkosten, der hohen IT-Sicherheit, die durch regelmäßige Updates und hohe Stabilität der Software gewährleistet wird, aber auch in der Offenheit und einfachen Anpassung der Software an die unternehmensinternen Prozesse. Der Aspekt der Kooperation und gemeinsamen Innovationsgestaltung wird auch genannt, jedoch lediglich von gut 8 Prozent der befragten Unternehmen als Vorteil anerkannt. Damit einher geht auch die Beobachtung, dass wesentlich mehr Unternehmen Open Source einsetzen als selbst aktiv mitentwickeln oder zur Verfügung stellen.⁵⁸

53 Vgl. Pimcore 2022.

54 Vgl. Bitkom 2019.

55 Vgl. Red Hat 2021.

56 Vgl. Gartner 2020b.

57 Vgl. Red Hat 2021.

58 Vgl. Bitkom 2019.

Dies spiegelt sich auch in der vergleichsweise moderaten Beteiligungsaktivität von Unternehmen an Open Source-Projekten wider. Lediglich 31 Prozent der mehr als 800 Befragten Unternehmen geben in der Bitkom-Studie an, sich an Open Source-Projekten zu beteiligen, und nur 2 Prozent entwickeln eigenständig OSS.⁵⁹ Damit einher geht auch der Befund eines auffälligen strategischen Mangels: 77 Prozent der befragten Unternehmen geben an, über keinerlei Open Source-Strategie für das eigene Unternehmen zu verfügen. Lediglich 21 Prozent der Unternehmen bestätigen, eine solche Strategie zu haben. Davon verfügen 12 Prozent über eine Strategie zur Verwendung von OSS, 7 Prozent über eine Strategie zur Beteiligung und Verwendung und 2 Prozent über eine Strategie zur Beteiligung an Open Source-Projekten.⁶⁰ Daraus lässt sich schlussfolgern, dass der Großteil der befragten Unternehmen die Potenziale, die mit einer eigenen Bereitstellung von OSS einhergehen, noch nicht vollends erschlossen hat.

Als Hemmnisse beziehungsweise Nachteile bei der Verwendung von OSS geben die befragten Unternehmen der Bitkom-Studie vor allem fehlende Fachkräfte (12 Prozent), unklare Gewährleistungssituationen (6 Prozent) oder die aufwendige Umstellung von proprietärer Software auf OSS (6 Prozent) an. Eine Befragung des Marktforschungsunternehmens Gartner von 74 Expertinnen und Experten zeigt zudem auf, dass die Verwendung von OSS vor allem gehemmt wird durch Bedenken hinsichtlich der Nachhaltigkeit des Projekts beziehungsweise der Software (23 Prozent), Sicherheitsbedenken (28 Prozent), Entscheidungsschwierigkeiten bezüglich kommerziellen Supports (20 Prozent) wie auch Bedenken hinsichtlich der Lizenz (12 Prozent).⁶¹

Die Überwindung solcher Hemmnisse auf der mikroökonomischen Seite kann jedoch zu positiven makroökonomischen Effekten führen.⁶² So analysieren Blind et al. mithilfe ökonomischer Verfahren auf Basis eines umfangreichen Panel-Datensatzes die gesamtwirtschaftliche Wirkung von Open Source: Das BIP der einzelnen Mitgliedsstaaten der Europäischen Union profitiert signifikant von OSS. Auch unter Berücksichtigung von Kontrollvariablen wie Forschungsausgaben und Patenten bleibt der Effekt signifikant. Die Anzahl der nationalen Kontributionen ist dagegen kein signifikanter Treiber für das nationale BIP. Die Autorinnen und Autoren argumentieren, dass quelloffene Software als öffentliches Gut angesehen werden kann.⁶³ Das sogenannte „Free Riding“, das heißt die Nutzung durch viele, ohne eigene Beiträge zu leisten, ist ein damit verbundenes Problem, wie auch die Ergebnisse der weiter oben erwähnten Studien zu Beteiligungsaktivitäten zeigen.⁶⁴ Die Open Source-Beiträge der europäischen Community schätzen Blind et al. auf einen Beitrag von gut 60 bis 95 Milliarden Euro

zum europäischen BIP im Jahr 2018. Zudem hat die Kontribution zu OSS-Projekten einen positiven Einfluss auf die Arbeitsproduktivität. Weiter besteht ein positiver Zusammenhang zwischen Open Source-Kontributionen und der nationalen Wettbewerbsfähigkeit (gemessen in Exporten und Trade in Value Added). Einen signifikanten beziehungsweise direkten Einfluss von OSS auf die Innovationskraft konnten die Studiersteller zwar nicht erkennen – jedoch zeigt sich, dass OSS und Patente im Bereich der Software- und Computertechnologie als komplementär zu betrachten sind: Ein Anstieg an außereuropäischen Kontributionen führt zu einer Steigung der europäischen Patente.⁶⁵ Auch wenn der Einfluss von Open Source auf die Innovationskraft nicht signifikant oder nur marginal ist, so ist der Effekt von OSS auf Start-ups höchst signifikant. Die Studie kommt zum Schluss, dass eine 10-prozentige Erhöhung der Kontributionen durch EU-Mitgliedsstaaten zur Gründung von gut 650 neuen Start-ups im Bereich der Informationstechnologie führen könnte.

4.2.1. Wirtschaftliche Bedeutung von Open Source für KMU sowie Start-ups

Der Einfluss von OSS auf Neugründungen von Start-ups kann vor allem in der kostenfreien Ressource von Softwarekomponenten liegen. Start-ups können sich an OSS-Komponenten bedienen, darauf basierend eigene Produkte entwickeln und neue Geschäftsmodelle realisieren. Nach der Bitkom-Studie setzen rund 65 Prozent der befragten Unternehmen in der Größenklasse 100 bis 199 Beschäftigte und 71 Prozent in der Kategorie 200 bis 499 Beschäftigte OSS ein. Unabhängig vom Einsatz konnte die Studie zeigen, dass kleine und mittlere Unternehmen (KMU) interessierter an OSS sind als Großkonzerne.⁶⁶ Auffällig ist jedoch, dass Großkonzerne häufiger eine Open Source-Strategie verfolgen als KMU, obwohl diese OSS eine höhere Bedeutung beimessen. Insbesondere die Erlösmöglichkeiten, die mit OSS verbunden sind, aber auch der Zugang zu neuen Marktsegmenten sind für KMU relevant. Aus diesem Grund stellen KMU neben Mikroorganisationen die größte Gruppe der Kontributoren auf GitHub, einem der größten Versionsverwaltungssysteme für Software-Entwicklungsprojekte weltweit.⁶⁷ Im Hinblick auf Industrie 4.0-Anwendungen kann OSS die Einstiegshürden für KMU senken. Neue Technologien können risikoarm beziehungsweise ohne großen finanziellen Aufwand erprobt werden. KMU können auf der Basis von OSS neue, digitale Dienste und Services für ihre Kunden leichter realisieren. Großkonzerne bewerten hingegen die Kostenersparnis als größeren Mehrwert.⁶⁸ Insgesamt ergeben sich für Großkonzerne absolut gesehen höhere Kosten für die Anschaffung proprietärer Software als für KMU, sodass die Ersparnis durch OSS hier besonders relevant erscheint.

59 Vgl. Bitkom 2019.

60 Vgl. ebd.

61 Vgl. Gartner 2019.

62 Vgl. Blind et al. 2021.

63 Öffentliche Güter zeichnen sich im Wesentlichen dadurch aus, dass sie durch Konsum nicht geringer werden und niemand von einer Nutzung ausgeschlossen werden kann.

64 Vgl. Bitkom 2019.

65 Vgl. Blind et al. 2021.

66 Vgl. Bitkom 2019.

67 Vgl. Blind et al. 2021.

68 Vgl. ebd.

Eine Expertise des Forschungsbeirats aus dem Jahr 2019 zeigt, dass KMU vor allem die digitale vertikale Integration anstreben und sich Industrie 4.0-Projekten gegenüber aufgeschlossen zeigen.⁶⁹ Dabei stehen KMU vor allem vor der Herausforderung, dem Fachkräftemangel zu begegnen und qualifiziertes Personal zu gewinnen, das benötigte Digitalisierungskennnisse einbringen kann. Zudem nennen KMU den Mangel an Standards im Bereich der Schnittstellen als ein großes Hemmnis. Darüber hinaus existieren vor allem Schwierigkeiten und Herausforderungen im Bereich der cyber-physischen Systeme und der cyber-physischen Produktionssysteme. Auch hier fehlen Know-how und Ressourcen, um IT-Sicherheit gewährleisten zu können.⁷⁰ In diesen Bereichen kann der Zugang zu OSS einen Beitrag leisten. Zwar bedarf es gewisser Kenntnisse im Bereich der Software-Anwendung, aber gleichzeitig können KMU mit der Verwendung von OSS auf bereits etablierte und frei verfügbare Formate zurückgreifen und entsprechende Entwicklungsressourcen minimieren. Ebenso kann OSS einen Beitrag zur Standardisierung innerhalb von Industrie 4.0 leisten (siehe Abschnitt 4.2.3).

4.2.2. Geschäftsmodelle und kommerzielle Nutzung von Open Source Software

Die wirtschaftlichen Potenziale von Open Source lassen sich zum einen auf indirekte Effekte (siehe Abschnitt 4.2.3) zurückführen, aber auch auf Geschäftsmodelle, die mithilfe von Open Source realisiert werden können. Die häufig anzutreffende Auffassung, dass Open Source zwar als Kollaborationstool Vorteile mit sich bringe oder ein positiver Kostenfaktor sei, aber mit OSS kein Geld zu verdienen sei, lässt sich durch eine Vielzahl an Open Source-Geschäftsmodellen widerlegen. Bekannte Geschäftsmodellmuster sind zum Beispiel duale Lizenzierung, Services (wie Support, Wartung, Hosting, Entwicklung, Beratung und Schulung), Subskription und Verbundangebote.⁷¹

Das Geschäftsmodell der *dualen Lizenzierung* setzt auf eine Mehrfachlizenzierung: Die Software steht sowohl unter einer Open Source-Lizenz als auch einer proprietären Lizenz. Das Geschäftsmodell der dualen Lizenzierung erfolgt häufig unter der Verwendung einer Copyleft-Lizenz, bietet jedoch außerdem Unternehmen und Interessierten, die sich nicht an die Auflage des Copyleft halten möchten, die Möglichkeit, eine proprietäre Lizenz zu erwerben. Die proprietäre Lizenz kann sich dabei gänzlich oder nur teilweise auf die unter der Open Source-Lizenz stehende Software beziehen. Möglich ist auch, Kernfunktionen und Basisanwendungen quelloffen bereitzustellen und spezielle Erweiterungen oder Features unter eine proprietäre Lizenz zu stellen. Bekannte Lösungen, die dem Prinzip der dualen Lizenzierung folgen, sind zum Beispiel MySQL oder Asterisk.

Neben der Ausgabe unterschiedlicher Lizenzen können Unternehmen sich jeglichen *Service* oder komplementären Dienst rund um die Software vergüten lassen. Gängige Service-Modelle umfassen

den Support, die Wartung und das Hosting von Software, darüber hinaus auch unternehmensindividuelle Entwicklung und Anpassung sowie Beratungs- und Schulungsangebote. Dieses Geschäftsmodell wird beispielsweise von Red Hat verfolgt, das unter anderem mit der Red Hat Enterprise Linux Version eine speziell auf Unternehmen abgestimmte Linux-Distribution vertreibt. In Verbindung damit bietet Red Hat weitere OSS-Lösungen zum Beispiel im Bereich des Cloud Computings an. Flankiert wird das Angebot durch Support, Consulting und Schulungen.

Gleichzeitig bietet Red Hat auch das gängige Open Source-Geschäftsmodell des Abonnements an. Hierbei werden bestimmte Services, zum Beispiel der Support oder die Wartung der Software, an einen bestimmten Zeitraum geknüpft. Da durch die freie Verfügbarkeit der Software keine Abhängigkeit von Herstellern eintritt (Vendor Lock-In), müssen sich Open Source-Subskriptionsmodelle vor allem durch die Qualität des Services auszeichnen. Red Hats Unternehmenserfolg baut primär auf der communitygetriebenen Open Source-Entwicklung auf, deshalb muss das Unternehmen auch ein aktives Community Management betreiben: Als Sponsor des Fedora-Projekts unterstützt Red Hat die Open Source Community bei der Weiterentwicklung der Linux-Distribution, basierend auf dem freien Paketverwaltungssystem, das ursprünglich von Red Hat entwickelt worden war. So gibt Red Hat einen Teil der Erlöse, die es auf Basis von community-getriebener Open Source-Entwicklungen generiert, an die entsprechende Community zurück und sichert so den Fortbestand seiner Entwicklungsgrundlage.⁷²

Auch rund um die Entwicklung und Entstehung von OSS kann ein Geschäftsmodell aufgebaut werden. *Verbundprojekte* oder *Mediatoren* können unterschiedliche Akteure im Sinne einer gemeinsamen Plattform zusammenführen.

Neben solchen direkten Open Source-Geschäftsmodellmustern dient das Angebot von OSS oder zumindest die kostenfreie Zurverfügungstellung von Software Development Kits und Application Programming Interfaces (API) auch als strategisches Element zur Realisierung des eigentlichen Geschäftsmodells. Insbesondere große Plattformen im B2C- oder C2C (Consumer-to-Consumer)-Bereich nutzen OSS oder die Bereitstellung von Programmierschnittstellen zur Integration von komplementären Anbietern in ihrem Ökosystem. Durch die Bereitstellung von offenen Programmierschnittstellen oder gar ganzer Betriebssysteme wie Android können indirekte Netzwerkeffekte gestärkt werden, indem komplementäre Anbieter (zum Beispiel von Apps) ihre Services auf Basis des freien Angebots bereitstellen. Auch im Bereich der Hardware- und Software-Kompatibilität können lukrative Geschäftsmodelle realisiert werden. So kann zum Beispiel die Software unter einer Open Source-Lizenz stehen, aber die entsprechende Hardware wiederum muss kostenpflichtig erworben werden und vice versa.⁷³

69 Vgl. Forschungsbeirat/acatech 2019a.

70 ebd.

71 Vgl. Koenig 2004, Spijkerman/Jansen 2018, Müller et al. 2019.

72 Vgl. Munga et al. 2009.

73 Vgl. Blind et al. 2021.

Abbildung 5: Basisfunktionen besitzen eine hohe Open Source-Eignung



Quelle: eigene Darstellung

Oft lässt sich feststellen, dass OSS eine flankierende und das Geschäftsmodell unterstützende Funktion einnimmt. Die bereitgestellte OSS ist nicht zwingend marktdifferenzierend beziehungsweise betrifft nicht unmittelbar das geistige Kerneigentum des Unternehmens (beispielsweise Hardware). OSS dient vielmehr dazu, die strategischen Ziele zu erreichen, und ist eine flankierende beziehungsweise komplementierende Leistung zum eigentlichen Kerngeschäft, das beispielsweise in der Beratung, in der Konstruktion von Anlagen und Maschinen, im Betreiben einer digitalen Plattform mit Einnahmen über Drittanbieter, oder in der Konzeption individueller Softwareanwendungen liegt.

Die Entwicklung und Bereitstellung von OSS hat daher wenig mit einem altruistischen Kollaborationsmodell gemein, sondern dient ganz und gar kommerziellen Interessen. Es wird deutlich: OSS liefert die Grundlage für erfolgreiche und nachhaltige Geschäftsmodelle. Das steht in keiner Weise im Widerspruch zu einer community-getriebenen Entwicklung. Unternehmen wie Microsoft und Google fördern daher verschiedene Open Source Communities explizit mit finanziellen, aber auch mit personellen Ressourcen. Blind et al. stellen fest, dass von den 1.151 europäischen Unternehmen der Stichprobe aus 14 verschiedenen Branchen vor allem Firmen aus dem IT-Bereich mit einem Anteil von gut 77 Prozent durch eigene Beiträge aktiv zur OSS-Entwicklung auf GitHub beitragen, gefolgt von Forschungseinrichtungen und öffentlichen Institutionen mit 7 Prozent. Unternehmen aus dem Maschinenbau tragen lediglich mit 1 Prozent aktiv zur Open Source-Entwicklung bei.⁷⁴

4.2.3. Indirekte Mehrwerte von Open Source Software: Open Innovation und Standardisierung

Indirekte Mehrwerte von OSS ergeben sich zum einen aus der Kollaborationsmethode, die als Form der Open Innovation zu verstehen ist, zum anderen trägt OSS dazu bei, De-jure- wie auch De-facto-Standards zu etablieren. Unternehmen können folglich nicht nur Geschäftsmodelle mithilfe von OSS realisieren, sondern auch von solchen indirekten Mehrwerten profitieren.

Mithilfe eines Open Source-Lizenzmodells können Unternehmen in der Zusammenarbeit mit Partnern oder Kundinnen und Kunden aufwendige Geheimhaltungsvereinbarungen oder Lizenzfragen umgehen. Durch die Bereitstellung einer Open Source-Lizenz können Stakeholder gemeinschaftlich im Bereich der Software-Entwicklung arbeiten und somit einen offenen und gemeinsamen Co-Creation-Prozess forcieren im Sinne von Open Innovation, das heißt der gezielten Öffnung des Innovationsprozesses durch Unternehmen. Ziel ist dabei, das strategische Innovationspotenzial durch Know-how und Mitarbeit der Außenwelt zu erhöhen.⁷⁵ Die gemeinschaftliche Entwicklung und Arbeit an Software-Projekten verschafft Unternehmen den Vorteil verbesserter Software. Externe Entwicklerinnen und Entwickler können nicht nur Fehler beheben, sondern auch neue Features hinzufügen und damit Anwenderinnen und Anwendern neue Nutzungspotenziale eröffnen. Dieser Ansatz kontrastiert die klassische Unternehmensstrategie der vertikalen Integration beziehungsweise die proprietäre, das heißt patentrechtlich geschützte Entwicklung von geistigem Eigentum, zum Beispiel in Form proprietärer Datenmodelle und Schnittstellen.

In einer immer komplexeren digitalen und vernetzten Welt verfügen Unternehmen oftmals nicht über ausreichend Wissen und Ressourcen, um losgelöst Durchbrüche in ihrer jeweiligen Branche oder Nische zu erzielen. Daher ist das Teilen von Wissen und die Kooperation im Sinne von Open Innovation-Prozessen ein beliebtes Mittel, externes Know-how zu gewinnen. Auch die klassische Software-Entwicklung kann von den offenen „Denk“- und Arbeitsweisen der Open Source-Projekte profitieren. Unternehmen teilen sich hier Entwicklungsressourcen und profitieren mitunter von einer diversen, internationalen Entwicklungs-Community.⁷⁶

Eine Open Source-Strategie zu implementieren, dient nicht nur der Entwicklung der im Unternehmen benötigten Software, sondern kann darüber hinaus die gesamte Branche stärken. Zum einen erlaubt die Bereitstellung von OSS eine verbesserte und branchenübergreifende Zusammenarbeit, die bis hin zur Schaffung von Komplementärgütern (Software und Hardware), die gemeinsam

74 Vgl. Blind et al. 2021.

75 Vgl. Chesbrough 2010.

76 Vgl. West/Gallagher 2006.

nachgefragt werden und sich ergänzen, reichen kann. Zum anderen können neue Produktvarianten in großer Vielfalt durch die formierten Ökosysteme entstehen und aufrechterhalten werden.⁷⁷ Zudem ist die Bereitstellung von OSS oftmals ein entscheidender Aspekt im Sinne der Nachhaltigkeit: Produkte, die eine lange Lebensdauer haben, profitieren von einem offenen Quellcode, denn so können notwendige Anpassungen unabhängig von Beschäftigten und Unternehmen sichergestellt werden. Je mehr Unternehmen eine bestimmte Software, beispielsweise im Bereich der Maschinensteuerung, gemeinsam in ihrem Ökosystem nutzen, zum Beispiel weil sie komplementäre Anbieter sind oder weil die Software kostenfrei zur Verfügung steht, umso eher können sich Standards etablieren.

Die strategische Entscheidung von Unternehmen, OSS bereitzustellen, kann somit eng mit der Bemühung um Standards zusammenhängen. Dabei ist zwischen De-facto-Standards und De-jure-Standards zu unterscheiden. De-facto-Standards oder auch Industriestandards werden im Gegensatz zu einem De-jure-Standard nicht formal durch ein Standardisierungsgremium verabschiedet, sondern entstehen durch die Herausbildung einer praktikablen beziehungsweise sich durchsetzenden Lösung. Zu beobachten ist dabei, dass die Beweggründe, Open Source Communities aktiv beizutreten, oftmals den Beweggründen gleichen, die Unternehmen anführen, wenn sie Standardisierungsorganisationen zur Etablierung von De-jure-Standards beitreten.⁷⁸ Sowohl der Teilnahme an Open Source-Ökosystemen wie auch dem Engagement in Standardisierungsorganisationen liegt das Bedürfnis nach einheitlichen Prozessen und kollaborativ erarbeiteten Lösungsansätzen in der jeweiligen Branche zugrunde. Was die Stakeholdergruppen voneinander unterscheidet, ist, dass in Standardisierungsorganisationen mehrheitlich größere Unternehmen mit starken Patentaktivitäten vertreten sind. Unternehmen, die sich in Standardisierungsorganisationen engagieren, sind weitgehend homogen, wohingegen sich Open Source Communities vor allem durch ihre Heterogenität und Diversität auszeichnen und eher kleinere Unternehmen mit geringerer Patentedichte anziehen. Es gibt jedoch auch viele große Unternehmen, die sowohl in Standardisierungsorganisationen wie in Open Source Communities aktiv sind. Die Entwicklung von De-jure- und De-facto-Standards stehen nicht in Konkurrenz, sondern ergänzen sich: Durch verabschiedete Standards werden vor allem Spezifikationen abgebildet, wohingegen die Entwicklung von De-facto-Standards Anwendungsbezüge herstellt. Oftmals können bereits etablierte, community-getriebene Lösungen als Vorlage für einen neuen Standard dienen, oder Standards und Lösungen werden parallel entwickelt und angeglichen. Insofern sind die Beziehungen zwischen Standardisierungsorganisationen und Open Source Communities multidirektional.⁷⁹

Auch die gemeinschaftliche Entwicklung von Software unter Berücksichtigung des Anwendungsbezugs kann zu einer De-facto-Standardisierung führen. Getrieben werden De-facto-Standards häufig durch den Wettbewerb verschiedener Unternehmen. Verschiedene Lösungen werden entwickelt und erleben eine Marktdiffusion.⁸⁰ Entscheidend für den Erfolg und damit auch für die Etablierung eines De-facto-Standards ist die Akzeptanz einer Lösung am Markt. Eine Software unter eine Open Source-Lizenz zu stellen, kann als strategisches Mittel zur schnelleren Marktdurchdringung betrachtet werden. Unternehmen, die ihre Lösung als Open Source bereitstellen, hoffen auf eine höhere Akzeptanz bei Nutzerinnen und Nutzern, eine schnellere Adoption und somit auch auf eine eventuelle Standardisierung. Die Möglichkeit, die Software proprietär zu lizenzieren und entsprechende Einnahmen zu generieren, entfällt. Jedoch können Unternehmen anderweitig von der Marktdiffusion profitieren, indem zum Beispiel eigene Komplementärgüter verstärkt nachgefragt werden. Neben diesem wettbewerbsorientierten Ansatz können De-facto-Standards auch bewusst auf Basis von gemeinsamen, nicht wettbewerbsgetriebenen Bemühungen entwickelt und etabliert werden. Dies wäre insbesondere im Bereich von Basisfunktionen und Standardtechnologien mit geringem Marktdifferenzierungspotenzial, beispielsweise für IoT oder die intelligente Produktion, sinnvoll. Hier bieten sich neben der Software vor allem Anwendungen aus dem Hardware-Bereich an.⁸¹

Zusammenfassend lässt sich feststellen, dass die Bereitstellung von Code und das Engagement in Open Source Communities nicht nur direkte Mehrwerte im Sinne eines Geschäftsmodells für Unternehmen bieten, sondern vor allem indirekte, sich gegenseitig bedingende Mehrwerte für Unternehmen schaffen. Durch die gemeinschaftliche Entwicklung von Code können Unternehmen Ressourcen einsparen, Kenntnisse und Wissen bündeln oder vom Know-how externer Entwicklerinnen und Entwickler profitieren und somit ihren Innovationsprozess beschleunigen. Zudem bietet die Bereitstellung und Mitwirkung an Open Source Software die Möglichkeit, bestimmte Standardisierungsbestrebungen zu stärken und die Mehrwerte für die entsprechende Branche zu festigen. In Abhängigkeit von der strategischen Ausrichtung und vom strategischen Ziel müssen Unternehmen nicht nur die Open Source-Lösung an sich unter technischen Aspekten bewerten, sondern auch die Gestaltung und Governance des entsprechenden Open Source-Ökosystems berücksichtigen. Die Governance eines Open Source-Projekts, das Management der Community und ihr Selbstverständnis sind entscheidende Faktoren für den Erfolg einer Open Source-Lösung. Diesen und weiteren Rahmenbedingungen widmet sich das folgende Kapitel.

77 Vgl. West/Gallagher 2006.

78 Vgl. Blind/Böhm 2019.

79 Vgl. ebd.

80 Vgl. Wiegmann et al. 2017.

81 Vgl. Blind et al. 2021.

5. Open Source-Ökosysteme und Open Source-Strukturen

Das allgemeine Verständnis von Open Source hat sich in den vergangenen Jahren stark gewandelt, ausgehend von einem noch zu Beginn der 2000er Jahre vorherrschenden Verständnis einer „commons-based peer production“ beziehungsweise einer „voluntary culture“ mit „self-governing groups of individuals“ hin zu einer stärker industriegetriebenen Form kontinuierlicher Forschungs- und Entwicklungskooperation.⁸² So ist auch zu beobachten, dass Entwicklerinnen und Entwickler zunehmend von Unternehmen speziell zum Beitragen an Open Source-Projekten angestellt werden oder ihre Tätigkeiten im Open Source-Umfeld als Bestandteil ihrer Jobs bezahlt werden.⁸³ Diese Entwicklung kann als Ausdruck einer zunehmenden Akzeptanz wie der zunehmenden Bedeutung von Open Source im Allgemeinen und OSS im Speziellen verstanden werden.

Bevor die Situation von Open Source in Industrie 4.0 analysiert wird, werden zunächst grundlegende und für die angestrebten Einschätzungen und Handlungsempfehlungen zu berücksichtigende Rahmenbedingungen erläutert. Dies geschieht in drei thematischen Abschnitten. Zunächst erfolgt eine Darstellung der gängigen Organisationsformen und Governance-Modelle von Open Source-Projekten. Hieran schließt sich eine Betrachtung der Einbettung beziehungsweise Integration von Open Source-Aktivitäten in Unternehmen an. Schließlich soll erörtert werden, anhand welcher Messgrößen Open Source-Projekte beziehungsweise -Ökosysteme bewertet werden können.

5.1. Organisationsstrukturen von und Prozesse in Open Source-Projekten

Zum besseren Verständnis der Organisationsstrukturen und Prozesse beim Management von Open Source-Projekten ist es von Bedeutung, den Blick auf die unterschiedlichen Motivations- beziehungsweise Interessenslagen der Beteiligten zu lenken. Hierzu können Stakeholder und Open Source-Beteiligte in *Einzelpersonen* und *Unternehmen* unterschieden werden⁸⁴.

Motivationslagen von Einzelpersonen und Unternehmen

Es existieren verschiedene Erhebungen und Veröffentlichungen, die sich damit beschäftigen, warum eine Beteiligung von *Einzelpersonen* an OSS-Projekten erfolgt. Es lassen sich intrinsische

und Community-bezogene Motivationen, wie eine grundsätzliche Zustimmung zum Open Source-Ansatz, angestrebte Reputation, wechselseitiger Nutzen, Lernen und Eigennutzen, von den rein extrinsischen Treibern Karriere und Bezahlung unterscheiden.⁸⁵ Ergänzend hierzu unterscheiden Bagozzi und Dholakia kognitive (Einstellungen, wahrgenommene Verhaltenskontrolle, Identifikation mit der Open Source-Bewegung), affektive (positive und negative antizipierte Emotionen) und soziale (soziale Identität) Determinanten für die Teilnahme an Open Source-Projekten.⁸⁶

Der im Auftrag der Linux Foundation durchgeführte „2020 FOSS Contributor Survey“ stellt fest, dass bei Einzelpersonen nach wie vor intrinsische, nicht monetäre Gründe für die Mitarbeit an Open Source-Projekten dominieren: Das Vornehmen einer Korrektur oder Hinzufügen einer benötigten Funktion, Spaß am Lernen und Erfüllung eines Bedürfnisses nach kreativer und unterhaltsamer Arbeit. Eine extrinsische Motivation durch Bezahlung hingegen spielt nur eine untergeordnete Rolle. Insgesamt gilt, dass sich die Motivationen im Laufe der Zeit ändern und langfristige und kurzfristige Motivatoren unterschieden werden können.⁸⁷

In oben genannter Studie wurden Kontextfaktoren, zum Beispiel die Governance, Glaubwürdigkeit und Offenheit der Sponsor-Unternehmen oder die Qualität des Projektmanagements nicht untersucht, wenngleich andere Studien einen starken Einfluss solcher Faktoren auf die Motivation nachgewiesen haben.⁸⁸

Auch die Gründe von *Unternehmen* zur Beteiligung an OSS-Projekten sind vielschichtig.⁸⁹ Im Sinne einer Open Innovation nutzen sie OSS-Gemeinschaften, um zum Beispiel Standardtechnologien (engl.: commodity technologies oder commodities) bereitzustellen, informellen Wissensaustausch zu betreiben und potenzielle Verbesserungen an bestehenden Produkten zu entwickeln.⁹⁰ Dabei durchlaufen sie häufig vier Phasen, die von West und Bogers im Kontext der Open Innovation als „Beschaffung“, „Integration“, „Kommerzialisierung“ und „Interaktion“ definiert worden sind.⁹¹ Eine Übertragung dieser vier Phasen auf den speziellen Fall von OSS-Projekten liefert Abbildung 6. In den ersten beiden Phasen „Konsum“ und „(An-)Teilnahme“ agieren Unternehmen im Wesentlichen als Konsumenten beziehungsweise Nutzer von OSS. Erst in den letzten beiden Phasen erfolgt eine Beteiligung der Beschäftigten an OSS oder die direkte finanzielle Unterstützung von Projekten – verbunden mit den Zielen, Zugang, Legitimität und Einfluss zu gewinnen.

82 Vgl. Blind et al. 2021, Lerner/Tirole 2002, siehe Benkler 2002, S. 1.

83 Vgl. Nagle et al. 2020.

84 Vgl. Blind et al. 2021.

85 Vgl. Shah/Nagle 2020, Nagle et al. 2020.

86 Vgl. Bagozzi/Dholakia 2006.

87 Vgl. Nagle et al. 2020, Shah 2006.

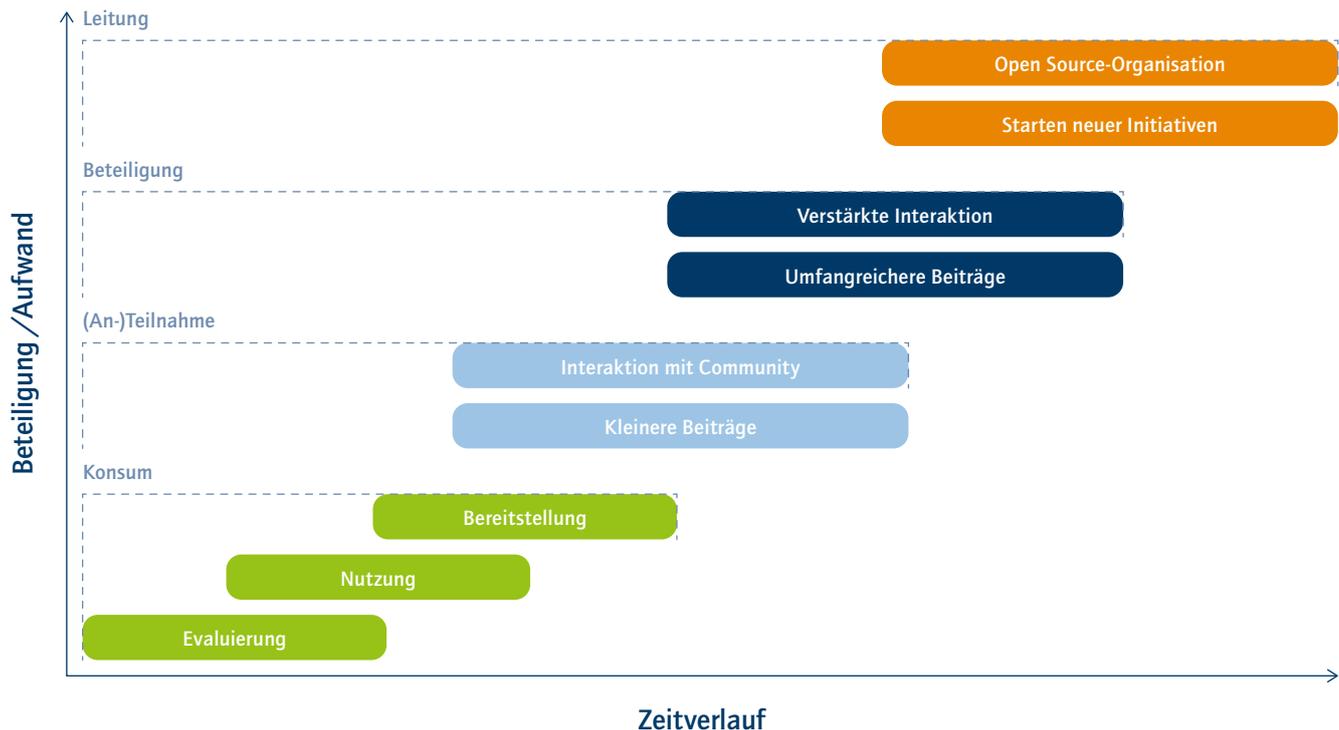
88 Vgl. Shah/Nagle 2020, Iskoujina/Roberts 2015, Spaeth et al. 2015.

89 Vgl. Blind et al. 2021.

90 Vgl. Dahlander/Magnusson 2008, West 2003, Henkel 2006.

91 Vgl. Dahlander/Wallin 2006, Henkel 2006, West/Bogers 2014.

Abbildung 6: Im Zeitverlauf ansteigende Open Source-Beteiligung von Organisationen



Quelle: eigene Darstellung in Anlehnung an Aniszcyk 2021

Hinsichtlich der Offenlegung von Quellcode oder der Beteiligung an OSS verfolgen Unternehmen in Abhängigkeit von ihren Zielen verschiedene Ansätze der Zugänglichkeit und Transparenz.⁹² Dabei gilt vereinfacht der folgende Zusammenhang: Starkes Wachstum wird von einer offenen Strategie begünstigt, während proprietäre Strategien den Unternehmen die Kontrolle erleichtern. Die Wahl für eine bestimmte Strategie kann im Laufe des Produktlebenszyklus und in Abhängigkeit von den Unternehmenskompetenzen variieren.⁹³

Nach West und Gallagher können Unternehmen in verschiedener Hinsicht von einer Offenlegung profitieren:⁹⁴

- Bündelung von Aufwand und Risiko im Bereich Forschung und Entwicklung
- Auslagerung von noch nicht kommerzialisierten Technologien oder Lösungen, die das Potenzial haben De-facto-Standards zu etablieren, um Verbesserungen an der Lösung, komplementäre Produkte oder Funktionalitäten anzustoßen oder laufende Kosten für Entwicklung und Support zu verringern
- Monetarisierung von Komplementärgütern auf der Basis eines offenen Kerns

Open Source Communities

Entscheidend für Erfolg und Misserfolg jeder OSS sind Communities. Das grundlegende Konstrukt von Open Source Communities geht auf die Anfänge der Computerentwicklung zurück und hat seine Wurzeln in der akademischen und unternehmerischen Sharing- und Hacker-Mentalität. Fast alle Open Source Communities weisen die folgenden Hauptmerkmale auf: geografische Verteilung der Entwicklungsressourcen, dezentralisierte Entscheidungsmöglichkeiten, Transparenz sowohl bei der Entwicklung als auch bei der Entscheidungsfindung, Einfluss in der Gemeinschaft wird durch ständige, wertvolle Beiträge verdient (Meritokratie). Darüber hinaus weisen die meisten Gemeinschaften eine straffe vertikale Hierarchie mit einer lockeren horizontalen Struktur auf, die es ermöglicht, dass kleine Änderungen nach oben fließen und viele schnelle Überprüfungszyklen durchlaufen. Die gegenseitige Kontrolle der Kontributorinnen und Kontributoren und das Gleichgewicht der Macht zwischen ihnen tragen zur Sicherung der Softwarequalität bei.⁹⁵

Wenngleich OSS-Projekte in ihrer Organisation und Struktur ganz grundlegend voneinander abweichen können, so lassen sich dennoch zwei unterschiedliche Typen von Communities identifizieren, die sich gegenseitig bedingen, teilweise überschneiden und idealerweise um jedes Projekt herum entwickeln.⁹⁶

92 Vgl. West/O'mahony 2008.

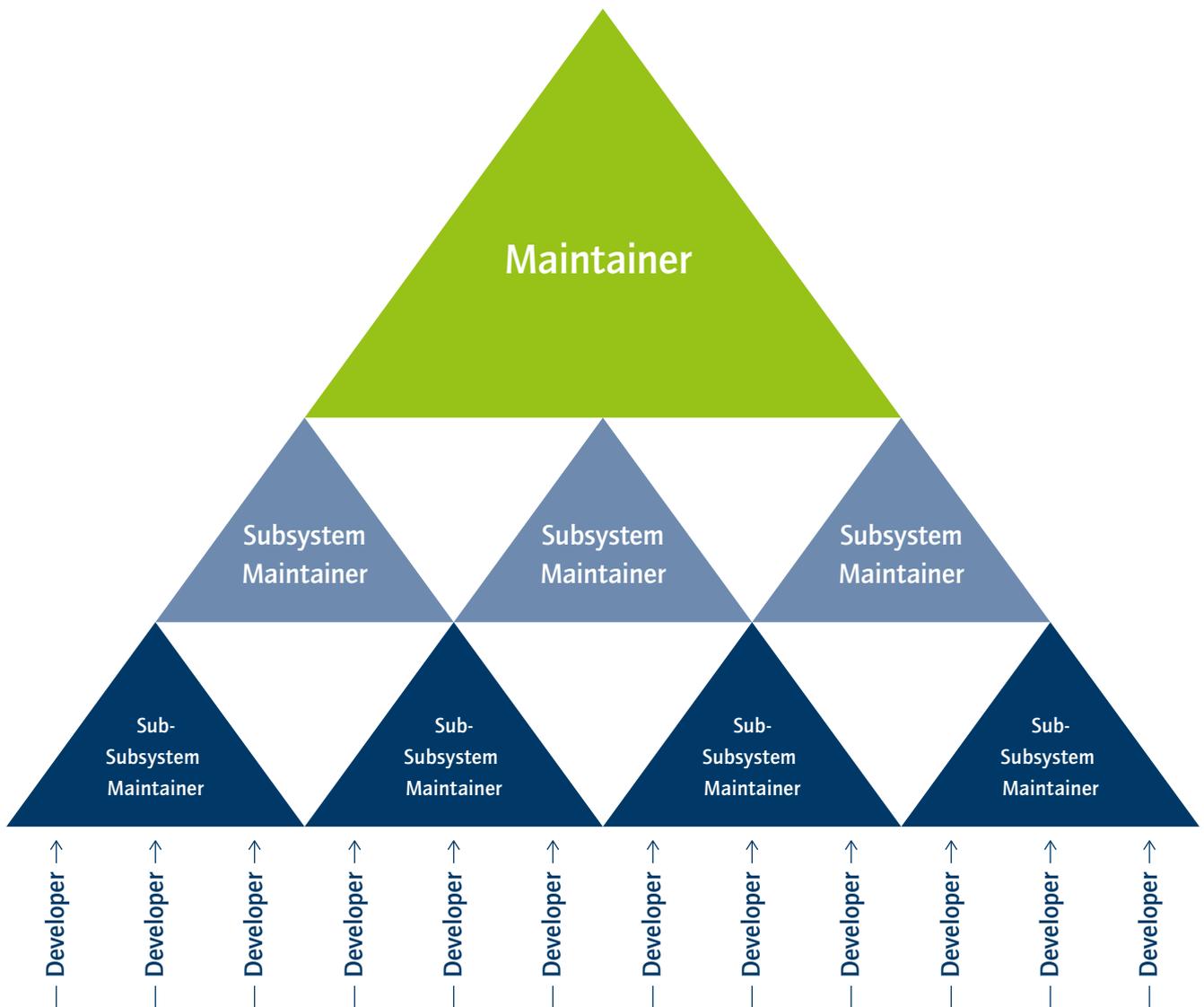
93 Vgl. Appleyard/Chesbrough 2017, West/Bogers 2014.

94 Vgl. West/Gallagher 2006.

95 Vgl. GitHub 2022a.

96 Vgl. Eclipse Foundation 2020.

Abbildung 7: Beispielhafter hierarchischer Aufbau einer Entwicklungs-Community



Quelle: eigene Darstellung in Anlehnung an Aniszczuk 2021

Entwicklungs-Communities sind die Schlüsselkomponente eines jeden Open Source-Projekts. Sie werden ins Leben gerufen von Personen, die ein Projekt initiieren und im Besitz der administrativen Eigentümerschaft der Organisation beziehungsweise der Open Source Repositories sind. Diese Rolle innerhalb eines Open Source-Projekts heißt *Autor* oder *Eigner*. Zur Community gehören des Weiteren Personen, die zur Entwicklung des Projekts beitragen, sei es durch Diskussionsbeiträge, Verbesserungsvorschläge für Dokumentation und Code oder anderes mehr. Diese Rolle nennt sich *Kontributor*. Einige Kontributorinnen und Kontributoren haben eine besondere Rolle, sie sind für die Umsetzung der Vision und die organisatorischen Aspekte des Projekts verantwortlich und werden *Maintainer* genannt.

Eine aktive und engagierte Anwendungs-Community von Personen, die die veröffentlichten, beispielhaften Anwendungen des Projekts nutzen, ist der Beweis dafür, wie nützlich sie sind und dass sie gebraucht werden. Wie für andere Allgemeingüter besteht auch bei Open Source Software das Problem, dass viele Personen eine OSS nutzen, ohne dazu beizutragen (Trittbrettfahrer, engl.: free rider problem).⁹⁷ Wenn zu wenige Personen oder Organisationen zu einem Projekt beitragen, kann das Projekt stagnieren und schließlich aufhören zu existieren. Eine zentrale Aufgabe des Managements von OSS-Projekten ist daher, das (zahlenmäßige) Verhältnis zwischen der Entwicklungs- und der Anwendungs-Community auszubalancieren. Diese Balance ist dabei lösungsabhängig und kann nicht exakt definiert werden. Es lässt sich beobachten, dass Personen, die anfänglich nur nutzen und nicht beitragen, im Laufe der Zeit

97 OSS kann als öffentliches Gut betrachtet werden. Das Problem entsteht, wenn ein öffentliches Gut stark nachgefragt und genutzt wird, aber von den Nutzenden zu wenig für dessen Erhalt unternommen wird. Dies ist allgemein als die „Tragik der Allmende“ bekannt.

aktiver werden.⁹⁸ Die sogenannte Adopter Community stellt eine Spezialform der Gemeinschaft der Anwendenden dar. Darunter subsumiert werden Personen, die die angebotenen Frameworks und Anwendungen nicht nur nutzen, sondern innerhalb ihrer Unternehmen in ihre Unternehmens-IT einbetten (beispielsweise über API) und (proprietäre) Erweiterungen erstellen. Auch hier gilt, dass der Aufbau, die Förderung und Pflege einer solchen Community außerhalb des Entwicklungsprojekts Zeit, Energie und Kreativität seitens der Projektleitung erfordert, aber dem langfristigen Erfolg des Open Source-Projekts zuträglich ist.

Governance von Open Source-Projekten

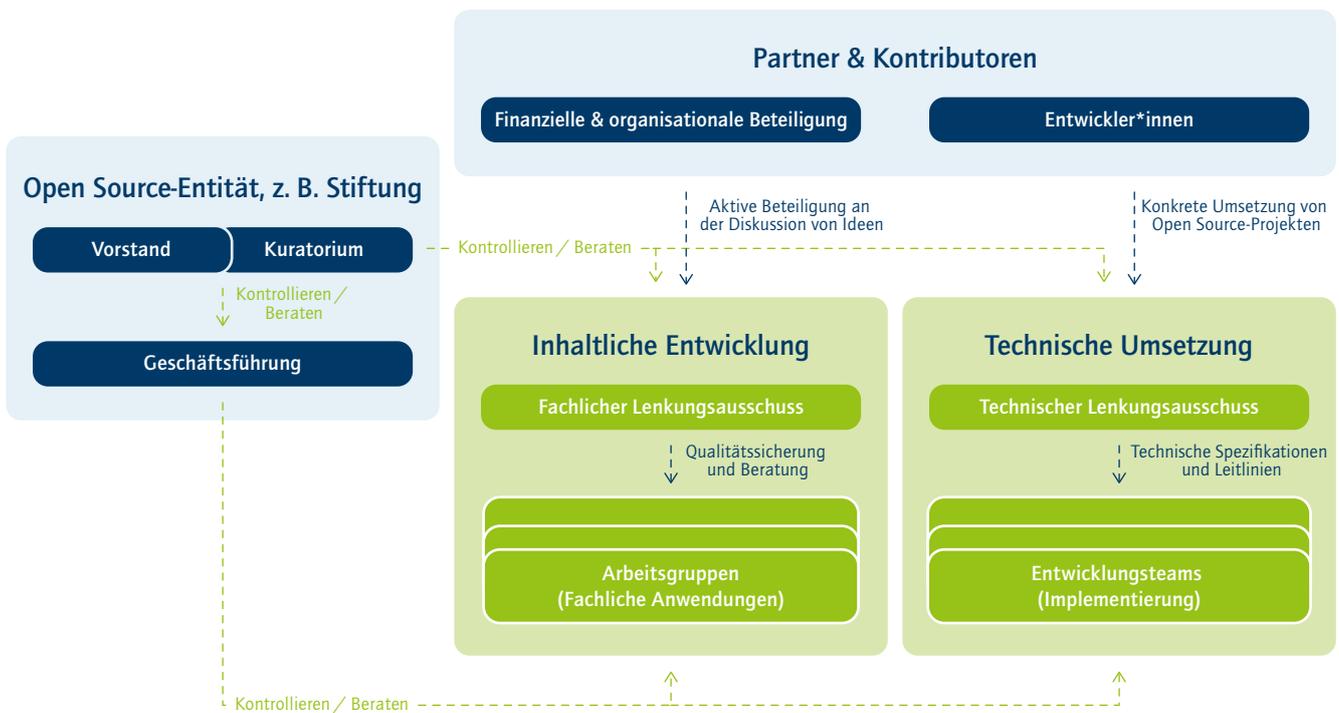
Die OSS-Gemeinschaft insgesamt und insbesondere der für Industrie 4.0 relevante Teil haben ein Modell der Zusammenarbeit entwickelt, das über das frühere Verständnis der von Individuen freiwillig getragener Kollaboration hinausgeht und von einer von der Industrie betriebenen kontinuierlichen Forschungs- und Entwicklungszusammenarbeit geprägt ist. Dabei gilt weiterhin, dass Einzelpersonen und auch Organisationen nur dann zum Entwicklungsprozess beitragen, wenn dies in ihrem eigenen Interesse ist. Teilnehmende können nicht gezwungen werden, eine Entscheidung der Gemeinschaft umzusetzen. Sie haben immer die Wahl, sich nicht an der Umsetzung von Entscheidungen zu beteiligen oder die Gemeinschaft ganz zu verlassen. Aus diesem Grund streben OSS-Gemeinschaften im Allgemeinen einen Konsens an, wenn sie

Entscheidungen treffen, und nehmen häufig Rücksicht auf Minderheitenmeinungen.⁹⁹

Regeln, Gepflogenheiten und Prozesse, die festlegen, welche Mitwirkenden die Befugnis haben, bestimmte Aufgaben auszuführen, werden zunehmend in Governance-Modellen formuliert und formalisiert. In der Praxis haben sich verschiedene Governance-Modelle herausgebildet. Zum einen können offene, einladende Strukturen identifiziert werden, die demokratisch und konsensbasiert vorgehen. Sie zeichnen sich dadurch aus, dass die Entscheidungskontrolle innerhalb eines Konsortiums verteilt oder durch einen dezentralen Peer-to-Peer-Ansatz ausgeübt wird. Geringe Eintrittshürden signalisieren die Offenheit. Gepaart mit transparenten (Entscheidungs-) Vorgängen sind sie einer Zusammenarbeit in großem Maßstab zuträglich. Die andere Seite des Spektrums bilden geschlossene und zentralistische Governance-Modelle, wie im Fall des Android Open Source-Projekts. Solche Projekte werden durch eine Entität dominiert. Diese Dominanz kann so weit gehen, dass das Urheberrecht zentral bei einer Organisation liegt.

Der letzte Entscheidungsträger in einer OSS-Gemeinschaft ist die Gesamtheit aller Mitwirkenden, wobei die Stimmen möglicherweise nach dem Verdienst der Mitwirkenden in der leistungsorientierten Organisation gewichtet werden (meritokratisches System). In Abhängigkeit vom jeweiligen Ansatz der Gewichtung von Stimmen können verschiedene Governance-Modelle unterschieden werden.

Abbildung 8: Beispiel des organisationalen Zusammenspiels von Stiftung, Projekten und Unternehmen im Open Source-Kontext



Quelle: eigene Darstellung

98 Vgl. West/Bogers 2014.

99 Vgl. Blind/Böhm 2019, Böhm 2019.

Sie reichen von der Führung eines Open Source-Projekts durch einen „wohlwollenden Diktator“ bis hin zu Modellen mit kooperativen und egalitären Merkmalen.¹⁰⁰ Repräsentative Gremien, die entweder von den Mitwirkenden gewählt oder nach den finanziellen Beiträgen von Organisationen zur Gemeinschaft ernannt werden, sind mit der täglichen Verwaltung betraut, können aber nur selten Entscheidungen treffen, auch nicht gegen eine einflussreiche Minderheit von Mitwirkenden.¹⁰¹

Einige Projekte beschäftigen technisches oder administratives Personal. Die Community-Organisationen agieren dann ähnlich wie Unternehmen, die einen Projektauftrag ausführen. Nur wenige OSS-Gemeinschaften sind als unabhängige juristische Personen gegründet worden. Die meisten größeren Kollaborationen, die sich in den letzten Jahren gebildet haben, sind unter einer Dachorganisation, wie der Eclipse Foundation, der Apache Foundation, der gemeinnützigen Organisation Software in the Public Interest oder der Linux Foundation angesiedelt, die als juristische Personen administrative Unterstützung, technische Infrastruktur und andere Funktionen wie Community-Koordination, Veranstaltungsorganisation, Marketing und Fundraising-Koordination etc. bereitstellen.¹⁰²

5.2. Einbettung von Open Source-Projekten in Unternehmen

Einhergehend mit der zunehmenden Akzeptanz, Verbreitung und Professionalisierung von OSS-Projekten ist eine voranschreitende Professionalisierung von Strukturen und Prozessen innerhalb der Open Source entwickelnden Unternehmen zu beobachten. Ein Ausdruck dieser Entwicklung und von besonderer Bedeutung ist das OpenChain-Projekt, inklusive des hieraus hervorgegangenen ISO/IEC-Standards 5230:2020 für Open Source Compliance, der die Kernanforderungen an ein Qualitätssicherungsprogramm definiert, und der Konformitätsmethode Open Source Program Offices (OSPO), die Unternehmen in der Einhaltung dieser Anforderungen unterstützt. Sie werden im Folgenden kurz dargestellt.

OpenChain-Projekt und Compliance-Standard

Das Ziel der im Jahr 2013 ins Leben gerufenen und 2016 von der Linux Foundation übernommenen Initiative OpenChain ist es, Kern- beziehungsweise Minimalanforderungen an ein Compliance-Programm für OSS zu definieren. Motiviert wurde die Initiative dadurch, dass der überwiegende Anteil aktueller Software nicht mehr von Grund auf neu entwickelt wird, sondern auf Open Source-Ressourcen Dritter basiert. Die Verwendung von vorgefertigten Bibliotheken und Open Source-Komponenten beschleunigt die Entwicklung, senkt Produktionskosten und verkürzt die Dauer von der Produktentwicklung bis zur Platzierung des Produkts am

Markt (Time-to-Market). Unternehmen mussten in der Folge anerkennen, dass Unternehmenssoftware teilweise außerhalb ihrer eigenen Organisationen und Netzwerke entsteht. Im Zuge der Absicherung der so entstehenden Software Supply Chains begannen Unternehmen, Compliance-Richtlinien zu entwickeln, die im OpenChain-Projekt gebündelt und harmonisiert wurden und im Zuge der Etablierung als ISO/IEC-5230-Standard zusätzliche Legitimation erhielt.

Compliance nach dem OpenChain-Projekt umfasst dabei Rechte und Pflichten im Rahmen der Distribution und der Modifikation von OSS sowie Aufgaben und Qualitätsstandards zur Sicherung des (internen) Gebrauchs von OSS. Häufige Compliance-Maßnahmen sind die Identifizierung aller in der gelieferten Software verwendeten Open Source-Komponenten, eine Bestimmung aller Verpflichtungen, die durch diese Komponenten beziehungsweise ihre Lizenzen entstehen und das Nachverfolgen, ob alle Verpflichtungen erfüllt wurden oder werden. Die Verantwortung zur Durchführung dieser Prozesse wird in der Regel in Open Source Program Offices (OSPO) verortet.

Open Source Program Office

Seit Beginn der 2010er Jahre haben insbesondere Software- und IT-Unternehmen OSPO als Best Practice für die interne Open Source-Verwaltung übernommen und tauschen sich über semiformale Netzwerke, wie die von der Linux Foundation im Jahr 2012 gegründete TODO Group, aus. Das Konzept hat in den letzten Jahren in der betrieblichen Praxis Fuß gefasst und beschränkt sich mittlerweile nicht mehr nur auf den Softwaresektor. OSPO sind, wenn auch in deutlich geringerem Umfang, im öffentlichen Sektor und Bildungsbereich sowie in den Bereichen Informations- und Kommunikationstechnologie, Logistik, Automotive, Healthcare, Handel und Versicherungswesen anzutreffen.¹⁰³

Ein OSPO ist wie ein Kompetenzzentrum für die Open Source-Abläufe und -Strukturen einer Organisation konzipiert. Es kann Richtlinien für die Verwendung, Verteilung, Auswahl, Prüfung und andere Aspekte des Codes festlegen, aber auch die Schulung von Entwicklerinnen und Entwicklern organisieren, die Einhaltung rechtlicher Vorschriften sicherstellen sowie den Aufbau und das Engagement von Communities fördern, wenn das der Organisation strategisch zugutekommt.¹⁰⁴

Es gibt keine allgemeingültige Vorlage für den Aufbau eines Open Source-Programms, die für alle Unternehmen einer Branche oder sogar für alle Branchen gilt. Man kann die Funktionen eines typischen OSPO jedoch unterscheiden in:

- Minderung rechtlicher Risiken (zum Beispiel Reviews und Handlungsempfehlungen zu Lizenzkompatibilität)

100 Vgl. RedHat 2020.

101 Vgl. Blind/Böhm 2019, Böhm 2019.

102 Vgl. Izquierdo/Cabot 2018.

103 Vgl. GitHub 2021 a.

104 Vgl. GitHub 2021 b.

- Verbesserung der Praktiken von Entwicklungs-Teams (zum Beispiel Auf-/Ausbau und Pflege der unternehmensinternen Open Source-Kultur, Schnittstelle zu externen Entwicklerinnen und Entwicklern, Aufsetzen von Infrastruktur und Werkzeugen zum Erstellen und Nutzen von Open Source, Sicherung einer hohen Codequalität und einer angemessenen Dokumentation)
- Erschließung finanzieller Vorteile (zum Beispiel Definition, Kommunikation und Umsetzung einer Open Source-Strategie, Sicherstellung des effizienten Einsatzes von Open Source in Produkten und Services)

In Abhängigkeit vom Hauptgeschäft eines Unternehmens und von der gewählten Open Source-Strategie haben sich in der Praxis verschiedene Formen der Einbettung in Unternehmen entwickelt. Typischerweise sind folgende Ansätze zu finden.¹⁰⁵

- **OSPO im Bereich Recht:** Diese Form ist insbesondere in Unternehmen mit umfangreichen Patent-Portfolios, beispielsweise Unternehmen aus dem Bereich Elektronik beziehungsweise Hardware-Entwicklung, zu finden. Compliance-Themen sowie patentrechtliche Fragen lassen sich so meist zügig klären.
- **OSPO im Bereich Entwicklung:** In ingenieurwissenschaftlich orientierten Umfeldern sind OSPO in technischen Abteilungen oder der Produktentwicklung verortet. Hierdurch wird ein fachlich-inhaltliches Zusammenwirken von OSPO und technischer Entwicklung befördert. Compliance-Themen stehen tendenziell weniger im Fokus als in OSPO unter Führung von Rechtsabteilungen.
- **OSPO im Bereich Marketing:** In einigen Unternehmen sind OSPO in der Marketinggruppe angesiedelt. In diesen Fällen wird Open Source häufig als Vertriebsstrategie genutzt. Durch sogenannte Developer Relations-Abteilungen verbinden sich Software-Unternehmen zunehmend mit Entwicklungs-Communities.

5.3. Eigenschaften und Messgrößen gesunder Open Source-Projekte und Open Source-Ökosysteme

Die Wahl zwischen verschiedenen Open Source-Projekten ist eine zentrale und herausfordernde Aufgabe. Eine einmal getroffene Entscheidung wird viele der zukünftigen Entwicklungen und Tätigkeiten innerhalb einer Organisation bestimmen. Im Kern der Auswahlentscheidung steht die Frage, ob eine Nutzung und Beteiligung als Organisation sinnvoll und erstrebenswert ist (siehe Abbildung 6). Zentral für die Beantwortung dieser Frage ist die Beurteilung der „Gesundheit“, gemeint ist mit diesem Begriff vor allem die Nachhaltigkeit, des Projekts beziehungsweise des Ökosystems. Ein gesundes und langfristig fortbestehendes Open Source-Ökosystem

beziehungsweise -Projekt ist nicht nur für die Nutzenden wichtig. Auch andere Beteiligte, zum Beispiel Personen, die ihre Arbeitskraft oder ihr Kapital einsetzen, wollen wissen, ob ihr Ökosystem gesund ist und gut funktioniert. In der wissenschaftlichen Literatur sind verschiedene Ansätze zur Bestimmung der Gesundheit von Projekten und Ökosystemen zu finden.

Auf der Basis einer umfassenden Recherche entwickelt Jansen eine Übersicht gängiger Messgrößen zur Bestimmung der Gesundheit, verstanden als Langlebigkeit und Wachstumstendenz von Open Source-Ökosystemen.¹⁰⁶ Das von Jansen entwickelte Rahmenwerk unterscheidet Messgrößen in den Bereichen Produktivität, Robustheit und Nischenbildung beziehungsweise Spezialisierung. Dabei werden Messgrößen von Netzwerken, das heißt vollständigen Open Source-Ökosystemen, und projektspezifische Messgrößen unterschieden. Diese auf Crowston et al. basierende Trennung der Betrachtungsebenen berücksichtigt die Tatsache, dass Open Source-Projekte häufig nicht als isolierte Projekte betrachtet werden können, sondern Teil eines Netzwerks verschiedener Projekte sind, die sich um ein sogenanntes Keystone-Projekt, ein zentrales Projekt, herum organisieren.¹⁰⁷ Die Messgrößen der Netzwerkebene betreffen Metriken, die die Gesundheit des Ökosystems beschreiben und nur auf dieser Ebene beschrieben werden können (beispielsweise projektübergreifende Veranstaltungen, das Entstehen neuer Projekte im Netzwerk und ihre Verbindungen sowie Konsistenz zueinander). In der darunterliegenden Projektbetrachtungsebene werden allgemeine Projektkennzahlen definiert und erfasst.

Ein anderer Ansatz bestimmt qualitative Eigenschaften von Open Source-Ökosystemen.¹⁰⁸ Zentraler Punkt ist hier die Unterscheidung zwischen der Qualität der Open Source Community und netzwerkbezogenen Messgrößen. Die Qualität der Open Source Community lässt sich bestimmen durch:

- **Erhaltungsfähigkeit**, definiert als die Fähigkeit einer Gemeinschaft, die für die Erhaltung ihrer Produkte erforderlichen Ressourcen bereitzustellen (gemessen unter anderem an der Größe, Aktivität und dem Zusammenhalt)
- **Nachhaltigkeit**, verstanden als die Wahrscheinlichkeit, dass eine Gemeinschaft in der Lage bleibt, die von ihr entwickelten Produkte über einen längeren Zeitraum hinweg aufrechtzuerhalten (ermittelt unter anderem durch Heterogenität und Regenerierungsfähigkeit der Gemeinschaft, Ausgeglichenheit beziehungsweise Ausgewogenheit der Beteiligung und Sichtbarkeit des Projekts beziehungsweise Ökosystems)
- **Prozessreife**, verstanden als die Fähigkeit einer Entwicklungsgemeinschaft, durchgängig entwicklungsbezogene Ziele durch Befolgung etablierter Prozesse zu erreichen (beispielsweise durch etablierte Governance-Regelungen oder einen funktionierenden Code of Conduct).

¹⁰⁵ Vgl. Aniszczyk 2021.

¹⁰⁶ Vgl. Jansen 2014.

¹⁰⁷ Vgl. Crowston et al. 2006.

¹⁰⁸ Vgl. Franco-Bedoya et al. 2015.

Daneben haben sich aus der Open Source-Praxis heraus verschiedene Vorhaben wie das CHAOSS-Projekt der Linux Foundation oder Good-Practice-Dokumentationen, beispielsweise „The Open Source Way-Guidebook“ oder das „Open Source 101“ der TODO Group, entwickelt, die dabei unterstützen, die Leistung und Gesundheit von OSS-Projekten zu bestimmen und Gestaltungshilfen zu geben.¹⁰⁹ Neben eindeutig messbaren Kriterien und Kennzahlen (lines of code, number of contributors and downloads etc.) werden immer auch qualitative beziehungsweise „weiche“ Faktoren als zentral für Erfolg und Misserfolg beschrieben und in der Regel im Rahmen von Code of Conducts formuliert. Besonders plakativ geschieht dies in „The Apache Way“ der Apache Software Foundation:

„The biggest mantra at Apache is: ‚Community over code‘. In brief, this means that the most successful long-lived projects value a broad and collaborative community over the details of the code itself. [...] Obviously, the code needs to be useful; that’s what makes it worthwhile to work on over time. But while the best code around right now may turn into a tool that everyone uses today, the best community around will create and maintain a project that everyone helps build and improve for the long term.“¹¹⁰

Eine zusammenfassende Übersicht gängiger Messgrößen wird in Tabelle 3 gegeben. Bei Anwendung dieser Messgrößen ist zu berücksichtigen, dass OSS auch eine Komposition verschiedener Softwarekomponenten darstellen kann. Wie in Kapitel 4.1 erläutert, werden auch Komponenten entwickelt, die alleinstehend nicht nutzbar sind, sondern explizit für die Kombination mit anderen Komponenten entwickelt wurden. Dementsprechend kann es irreführend sein, die Messgrößen auf ein einzelnes Quellcode-Repository anzuwenden. Es lassen sich Software-Komponenten finden, die auf den ersten Blick nicht weiterentwickelt werden, da keine neuen Code-Beiträge in Form von sogenannten Commits erfolgen. Isoliert betrachtet wirken solche Repositorien verwaist. Man könnte auf eine inaktive oder nicht länger existente Community schließen. Zwar kann dies der Fall sein, genauso besteht aber die Möglichkeit, dass es sich um eine Kernkomponente handelt, an der nur sehr selten Veränderungen durchgeführt werden (müssen). Ein Beispiel wäre ein Datenmodell, das einmal definiert nicht verändert werden muss. Die Community entwickelt möglicherweise in anderen Repositories sehr aktiv Komponenten, die die Kernkomponente um weitere Funktionalität erweitern. Die Community kann dementsprechend intakt sein, auch wenn dies aus einer Bewertung des isolierten Repository mithilfe der gewählten Messgröße nicht hervorging. Gerade bei der Betrachtung quantitativer Messgrößen muss dies berücksichtigt werden.

109 Vgl. GitHub 2022b, Proffitt et al. 2020.

110 Siehe Curcuru 2018.

Tabelle 3: Gängige Open Source-Messgrößen

Kategorie	Kriterien	Quantitative Messkriterien	Qualitative Messkriterien / Indikatoren
Vitalität / Lebendigkeit	Aktivität	<ul style="list-style-type: none"> Anzahl Commits (hinzugefügte Dateien / Codezeilen, geänderte Dateien / Codezeilen) Abgelehnte Änderungen Zeitliche Abstände der Commits Anzahl Forks Anzahl Releases Anzahl aufkommender Tickets (Probleme) Anzahl bearbeiteter Tickets Alter der Tickets Dauer zur Klärung / Bearbeitung eines Tickets Nutzeraktivitätsdiagramme Anzahl Downloads Anzahl Events und Meetings 	
	Größe	<ul style="list-style-type: none"> Anzahl der Beitragenden (Committers) Anzahl der Organisationen, die das Projekt nutzen 	
	Wachstum und Weiterentwicklung	<ul style="list-style-type: none"> Wachstum der Beitragenden (Committer) Gesamtgröße der Zielgruppe Inaktive Beitragende Entwicklungsgeschwindigkeit Kontinuität der Aktivitäten (Burstiness) 	<ul style="list-style-type: none"> Sichtbarkeit / Reichweite (Web-Präsenz, Social Media-Präsenz) Mentorship Risiko der Abwanderung (Bus Factor)
	Widerstandsfähigkeit	<ul style="list-style-type: none"> Dauer seit Projektgründung Anzahl der Partnerschaften mit anderen Projekten Anzahl der Stakeholder Anzahl der Kapitalgeber / Sponsoren Vielfalt der unterstützten Technologien Wie sind die Aufgaben verteilt? (Elephant Factor) 	<ul style="list-style-type: none"> Organisatorische Reife
Strukturiertheit / Qualität	Projektstruktur		<ul style="list-style-type: none"> Ersichtlichkeit der Projektziele Struktur der Verantwortlichkeiten Entwicklungsmodell
	Dokumentation		<ul style="list-style-type: none"> Qualität der Dokumentation Verantwortlichkeiten für die Dokumentation Zugänglichkeit der Dokumentation
	Code	<ul style="list-style-type: none"> Code-Qualität, z.B. Testabdeckung, Bugs, Code Smells Laufzeiteffizienz Testabdeckung Getestete Subroutinen Getestete Statements Laufzeitoptimiert Speichereffizient 	<ul style="list-style-type: none"> Automatisierungsgrad des Entwicklungsprozesses
Community & Ökosystem	Diversität	<ul style="list-style-type: none"> Unterstützte Sprachen Anzahl verschiedener Teilnehmernationen 	<ul style="list-style-type: none"> Backgrounds der Teilnehmenden
	Kommunikation / Outreach	<ul style="list-style-type: none"> Anzahl der Kommunikationskanäle 	<ul style="list-style-type: none"> Moderation der Kommunikationskanäle Kommunikation der Führung
	Lizenz	<ul style="list-style-type: none"> Lizenzabdeckung (License Coverage) OSI Approved Licenses 	<ul style="list-style-type: none"> Etablierung der Lizenz
	Projektkultur (intern)	<ul style="list-style-type: none"> Verhaltenskodex (Code of Conduct) Einhaltung des Verhaltenskodex 	<ul style="list-style-type: none"> Inklusion Führungsstandard

6. Strukturierte Erfassung von Open Source-Initiativen

Open Source-Projekte beziehungsweise Open Source-Ökosysteme zeichnen sich durch inhaltliche Vielschichtigkeit und eine große Bandbreite an organisatorischen Ausprägungen und strategischen Ausrichtungen aus. Die nachfolgenden Abschnitte haben zum Ziel, die bisherigen Erkenntnisse der Kapitel 3 und 4 in ein einheitliches Rahmenwerk zu überführen, das eine schnelle und gleichzeitig strukturierte Zuordnung beziehungsweise Einordnung von Open Source-Projekten und ihren Ökosystemen erlaubt. Hiermit wird ein Werkzeug bereitgestellt, das Unternehmen bei der zielgerichteten Auswahl existierender Projekte unterstützt und Handlungsoptionen für die Gestaltung der eigenen Beteiligung liefert.

6.1. Open Source-Taxonomie

In der Praxis hat sich eine Vielzahl an unterschiedlichen Projekten entwickelt, die auf dem Konzept von Open Source basieren. Neben den eingesetzten Geschäftsmodellen werden Open Source-Projekte durch ihre Communities geprägt, da sie an der Produktentwicklung teilhaben (siehe Kapitel 5.1).¹¹¹ Um Potenziale von Open Source voll ausschöpfen zu können, sollte deshalb während der Planung eines Open Source-Projekts gleichzeitig eine Community-Strategie entwickelt werden. Open Source-Projekte entstehen sowohl im kommerziellen als auch im nicht kommerziellen Rahmen und zeichnen sich durch ihre Heterogenität untereinander aus. Um trotz ihrer Heterogenität allgemeine Gestaltungsoptionen zu identifizieren, wird eine Dekomposition essenzieller Elemente von Projekten benötigt. Hierfür eignen sich Dimensionen, die jedes Open Source-Projekt charakterisieren und einordnen können. Dies ermöglicht zudem den Nachbau erfolgreicher Projektstrukturen bei der Gestaltung eigener Open Source-Projekte. Zur Identifizierung relevanter

Dimensionen und Charakteristika wird zunächst eine Taxonomie zur Beschreibung von Open Source-Projekten im Kontext von Industrie 4.0 entwickelt.

Taxonomien werden zur Strukturierung komplexer Themenfelder eingesetzt. Sie gliedern diese in ihre elementaren Bestandteile auf. So können komplexe Strukturen durchdrungen und eine Wissensgrundlage für weitere Untersuchungen geschaffen werden.¹¹² Zur Entwicklung der Open Source-Taxonomie wird eine Kombination konzeptioneller und empirischer Herangehensweisen gewählt. Zum einen werden bestehende Konzepte aus der Literatur extrahiert, zum anderen werden existierende Open Source-Projekte im Bereich von Industrie 4.0 analysiert und in die Erstellung der Taxonomie mit einbezogen. Durch den holistischen und iterativen Entwicklungsansatz wird sichergestellt, dass alle relevanten Dimensionen und Charakteristika von Open Source-Projekten abgedeckt sind. Eine strukturierte Literaturrecherche identifiziert Zeitschriftenaufsätze, Bücher und Dokumente für die theoretische Wissensbasis.¹¹³ Insbesondere Dokumente aus etablierten Communities, beispielsweise Reports oder Whitepaper, ermöglichen die praxisnahe Erfassung relevanter Dimensionen für die Charakterisierung von Open Source-Projekten.

Aufgrund der hohen Relevanz aktiver Communities im Open Source-Bereich sind ihre Charakteristika zentral für die Analyse von Open Source-Projekten. So bilden neben Geschäftsmodellen (MD₁) auch das Ökosystem (MD₂) und die dazugehörige Governance (MD₃) häufig die Grundlage für eine Beschreibung von Open Source-Projekten.¹¹⁴ Daher bilden diese drei Metadimensionen den übergeordneten Rahmen der nachfolgend beschriebenen Taxonomie für Open Source in Industrie 4.0 (siehe Tabelle 4).

111 Vgl. Raymond 2001.

112 Vgl. Nickerson et al. 2013.

113 Vgl. Vom Brocke et al. 2015.

114 Vgl. Mozilla/Open Tech Strategies 2018.

Tabelle 4: Taxonomie zur Beschreibung und Analyse von Open Source-Projekten im Kontext von Industrie 4.0

Dimension		Dimension					
Geschäftsmodell (MD ₁)	Technologiebereich (D ₁)	Konnektivität und Kommunikation (C ₁₁)	Prozessautomation und -flexibilisierung (C ₁₂)	Daten und Analytik (C ₁₃)	Sicherheit und Vertrauen (C ₁₄)	Mensch-Technik-Interaktion (C ₁₅)	
	Produktangebot (D ₂)	Commodity (C ₂₁)	Werkzeug (C ₂₂)	Infrastruktur (C ₂₃)	Support und Service (C ₂₄)	Standards und Schnittstellen (C ₂₅)	
	Wertversprechen (D ₃)	Funktioneller Wert (C ₃₁)	Ökonomischer Wert (C ₃₂)	Kollaborativer Wert (C ₃₃)	Markenwert (C ₃₄)	Sozialer Wert (C ₃₅)	
	Lizenzierung (D ₄)	Duale Lizenzierung (C ₄₁)	Permissive Lizenzierung (C ₄₂)	Copyleft-Lizenzierung (C ₄₃)	Nicht spezifiziert (C ₄₄)		
	Finanzielle Ausrichtung (D ₅)	Gewinnorientiert (C ₅₁)		Gemeinnützig (C ₅₂)		Unklar (C ₅₃)	
	Ertragsstrategie (D ₆)	Direkte Bepreisung (C ₆₁)	Indirekte Bepreisung (C ₆₂)	Förderung (C ₆₃)		Nicht monetär (C ₆₄)	
Ökosystem (MD ₂)	Projektbetreiber (D ₇)	Getrieben durch zentralen Akteur (C ₇₁)		Konsortium (C ₇₂)		Peer-to-Peer-Netzwerk (C ₇₃)	
	Betreiberhintergrund (D ₈)	Industrie (C ₈₁)		Forschung (C ₈₂)		Gemeinnütziger Bereich (C ₈₃)	Individuum (C ₈₄)
	Schlüsselpartner (D ₉)	Unternehmen (C ₉₁)		Forschungseinrichtung (C ₉₂)		Gemeinnützige Organisation (C ₉₃)	Individuen (C ₉₄)
	Nutzer*innen (D ₁₀)	Business (C ₁₀₁)	Konsument*innen (C ₁₀₂)	Entwickler*innen (C ₁₀₃)	Gemeinnützige Organisation (C ₁₀₄)		Staat (C ₁₀₅)
Governance (MD ₃)	Entscheidungskontrolle (D ₁₁)	Zentral (C ₁₁₁)		Ausgeglichen (C ₁₁₂)		Dezentral (C ₁₁₃)	
	Community-beziehung (D ₁₂)	Symbiotische Beziehung (C ₁₂₁)		Kommensalistische Beziehung (C ₁₂₂)		Parasitäre Beziehung (C ₁₂₃)	
	Onboarding (D ₁₃)	Einfach (C ₁₃₁)		Mittel (C ₁₃₂)		Schwierig (C ₁₃₃)	

Geschäftsmodell

Geschäftsmodelle (MD₁) beschreiben, wie eine Organisation funktioniert und wie sie Gewinne erwirtschaftet. Mithilfe von Geschäftsmodellen lassen sich anhand einzelner Grundbausteine unternehmerische Erfolgsfaktoren identifizieren, eine weit verbreitete Analysemethode in Theorie und Praxis.¹¹⁵ Für die Analyse von Open Source-Projekten im Industrie 4.0-Kontext werden in der ersten Dimension die Technologiebereiche des Projekts untersucht.

Diese Untersuchung orientiert sich an den in Abbildung 3 vorgestellten Technologiebereichen: Konnektivität und Kommunikation (C₁₁), Prozessautomation und -flexibilisierung (C₁₂), Daten und Analytik (C₁₃), Sicherheit und Vertrauen (C₁₄) und Mensch-Technik-Interaktion (C₁₅). Im industriellen Bereich teilt sich das aus empirischen Fallbeispielen abgeleitete **Produktangebot (D₂)** in Commodity (C₂₁), Werkzeug (C₂₂), Infrastruktur (C₂₃), Support und Services (C₂₄) und Standards und Schnittstellen (C₂₅) auf. Eine genauere Erklärung der Kategorien folgt in Tabelle 5.

115 Vgl. Osterwalder 2004.

Tabelle 5: Produktangebote von Open Source-Projekten

Produktangebot	Beschreibung
Commodity	Standarddienstleistungen, die nicht geschäftsdifferenzierend sind, jedoch von Kundinnen und Kunden erwartet werden (z. B. Track and Trace).
Werkzeug	Digitale Werkzeuge, auf deren Basis weitere Produkte entwickelt werden können (z. B. Entwicklungssprachen).
Infrastruktur	(Digitale) Infrastruktur, die Kommunikations-, Kollaborations- und/oder Datenverarbeitungsfunktionen bietet (z. B. Speicher, Server, Netzwerke, Rechenkapazitäten oder physische Datenzentren).
Support und Services	Dienstleistungen, wie z. B. Training oder Beratung.
Standards und Schnittstellen	Standards und Schnittstellen fassen vereinheitlichende Maßnahmen (z. B. Modelle und Methodiken) sowie Komponenten, die zur Kommunikation mit einem System eingesetzt werden (z. B. APIs), zusammen.

Um ein erfolgreiches Produkt zu entwickeln, muss ein Unternehmen die Kundenbedürfnisse verstehen und für ihre Befriedigung ein Angebot machen, indem es ein **Wertversprechen (D₃)** schafft. Bei Open Source-Produkten können das sein: das Produkt selbst (C₃₁), ökonomische Vorteile (zum Beispiel das Preis-Leistungs-Verhältnis) (C₃₂), Vorteile durch Kollaborationen (C₃₃), der Markenwert (C₃₄) oder soziale Werte (C₃₅).¹¹⁶ Die **Lizenzierung (D₄)** eines Open Source-Produkts beeinflusst die Wahl des übergeordneten Geschäftsmodells. So eignet sich die duale Lizenzierung (zum Beispiel proprietäre und Copyleft-Lizenz) (C₄₁) für kommerzielle Open Core-Modelle, wohingegen Copyleft-Lizenzen (C₄₃) sogenannte soziale Geschäftsmodelle begünstigen, die auf den langfristigen Erhalt eines Non-Profit-Projekts abzielen.¹¹⁷ Ebenso beeinflusst die **finanzielle Ausrichtung (D₅)** die Wahl des Geschäftsmodells, wobei **Ertragsstrategien (D₆)** unabhängig von dieser eingesetzt werden können. So können Start-ups beispielsweise eine kommerzielle Version der OSS vertreiben (direkte Bepreisung) (C₆₁) und sich durch Förderungen (C₆₃) finanzieren. Andererseits nutzen Non-Profit-Organisationen neben Förderungen und nicht monetären Leistungen (zum Beispiel unvergüteter Arbeit) (C₆₄) indirekte Bepreisungen (zum Beispiel Merchandise) (C₆₂), um die Selbsterhaltungskosten ihres Projekts zu decken.¹¹⁸

Ökosystem

Aufgrund der kollaborativen Natur der Zusammenarbeit verschiedener Akteure im Open Source-Bereich stellt das **Ökosystem (MD₂)** einen zentralen Aspekt von Open Source-Projekten dar. Ein Projekt wird zumeist von einem **Projektbetreiber (D₇)** initiiert und gesteuert. Im Gegensatz zu klassischen proprietären Projekten, in denen meist eine zentrale Instanz (Keystone-Akteur) (C₇₁) den Projektbetreiber darstellt, zeichnen sich Open Source-Projekte durch vielfältige mögliche Betreiberstrukturen aus. So können Open

Source-Projekte beispielsweise zentral durch mehrere Instanzen (Konsortium) (C₇₂) oder durch viele Individuen (C₇₃) betrieben werden. Der Projektbetreiber kann aus der Industrie (C₈₁), der Forschung (C₈₂), dem Non-Profit-Bereich (C₈₃) und/oder aus dem privaten Bereich (C₈₄) stammen. **Schlüsselpartner (D₉)** lassen sich in Unternehmen (C₉₁), Forschungseinrichtungen (C₉₂), Non-Profit-Organisationen (C₉₃) und Individuen (C₉₄) einteilen.¹¹⁹ Zuletzt bilden die eigentlichen **Nutzer*innen (D₁₀)** der resultierenden OSS eine weitere Akteursdimension. Nach der Analyse von Fallbeispielen lassen sich folgende Anwendende unterscheiden: B2B-Bereich (C₁₀₁), individuelle Konsument*innen (C₁₀₂), Entwickler*innen (C₁₀₃), Non-Profit-Organisationen (C₁₀₄) oder staatliche Stellen (C₁₀₅).

Governance

Die letzte Metadimension behandelt die **Governance (MD₃)** der Entwicklungs-Community innerhalb eines Open Source-Projekts. Charakterisiert wird sie unter anderem durch die Entscheidungskontrolle (**D₁₁**), die bei einer zentralen Instanz liegen kann (C₁₁₁) oder aber durch etablierte Kontrollmechanismen verteilt wird. So werden bei einer ausgeglichenen Entscheidungskontrolle (C₁₁₂) externe Entwicklerinnen und Entwickler in interne Entwicklerkreise aufgenommen, wenn sie gewisse Anforderungen beispielsweise anhand ihrer vorherigen Leistungen in der Community oder in gezielten Qualifizierungsprogrammen erfüllen. In Peer-to-Peer-Netzwerken findet dagegen meist eine dezentrale Entscheidungsfindung und -kontrolle (C₁₁₃) statt. Eine weitere Möglichkeit, die Governance von Open Source-Projekten zu charakterisieren, bietet die Art der Beziehung zur **Community (D₁₂)**. Diese lässt sich in drei grobe Kategorien unterteilen: Organisationen, die sich mit hohem Engagement und positivem Einfluss in die Community einbringen, streben eine symbiotische Beziehung (C₁₂₁) an. In einer kommensalistischen Beziehung (C₁₂₂) existieren Organisationen

¹¹⁶ Vgl. Shanker 2012.

¹¹⁷ Vgl. Välimäki/Oksanen 2005, Dudley et al. 2007.

¹¹⁸ Vgl. Hecker 1999, Rajala et al. 2006.

¹¹⁹ Vgl. Weking et al. 2018.

und ihre dazugehörigen Communities nebeneinanderher, ohne sich gegenseitig zu schaden oder zu beeinflussen. Sobald eine Organisation ihrer Community (un-)bewusst schadet, zum Beispiel durch Missachtung von Lizenzvorschriften, liegt eine parasitäre Beziehung (C_{123}) vor.¹²⁰ Das letzte Merkmal für die Beschreibung der Governance eines Open Source-Projekts ist das **Onboarding (D_{13})** beziehungsweise die Frage, wie schwierig sich eine Teilnahme am Open Source-Projekt für Externe gestaltet. Je nach Teilnahmebedingungen, zum Beispiel notwendige Zertifizierungen oder fachliche Anforderungen, variiert das Onboarding zwischen einfach (C_{131}), mittel (C_{132}) und schwierig (C_{133}).¹²¹

6.2. Archetypen von Open Source-Projekten in Industrie 4.0

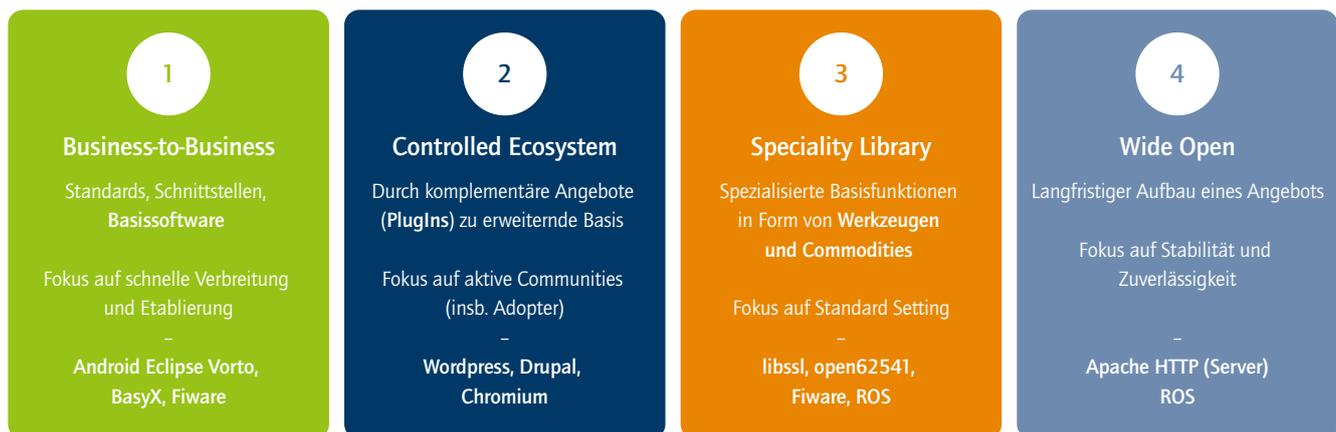
Durch die zuvor vorgestellten Dimensionen und Merkmale lassen sich eine Vielzahl heterogener Open Source-Projekte charakterisieren. Zur Bestimmung von Erfolgsfaktoren oder Barrieren, die für eine bestimmte Gruppe an Projekten zutreffend sind, empfiehlt es sich, Archetypen zu bilden. Sie beschreiben strukturdominante Gruppen, die beispielsweise durch Clusteranalysen abgeleitet werden, und als reproduzierbare Muster eingesetzt werden können. Aus einer Archetypen-Sammlung von The Mozilla Foundation und Open Tech Strategies sowie der Analyse von Fallbeispielen von Open Source-Projekten im Kontext von Industrie 4.0 lassen sich folgende vier Archetypen herleiten, die im Bereich von Industrie 4.0 von Relevanz sind (siehe Abbildung 9):

Business-to-Business Open Source

Der B2B-Archetyp zielt auf eine weite Verbreitung des Open Source-Projekts ab. Durch die Bereitstellung von Software sollen Anpassungen durch Industriepartner (zum Beispiel Zulieferer) erhöht und mögliche Konkurrenten verdrängt werden. Deshalb zählen Standards und Schnittstellen sowie Basissoftware zu geläufigen Produktangeboten. Sie sollen sich in der Industrie durchsetzen und zu strategischen Partnerschaften beitragen. Für die Nutzer ergeben sich ökonomische oder kollaborative Vorteile, zum Beispiel durch Kostenersparnis oder die Vertiefung wichtiger Partnerschaften. Zumeist werden permissive Lizenzierungen verwendet, um die Nutzung der Produkte nicht einzuschränken. Der B2B-Archetyp wird im kommerziellen Kontext weniger eingesetzt, um direkt mit dem Open Source-Produkt Einnahmen zu generieren, als vielmehr, um strategische Marktvorteile oder indirekte Gewinneinnahmen zu erzielen. So werden beispielsweise durch das Android-Betriebssystem von Google keine direkten Einnahmen generiert, stattdessen begünstigt das Betriebssystem die Nutzung weiterer Google-Produkte (zum Beispiel Kartendienste oder Suchmaschinen) und ermöglicht auf diese Weise neue datengetriebene Geschäftsmodelle, die Einnahmen generieren.

Treiber von B2B-Open-Source-Projekten sind Industrieunternehmen. Sie zielen zumeist auf die Stärkung sektorinterner Partnerschaften ab. Dementsprechend ist das Produktangebot auf Business-Kunden zugeschnitten. Typischerweise wird das Open Source-Projekt von einem Keystone-Betreiber kontrolliert, der im Allgemeinen wenig

Abbildung 9: Archetypen von Open Source-Projekten mit Beispielen aus Industrie 4.0 und darüber hinaus



Quelle: eigene Darstellung in Anlehnung an Mozilla/Open Tech Strategies 2018.

120 Vgl. Dahlander/Magnusson 2005.

121 Vgl. Mozilla/Open Tech Strategies 2018.

Wert auf Partizipation externer Entitäten legt und sich auf die Zusammenarbeit mit strategisch wichtigen Partnerorganisationen fokussiert. Durch die eher kommensalistische Beziehung zur Community gestaltet sich das Onboarding für externe Personen schwierig. Insgesamt eignet sich der B2B-Archetyp für Unternehmen, die die notwendigen Ressourcen zur Durchführung eines Open Source-Projekts aufbringen können und strategische Vorteile durch die Bereitstellung von OSS erreichen möchten.

Controlled Ecosystem

Für den Erfolg des Archetyps Controlled Ecosystem (kontrolliertes Ökosystem) ist der Aufbau einer aktiven Community entscheidend. Das Produktangebot besteht häufig aus einem Basis-System, für das externe Akteure weitere Produkte in Form von Plug-ins entwickeln können. Wie auch im Wide Open-Archetyp bietet das Controlled Ecosystem einen Mehrwert, der auf Kollaboration basiert. Jedoch ergeben sich auch ökonomische Vorteile, die beispielsweise auf Netzwerk-Effekte zurückzuführen sind. Im Gegensatz zu Wide Open werden kontrollierte Ökosysteme auch im kommerziellen Bereich eingesetzt und durch indirekte Ertragsstrategien, zum Beispiel Transaktionsgebühren, finanziert. Sowohl permissive als auch Copyleft-Lizenzen werden für die Basisarchitektur von kontrollierten Ökosystemen verwendet. Lizenzentscheidungen hängen vom weiteren Verwendungszweck der Basisarchitektur ab.

Auch hinsichtlich des Projekttreibers ist kein Merkmal dominant. Ähnlich zum Wide Open-Archetyp zielt der Controlled Ecosystem-Archetyp auf eine Vielzahl an Akteuren ab, die zum Ökosystem beitragen können. Es werden Kollaborationen mit externen Personen, Unternehmen und Forschungsinstituten, aber auch sektorinterne Partnerschaften angestrebt. Das Onboarding ist trotz dieser Offenheit als mittelschwer zu bewerten, da die zentrale Instanz bestimmte Bedingungen für eine Teilnahme vorgibt. Der Controlled Ecosystem-Archetyp bemüht sich um eine symbiotische Beziehung zur Community, da diese elementar für das Funktionieren des Open Source-Projekts ist. Trotzdem verbleibt die Entscheidungskontrolle bei der zentralen Instanz (Keystone-Akteur oder Konsortium). Sie wird auch als „wohlwollender Diktator“ mit großer Kompromissbereitschaft beschrieben.

Specialty Library

Specialty Libraries stellen hoch spezialisierte Basisfunktionen, beispielsweise in Form von Werkzeugen oder Commodities/Standardtechnologien, zur Nutzung oder Entwicklung weiterer Funktionen bereit. Dies ermöglicht Entwicklerinnen und Entwicklern, auf bereits existierenden Quellcode zurückzugreifen, sodass bei der Programmierung weiterer Funktionen Zeit und Kosten gespart werden.

Aufgrund der hohen Spezialisierung der Bibliotheken und ihrer standardisierten Funktionen haben Specialty Libraries das Potenzial, Standards durchzusetzen. Zur weiten Verbreitung der Funktionen werden permissive Lizenzen eingesetzt und restriktivere Regeln von Copyleft-Lizenzen vermieden. Die Basisfunktionen werden zumeist nicht zur direkten Ertragsgenerierung genutzt, jedoch können kommerzielle Produkte darauf aufbauend entwickelt werden.

Getrieben wird dieser Archetyp von einem Konsortium, bestehend aus spezialisierten Entwicklerinnen und Entwicklern, die Partnerschaften mit ähnlich spezialisierten Personen aus demselben Anwendungsfeld anstreben. Auch die Personen, die diese Bibliotheken nutzen, verfügen über ein gewisses Basiswissen in dem Anwendungsfeld. Es wird tendenziell wenig Community-Aufbau betrieben, da Externe ohne entsprechende Expertise keine Zielgruppe darstellen. Deshalb ist im Specialty Library-Archetyp insgesamt eine eher kommensalistische Beziehung zur Community zu beobachten. An Kontributionen werden hohe qualitative Anforderungen gestellt und durch das zentrale Kernentwicklerteam geprüft, sodass ein Onboarding schwierig ist.

Wide Open

Der Wide Open-Archetyp zeichnet sich durch die hohe Offenheit seiner Projekte aus und begrüßt aktiv Beiträge externer Quellen. Dieser Archetyp herrscht bei Open Source-Projekten vor, die über einen längeren Zeitraum ein bestimmtes Produktangebot entwickeln und stärker auf Stabilität und Zuverlässigkeit setzen als auf schnelle Produktentwicklungen. Zumeist werden Wide Open-Projekte von gemeinnützigen Organisationen oder von Individuen initiiert. Durch die hohe Anzahl an aktiv Teilnehmenden und ihre Kollaboration untereinander entstehen Communities, die das Wide Open-Projekt durch nicht monetäre Aktivitäten tragen können. Dies ermöglicht eine gewisse Flexibilität bei der Deckung der Selbstkosten des Projekts und hinsichtlich weiterer Ertragsstrategien. Charakteristisch für den Wide Open-Archetyp ist seine Offenheit gegenüber Interessierten. Begrüßt werden Beiträge verschiedenster Art (zum Beispiel Fehlerkorrekturen, Entwicklung neuer Funktionen) von externen Personen, Unternehmen oder Forschungseinrichtungen. Auch die Personen, die die Open Source-Produkte anwenden, sind ähnlich heterogen wie die Beitragenden. Dadurch ist der Wide Open-Archetyp sowohl im Business- als auch im Konsumentenbereich einsetzbar. Zumeist werden Wide Open-Projekte von einer Trägerorganisation initiiert, die eine dezentrale Entscheidungskontrolle implementiert. Entscheidungen werden meist demokratisch getroffen, manchmal ist die Einwilligung einer kontrollierenden Instanz notwendig. Aufgrund der hohen Abhängigkeit des Projekts von einer funktionierenden Community werden eine symbiotische Beziehung sowie geringe Einstiegsbarrieren in das Projekt angestrebt.

7. Open Source Software in Industrie 4.0

Dieser Abschnitt der Expertise zielt darauf ab, eine qualitative Analyse bedeutender Open Source-Projekte mit Bezug zu Industrie 4.0 vorzunehmen und relevante Merkmale entlang der im vorangegangenen Kapitel eingeführten Taxonomie zu identifizieren. Die hier beschriebenen Projekte in alphabetischer Reihenfolge sind aus einer Recherche und Untersuchung von etwa 100 Open Source-Projekten im Kontext von Industrie 4.0 ausgewählt. Betrachtet wurden hierbei ausschließlich Projekte, die bereits OSS veröffentlicht haben. Laufende Projekte, die eine Open Source-Stellung anstreben, wie FabOS, ein offenes, verteiltes, echtzeitfähiges und sicheres Betriebssystem für die Produktion, sind nicht aufgeführt.¹²²

Die Analyse von Fallbeispielen wendet die bisherigen Ergebnisse (siehe Kapitel 4 und 5) an. Zusammen mit den Ergebnissen der Expertenbefragung (siehe Kapitel 8) bilden die Ergebnisse der Fallanalysen eine Grundlage für die Ableitung von Handlungsoptionen (siehe Kapitel 10).

7.1. Übersicht relevanter Projekte

7.1.1. Apache Software Foundation und relevante Projekte

Die Apache Software Foundation (ASF) ist eine gemeinnützige Organisation zur Förderung von Open Source-Entwicklungen. Die Foundation ist aus der Entwicklung des Apache-HTTP-Servers Anfang 1999 entstanden und hat ihren Sitz in den USA. Derzeit existieren etwa 350 Projekte unter dem Dach der Foundation, davon etwa die Hälfte mit aktiver Entwicklung. Die Software wird unter den Bedingungen der Apache-Lizenz verbreitet. Die Apache-Projekte zeichnen sich durch einen gemeinschaftlichen, konsensbasierten Entwicklungsprozess aus. Jedes Projekt wird von einem selbst gewählten Team geleitet, das aktiv zum Projekt beiträgt. Die ASF ist eine Meritokratie, was bedeutet, dass die Mitgliedschaft in der Stiftung nur an Freiwillige vergeben wird, die aktiv zu Apache-Projekten beigetragen haben. Die ASF definiert einen Entwicklungs- sowie einen Entscheidungsprozess, der für alle Projekte verbindlich ist. Außerdem unterliegt die Mitarbeit einem Verhaltenskodex (Code of Conduct). Der rechtliche Schutz von Freiwilligen, die an Apache-Projekten mitarbeiten, gehört zu den primären Zielen der ASF. Charakteristisch für alle Projekte unter dem Dach der Apache Foundation ist das Präfix „Apache“ im Projektnamen.

Nachfolgend werden zwei Projekte, Apache Camel und Apache PLC4X, die unter dem Dach der ASF unter der Apache License 2.0 entwickelt werden, vorgestellt. Beide Projekte werden von der Foundation als IoT-Projekte kategorisiert. Neben diesen beiden sind als

weitere Apache Projekte mit Bezug zu Industrie 4.0 zu nennen: das Projekt Apache StreamPipes¹²³, eine Lösung für eine einfache Integration von IoT-Geräten und deren Daten, und das Projekt Apache IoTDB¹²⁴, ein Datenbank-Management-System für die Integration von Daten aus dem IoT. Das Projekt StreamPipes hat aktuell den Status „incubating“. Das bedeutet, die ASF entscheidet aktuell über die Aufnahme des Projekts.

Apache Camel

Apache Camel ist ein Open Source-Integrationsframework, das auf sogenannten Enterprise Integration Patterns aufbaut.¹²⁵ Es ermöglicht die Integration verschiedener Nachrichtenformate (zum Beispiel JSON, XML) und Kommunikationsprotokolle (zum Beispiel HTTP, MQTT) und eine regelbasierte Verarbeitung der Nachrichten und Daten. Dies erlaubt den Aufbau von Datenverarbeitungsprozessen und eine Integration in bestehende Anwendungen. Innerhalb von Apache Camel werden sechs verschiedene Komponenten entwickelt, darunter eine Bibliothek (Camel Spring Boot), die in Java-Anwendungen direkt verwendet werden kann. Auch eine Version für die direkte Ausführung auf einem Kubernetes Cluster (Apache Camel K und Apache Camel K Runtime) ist Bestandteil des Apache Camel-Projekts.¹²⁶ Apache Camel wird dem Technologiebereich „Konnektivität und Kommunikation“ zugeordnet und kann als Integrationslösung für die Vernetzung verschiedener Systeme, darunter auch Systeme und Geräte aus dem IoT, verstanden werden. Das Projekt wird seit März 2007 öffentlich auf GitHub entwickelt. Mehrere hundert Personen tragen zur Entwicklung bei. Viele der Hauptverantwortlichen arbeiten bei Red Hat.

Apache PLC4X

Apache PLC4X ist eine Sammlung von Bibliotheken zur Anbindung verschiedener speicherprogrammierbarer Steuerungen (SPS, englisch: programmable logic controller, PLC), unterschiedlicher Hersteller. Ziel des Projekts ist die Entwicklung von Lösungen für die einfache Integration von SPS ohne die Notwendigkeit einer Anpassung der SPS-Software. Die PLC4X-Bibliotheken verstehen die individuellen Protokolle (unter anderem S7) der einzelnen SPS und überführen diese in standardisierte Schnittstellen. Apache PLC4X kann an verschiedene Kommunikations- und Integrationslösungen angebunden werden, unter anderem auch an das zuvor beschriebene Apache Camel.¹²⁷

Seit Oktober 2017 wird PLC4X auf GitHub entwickelt. Insgesamt haben bisher 48 Personen an der Entwicklung mitgewirkt. Die Hauptverantwortlichen arbeiten für die beiden Unternehmen

122 Vgl. FabOS 2022.

123 Vgl. Apache StreamPipes 2019.

124 Vgl. Apache IoTDB 2022.

125 Vgl. Apache 2022.

126 Vgl. GitHub 2022c.

127 Vgl. Apache 2021a.

Tabelle 6: Einordnung von Apache Camel und PLC4X in die Open Source-Taxonomie für Industrie 4.0

	Dimension	Dimension				
Geschäftsmodell (MD ₁)	Technologiebereich (D ₁)	Konnektivität und Kommunikation (C ₁₁) ●	Prozessautomation und -flexibilisierung (C ₁₂) ○	Daten und Analytik (C ₁₃) ○	Sicherheit und Vertrauen (C ₁₄) ○	Mensch-Technik-Interaktion (C ₁₅) ○
	Produktangebot (D ₂)	Commodity (C ₂₁) ○	Werkzeug (C ₂₂) ●	Infrastruktur (C ₂₃) ○	Support und Service (C ₂₄) ○	Standards und Schnittstellen (C ₂₅) ●
	Wertversprechen (D ₃)	Funktioneller Wert (C ₃₁) ●	Ökonomischer Wert (C ₃₂) ●	Kollaborativer Wert (C ₃₃) ○	Markenwert (C ₃₄) ○	Sozialer Wert (C ₃₅) ○
	Lizenzierung (D ₄)	Duale Lizenzierung (C ₄₁) ○	Permissive Lizenzierung (C ₄₂) ●	Copyleft-Lizenzierung (C ₄₃) ○	Nicht spezifiziert (C ₄₄) ○	
	Finanzielle Ausrichtung (D ₅)	Gewinnorientiert (C ₅₁) ○		Gemeinnützig (C ₅₂) ●	Unklar (C ₅₃) ○	
	Ertragsstrategie (D ₆)	Direkte Bepreisung (C ₆₁) ○	Indirekte Bepreisung (C ₆₂) ●	Förderung (C ₆₃) ●	Nicht monetär (C ₆₄) ○	
Ökosystem (MD ₂)	Projektbetreiber (D ₇)	Getrieben durch zentralen Akteur (C ₇₁) ○		Konsortium (C ₇₂) ○	Peer-to-Peer-Netzwerk (C ₇₃) ●	
	Betreiberhintergrund (D ₈)	Industrie (C ₈₁) ●	Forschung (C ₈₂) ○	Gemeinnütziger Bereich (C ₈₃) ●	Individuum (C ₈₄) ○	
	Schlüsselpartner (D ₉)	Unternehmen (C ₉₁) ●	Forschungseinrichtung (C ₉₂) ○	Gemeinnützige Organisation (C ₉₃) ●	Individuen (C ₉₄) ○	
	Nutzer*innen (D ₁₀)	Business (C ₁₀₁) ●	Konsument*innen (C ₁₀₂) ○	Entwickler*innen (C ₁₀₃) ○	Gemeinnützige Organisation (C ₁₀₄) ○	Staat (C ₁₀₅) ○
Governance (MD ₃)	Entscheidungskontrolle (D ₁₁)	Zentral (C ₁₁₁) ○		Ausgeglichen (C ₁₁₂) ●	Dezentral (C ₁₁₃) ○	
	Community-beziehung (D ₁₂)	Symbiotische Beziehung (C ₁₂₁) ●		Kommensalistische Beziehung (C ₁₂₂) ○	Parasitäre Beziehung (C ₁₂₃) ○	
	Onboarding (D ₁₃)	Einfach (C ₁₃₁) ●		Mittel (C ₁₃₂) ○	Schwierig (C ₁₃₃) ○	

C-Ware und ConnectorIO. Beide Unternehmen bieten auch kommerziellen Support an. Das Projekt ist dem Technologiebereich „Konnektivität und Kommunikation“ zuzuordnen.

Bis zum heutigen Stand (01.2022) werden vier SPS-Protokolle (unter anderem S7, OPC-UA) nur in der Programmiersprache Java unterstützt. Weitere acht Protokolle (unter anderem EtherNet/IP) sind derzeit in Entwicklung. Unterstützung in den Programmiersprachen Go und C ist geplant.¹²⁸

7.1.2. AWS IoT Device SDKs

Mit AWS IoT bietet Amazon Web Services (AWS) einen Cloud-Dienst an, der die Verbindung von IoT-Geräten mit anderen Geräten und AWS Cloud-Diensten ermöglicht.¹²⁹ Software Development Kits (SDK) und zugehörige Schnittstellen-Dokumentationen stehen unter der Open Source-Lizenz Apache 2.0 zur Verfügung, um die Dienste in die Software auf eigenen IoT-Geräten integrieren zu können.¹³⁰ Die Software ist dem Technologiebereich „Konnektivität und Kommunikation“ zuzuordnen. Ziele des Projekts sind die einfache Verwaltung von IoT Geräten, die Anbindung an die AWS Cloud sowie die Entwicklung von Anwendungen aufbauend auf

128 Vgl. Apache 2021b.

129 Vgl. AWS 2022a.

130 Vgl. GitHub 2022d.

Tabelle 7: Einordnung von AWS IoT Device SDKs in die Open Source-Taxonomie für Industrie 4.0

Dimension		Dimension				
Geschäftsmodell (MD ₁)	Technologiebereich (D ₁)	Konnektivität und Kommunikation (C ₁₁) ○	Prozessautomation und -flexibilisierung (C ₁₂) ●	Daten und Analytik (C ₁₃) ○	Sicherheit und Vertrauen (C ₁₄) ○	Mensch-Technik-Interaktion (C ₁₅) ○
	Produktangebot (D ₂)	Commodity (C ₂₁) ○	Werkzeug (C ₂₂) ●	Infrastruktur (C ₂₃) ●	Support und Service (C ₂₄) ○	Standards und Schnittstellen (C ₂₅) ○
	Wertversprechen (D ₃)	Funktioneller Wert (C ₃₁) ●	Ökonomischer Wert (C ₃₂) ○	Kollaborativer Wert (C ₃₃) ○	Markenwert (C ₃₄) ○	Sozialer Wert (C ₃₅) ○
	Lizenzierung (D ₄)	Duale Lizenzierung (C ₄₁) ●	Permissive Lizenzierung (C ₄₂) ○	Copyleft-Lizenzierung (C ₄₃) ○	Nicht spezifiziert (C ₄₄) ○	
	Finanzielle Ausrichtung (D ₅)	Gewinnorientiert (C ₅₁) ○		Gemeinnützig (C ₅₂) ●		Unklar (C ₅₃) ○
	Ertragsstrategie (D ₆)	Direkte Bepreisung (C ₆₁) ○	Indirekte Bepreisung (C ₆₂) ○	Förderung (C ₆₃) ●	Nicht monetär (C ₆₄) ○	
Ökosystem (MD ₂)	Projektbetreiber (D ₇)	Getrieben durch zentralen Akteur (C ₇₁) ●		Konsortium (C ₇₂) ○	Peer-to-Peer-Netzwerk (C ₇₃) ○	
	Betreiberhintergrund (D ₈)	Industrie (C ₈₁) ●	Forschung (C ₈₂) ●	Gemeinnütziger Bereich (C ₈₃) ●	Individuum (C ₈₄) ○	
	Schlüsselpartner (D ₉)	Unternehmen (C ₉₁) ●	Forschungseinrichtung (C ₉₂) ●	Gemeinnützige Organisation (C ₉₃) ○	Individuen (C ₉₄) ○	
	Nutzer*innen (D ₁₀)	Business (C ₁₀₁) ●	Konsument*innen (C ₁₀₂) ○	Entwickler*innen (C ₁₀₃) ○	Gemeinnützige Organisation (C ₁₀₄) ○	Staat (C ₁₀₅) ○
Governance (MD ₃)	Entscheidungskontrolle (D ₁₁)	Zentral (C ₁₁₁) ●		Ausgeglichen (C ₁₁₂) ○	Dezentral (C ₁₁₃) ○	
	Community-beziehung (D ₁₂)	Symbiotische Beziehung (C ₁₂₁) ○		Kommensalistische Beziehung (C ₁₂₂) ●	Parasitäre Beziehung (C ₁₂₃) ○	
	Onboarding (D ₁₃)	Einfach (C ₁₃₁) ○		Mittel (C ₁₃₂) ●	Schwierig (C ₁₃₃) ○	

den Gerätedaten und Gerätefunktionen. Diese Anwendungen werden alle in der AWS Cloud ausgeführt und gespeichert. Für die Entwicklung der Anwendungen und die Anbindung der Geräte stehen die zuvor genannten SDK zur Verfügung. AWS stellt hierzu mehrere Dokumentationen zur Verfügung.¹³¹

7.1.3. BaSys 4 / Eclipse BaSyx

Das Ziel des im Juli 2016 gestarteten und vom Bundesministerium für Bildung und Forschung (BMBF) geförderten Projekts Basissystem Industrie 4.0 (BaSys 4.0) ist die Entwicklung eines Basissystems für Produktionsanlagen, das die effiziente Wandelbarkeit

eines Produktionsprozesses als zentrale Herausforderung der vierten industriellen Revolution realisiert.¹³² Das BMBF-Projekt lief bis Juni 2019. Seit Juli 2019 läuft das Folgeprojekt BaSys 4.2, ebenfalls vom BMBF gefördert. Die Arbeiten werden nach dem Auslaufen der Förderung fortgesetzt. BaSys 4 definiert eine Referenzarchitektur für die Verknüpfung und Integration von Technologien und Produktionssystemen und ist den Bereichen „Konnektivität und Kommunikation“ und „Prozessautomation und -flexibilisierung“ zuzurechnen.

Die Referenzarchitektur und deren Konzepte werden im Rahmen des Open Source-Projekts Eclipse BaSyx der Eclipse Foundation unter der Eclipse Public License 2.0 (EPL 2.0) seit September

131 Vgl. AWS 2022b.

132 Vgl. BaSys 4.0 2022a.

Tabelle 8: Einordnung von BaSyx in die Open Source-Taxonomie für Industrie 4.0

	Dimension	Dimension				
Geschäftsmodell (MD ₁)	Technologiebereich (D ₁)	Konnektivität und Kommunikation (C ₁₁) ○	Prozessautomation und -flexibilisierung (C ₁₂) ●	Daten und Analytik (C ₁₃) ○	Sicherheit und Vertrauen (C ₁₄) ○	Mensch-Technik-Interaktion (C ₁₅) ○
	Produktangebot (D ₂)	Commodity (C ₂₁) ○	Werkzeug (C ₂₂) ●	Infrastruktur (C ₂₃) ●	Support und Service (C ₂₄) ○	Standards und Schnittstellen (C ₂₅) ○
	Wertversprechen (D ₃)	Funktioneller Wert (C ₃₁) ●	Ökonomischer Wert (C ₃₂) ○	Kollaborativer Wert (C ₃₃) ○	Markenwert (C ₃₄) ○	Sozialer Wert (C ₃₅) ○
	Lizenzierung (D ₄)	Duale Lizenzierung (C ₄₁) ●	Permissive Lizenzierung (C ₄₂) ○	Copyleft-Lizenzierung (C ₄₃) ○	Nicht spezifiziert (C ₄₄) ○	
	Finanzielle Ausrichtung (D ₅)	Gewinnorientiert (C ₅₁) ○		Gemeinnützig (C ₅₂) ●	Unklar (C ₅₃) ○	
	Ertragsstrategie (D ₆)	Direkte Bepreisung (C ₆₁) ○	Indirekte Bepreisung (C ₆₂) ○	Förderung (C ₆₃) ●	Nicht monetär (C ₆₄) ○	
Ökosystem (MD ₂)	Projektbetreiber (D ₇)	Getrieben durch zentralen Akteur (C ₇₁) ●		Konsortium (C ₇₂) ○	Peer-to-Peer-Netzwerk (C ₇₃) ○	
	Betreiberhintergrund (D ₈)	Industrie (C ₈₁) ●	Forschung (C ₈₂) ●	Gemeinnütziger Bereich (C ₈₃) ●	Individuum (C ₈₄) ○	
	Schlüsselpartner (D ₉)	Unternehmen (C ₉₁) ●	Forschungseinrichtung (C ₉₂) ●	Gemeinnützige Organisation (C ₉₃) ○	Individuen (C ₉₄) ○	
	Nutzer*innen (D ₁₀)	Business (C ₁₀₁) ●	Konsument*innen (C ₁₀₂) ○	Entwickler*innen (C ₁₀₃) ○	Gemeinnützige Organisation (C ₁₀₄) ○	Staat (C ₁₀₅) ○
Governance (MD ₃)	Entscheidungskontrolle (D ₁₁)	Zentral (C ₁₁₁) ●		Ausgeglichen (C ₁₁₂) ○	Dezentral (C ₁₁₃) ○	
	Community-beziehung (D ₁₂)	Symbiotische Beziehung (C ₁₂₁) ○		Kommensalistische Beziehung (C ₁₂₂) ●	Parasitäre Beziehung (C ₁₂₃) ○	
	Onboarding (D ₁₃)	Einfach (C ₁₃₁) ○		Mittel (C ₁₃₂) ●	Schwierig (C ₁₃₃) ○	

2017 implementiert.¹³³ Insgesamt 19 Partner beteiligen sich am BaSyx-Projekt, darunter Forschungsinstitutionen, wie die Rheinisch-Westfälische Technische Hochschule Aachen und die Fraunhofer-Gesellschaft, sowie Unternehmenspartner, wie Robert Bosch GmbH, ZF Friedrichshafen AG und Festo AG & Co. KG. Besonders aktiv in der Entwicklung sind die Fraunhofer-Gesellschaft und die Robert Bosch GmbH. Das BaSys 4-Projekt nennt sechs Satellitenprojekte, in denen BaSyx beziehungsweise einzelne Komponenten weiterentwickelt werden.¹³⁴

7.1.4. Catena-X / Eclipse Tractus-X

Das Catena-X Automotive Network hat als Ziel, ein offenes Ökosystem für den effizienten und sicheren Austausch von Informationen zwischen Unternehmen der Automobilindustrie zu entwickeln. Catena-X wird nicht explizit für Industrie 4.0 entwickelt, aber oft in diesem Kontext genannt. Catena-X repräsentiert eine der Anwenderinitiativen von Gaia-X, einem europäischen Projekt von Vertreterinnen und Vertretern aus Wirtschaft, Wissenschaft und Verwaltung zum Aufbau einer leistungs- und wettbewerbsfähigen, sicheren und vertrauenswürdigen Dateninfrastruktur.

133 Vgl. Eclipse BaSyx 2017.

134 Vgl. BaSys 4.0 2022b.

Tabelle 9: Einordnung von Tractus-X in die Open Source-Taxonomie für Industrie 4.0

Dimension		Dimension				
Geschäftsmodell (MD ₁)	Technologiebereich (D ₁)	Konnektivität und Kommunikation (C ₁₁) ●	Prozessautomation und -flexibilisierung (C ₁₂) ○	Daten und Analytik (C ₁₃) ○	Sicherheit und Vertrauen (C ₁₄) ●	Mensch-Technik-Interaktion (C ₁₅) ○
	Produktangebot (D ₂)	Commodity (C ₂₁) ○	Werkzeug (C ₂₂) ○	Infrastruktur (C ₂₃) ●	Support und Service (C ₂₄) ○	Standards und Schnittstellen (C ₂₅) ○
	Wertversprechen (D ₃)	Funktioneller Wert (C ₃₁) ●	Ökonomischer Wert (C ₃₂) ○	Kollaborativer Wert (C ₃₃) ●	Markenwert (C ₃₄) ○	Sozialer Wert (C ₃₅) ○
	Lizenzierung (D ₄)	Duale Lizenzierung (C ₄₁) ○	Permissive Lizenzierung (C ₄₂) ●	Copyleft-Lizenzierung (C ₄₃) ○	Nicht spezifiziert (C ₄₄) ○	
	Finanzielle Ausrichtung (D ₅)	Gewinnorientiert (C ₅₁) ○		Gemeinnützig (C ₅₂) ●		Unklar (C ₅₃) ○
	Ertragsstrategie (D ₆)	Direkte Bepreisung (C ₆₁) ○	Indirekte Bepreisung (C ₆₂) ○	Förderung (C ₆₃) ●	Nicht monetär (C ₆₄) ○	
Ökosystem (MD ₂)	Projektbetreiber (D ₇)	Getrieben durch zentralen Akteur (C ₇₁) ○		Konsortium (C ₇₂) ●	Peer-to-Peer-Netzwerk (C ₇₃) ○	
	Betreiberhintergrund (D ₈)	Industrie (C ₈₁) ●	Forschung (C ₈₂) ○	Gemeinnütziger Bereich (C ₈₃) ○	Individuum (C ₈₄) ○	
	Schlüsselpartner (D ₉)	Unternehmen (C ₉₁) ●	Forschungseinrichtung (C ₉₂) ○	Gemeinnützige Organisation (C ₉₃) ○	Individuen (C ₉₄) ○	
	Nutzer*innen (D ₁₀)	Business (C ₁₀₁) ●	Konsument*innen (C ₁₀₂) ○	Entwickler*innen (C ₁₀₃) ○	Gemeinnützige Organisation (C ₁₀₄) ○	Staat (C ₁₀₅) ○
Governance (MD ₃)	Entscheidungskontrolle (D ₁₁)	Zentral (C ₁₁₁) ●		Ausgeglichen (C ₁₁₂) ○	Dezentral (C ₁₁₃) ○	
	Community-beziehung (D ₁₂)	Symbiotische Beziehung (C ₁₂₁) ○		Kommensalistische Beziehung (C ₁₂₂) ●	Parasitäre Beziehung (C ₁₂₃) ○	
	Onboarding (D ₁₃)	Einfach (C ₁₃₁) ○		Mittel (C ₁₃₂) ●	Schwierig (C ₁₃₃) ○	

Orientiert an den Prinzipien von Gaia-X bietet Catena-X eine branchenspezifische Datenplattform, die die Bedürfnisse der Automobilwirtschaft berücksichtigt.

Die Entwicklungen von Catena-X unter dem Dach der Eclipse Foundation erfolgen seit Juni 2021 im GitHub Repository des Eclipse-Projekts und werden als Eclipse Tractus-X unter der Apache 2.0 License (Stand 11.2021) veröffentlicht.¹³⁵ Das Projekt ist den Technologiebereichen „Konnektivität und Kommunikation“ und „Sicherheit und Vertrauen“ zuzuordnen. Die Einordnung in die Open Source-Taxonomie findet anhand von Tractus-X statt.

7.1.5. Die Eclipse Foundation und das Top Level-Projekt Internet of Things

Die Eclipse Foundation AISBL ist eine 2004 gegründete und seit 2021 in Brüssel ansässige, internationale, gemeinnützige Vereinigung (Association internationale sans but lucratif), die von mehr als 300 Mitgliedern unterstützt wird. Als eine international anerkannte Stiftung bietet sie eine Umgebung für über 400 Open Source-Projekte, darunter Bibliotheken, Tools und Frameworks für nahezu alle Technologiebereiche. Darüber hinaus betreibt und verwaltet die Eclipse Foundation die erforderliche IT-Infrastruktur für die Entwicklung von OSS. Diese umfasst unter anderem eine Versionsverwaltung, Bugtracker und Projektwebseiten. Außerdem unterstützt die Eclipse Foundation bei Aufgaben des Managements von

135 Vgl. GitHub 2021d.

Intellectual Property (IP) beispielsweise durch die Überprüfung, ob durch die im Rahmen der Eclipse Foundation entwickelte Software Rechte Dritter verletzt werden. Hierbei wird sichergestellt, dass die entwickelte Software frei von Bibliotheken/Frameworks Dritter beziehungsweise nicht kompatiblen Open Source-Produkten ist. Des Weiteren unterstützt die Foundation die Software-Entwicklung durch Bereitstellung des Eclipse Development Process, bestehend aus Grundprinzipien des Entwicklungsprozesses (unter anderem: offener und transparenter Zugang, meritokratische Verantwortungsverteilung), Anforderungen an die Community-Entwicklung und Projektorganisation (unter anderem: Projektrollen und -verantwortlichkeiten, projektinterne Abläufe, Umgang mit IP-Fragen) sowie übergeordnete Rahmenbedingungen für Projekte (unter anderem: Projektlebenszyklus, von Projektvorschlag und Inkubationsphase über Projekt-Graduation bis hin zur Archivierung/Beendigung). Der Eclipse Development Process soll eine transparente Entwicklung mit hoher Qualität sicherstellen sowie beim Aufbau und der Pflege eines Ökosystems um die entwickelte Software unterstützen.

Eclipse Internet of Things

Die Eclipse Foundation bündelt unter dem Top Level-Projekt Eclipse Internet of Things (Eclipse IoT) alle Projekte, die einen Open Source-Beitrag für industrielle IoT-Lösungen entwickeln. Das Top Level-Projekt umfasst aktuell 47 einzelne Projekte, die sich folgenden Bereichen widmen:¹³⁶

- Implementierung von Standards und Protokollen
- Entwicklung von Frameworks und Diensten, die die Komplexität von Hardware, Kommunikation und Datenstapelmanipulation abstrahieren
- Entwicklung von Tools, die zu einer Vernetzung von Systemen beitragen können
- Entwicklung von Lösungen, die auf der Grundlage der drei vorherigen Bausteine einsatzbereite Lösungen erstellen

Im Folgenden werden exemplarisch drei Projekte dieses Top Level-Projekts vorgestellt und mithilfe der Taxonomie beschrieben.

Eclipse Vorto

Das Projekt Eclipse Vorto entwickelt einen Beitrag zum digitalen Zwilling im Kontext des IoT und wird den Technologiebereichen „Prozessautomation und -flexibilisierung“ sowie „Konnektivität und Kommunikation“.¹³⁷ Das Projekt entwickelt ein Meta-Informationsmodell, ein Toolset zur Erstellung von Informationsmodellen, einen Codegenerator und ein Repository zur Verwaltung bestehender

Informationsmodelle.¹³⁸ Hierbei zielt das Produktangebot insbesondere auf Unternehmen ab. Die Schlüsselpartner des Projekts sind Unternehmen, Forschungseinrichtungen und die Eclipse Foundation. Die Anzahl der Beiträge ist rückläufig von 98 (Oktober 2020) zu 3 (September 2021). Das Projekt wird unter den permissiven Lizenzen Eclipse Distribution License 1.0 (BSD) und Eclipse Public License 1.0 bereitgestellt, sodass auch in diesem Projekt eine kommerzielle Nutzung nicht ausgeschlossen wird.¹³⁹

Das Projekt wird durch eine Organisation, die Robert Bosch GmbH, vorangetrieben. Die Entscheidungskontrolle verbleibt damit weitgehend bei Personen aus dem Unternehmensumfeld. Zwar werden Regelwerke und Governance-Strukturen von der Eclipse Foundation bereitgestellt, allerdings sind die Dokumente für einen Einstieg in die Entwicklung nur in einem geringeren Detaillierungsgrad vorhanden, sodass das Onboarding verkompliziert wird. Insgesamt lässt sich eine kommensalistische Beziehung zur Community beobachten, da die Unternehmens- beziehungsweise Projektziele im Vordergrund stehen und eine aktive Weiterentwicklung der Community nur bedingt beobachtet werden kann.

Eclipse Paho und Eclipse Mosquitto

Die beiden Eclipse-Projekte Paho und Mosquitto implementieren das Kommunikationsprotokoll Message Queuing Telemetry Transport (MQTT), das im IoT und bei der Maschine-zu-Maschine-Kommunikation eingesetzt werden kann.¹⁴⁰

Paho entwickelt eine Bibliothek von Code-Bausteinen, die es ermöglichen, das Protokoll zu implementieren und in Anwendungen, die mit MQTT kommunizieren, einzusetzen. Die Bibliothek steht in zehn verschiedenen Programmiersprachen (darunter Java, C, C++) zur Verfügung.¹⁴¹

Mosquitto implementiert einen Broker, genauer eine Kommunikationsmiddleware auf einem Server, über den Anwendungen und IoT-Geräte mittels MQTT Nachrichten veröffentlichen und abonnieren können (publish/subscribe). Mosquitto wird auch als MQTT-Broker bezeichnet und für den Aufbau von Publish/Subscribe-Kommunikation zwischen einer beliebigen Anzahl von Geräten und Diensten eingesetzt und kann direkt – ohne Anpassungen oder weitere Entwicklungen – ausgeführt werden.

Beide Projekte gehören zum Technologiebereich „Konnektivität und Kommunikation“. Sie werden vom deutschen Unternehmen Cedalo AG unter den Lizenzen Eclipse Distribution License 1.0 (BSD) und Eclipse Public License 1.0 entwickelt. Sowohl Eclipse Paho als auch Mosquitto werden bereits häufig im industriellen Kontext eingesetzt. Mosquitto steht bei vielen Linux Distributionen, unter anderem Debian, in den Standardpaketquellen zur Verfügung.¹⁴²

136 Vgl. Eclipse IoT.

137 Vgl. Eclipse Vorto 2020a.

138 Vgl. Eclipse Vorto 2020b.

139 Vgl. Eclipse Vorto 2020a.

140 Vgl. Eclipse Paho 2019, Eclipse Vorto 2020b.

141 Vgl. Eclipse Paho.

142 Vgl. Debian 2022.

Tabelle 10: Einordnung von Eclipse IoT in die Open Source-Taxonomie für Industrie 4.0

	Dimension	Dimension					
Geschäftsmodell (MD ₁)	Technologiebereich (D ₁)	Konnektivität und Kommunikation (C ₁₁) ●	Prozessautomation und -flexibilisierung (C ₁₂) ○	Daten und Analytik (C ₁₃) ○	Sicherheit und Vertrauen (C ₁₄) ○	Mensch-Technik-Interaktion (C ₁₅) ○	
	Produktangebot (D ₂)	Commodity (C ₂₁) ○	Werkzeug (C ₂₂) ●	Infrastruktur (C ₂₃) ○	Support und Service (C ₂₄) ○	Standards und Schnittstellen (C ₂₅) ●	
	Wertversprechen (D ₃)	Funktioneller Wert (C ₃₁) ●	Ökonomischer Wert (C ₃₂) ○	Kollaborativer Wert (C ₃₃) ●	Markenwert (C ₃₄) ○	Sozialer Wert (C ₃₅) ○	
	Lizenzierung (D ₄)	Duale Lizenzierung (C ₄₁) ○	Permissive Lizenzierung (C ₄₂) ●	Copyleft-Lizenzierung (C ₄₃) ○	Nicht spezifiziert (C ₄₄) ○		
	Finanzielle Ausrichtung (D ₅)	Gewinnorientiert (C ₅₁) ○		Gemeinnützig (C ₅₂) ○		Unklar (C ₅₃) ●	
	Ertragsstrategie (D ₆)	Direkte Bepreisung (C ₆₁) ○	Indirekte Bepreisung (C ₆₂) ○	Förderung (C ₆₃) ●	Nicht monetär (C ₆₄) ○		
Ökosystem (MD ₂)	Projektbetreiber (D ₇)	Getrieben durch zentralen Akteur (C ₇₁) ○		Konsortium (C ₇₂) ●	Peer-to-Peer-Netzwerk (C ₇₃) ○		
	Betreiberhintergrund (D ₈)	Industrie (C ₈₁) ●	Forschung (C ₈₂) ○	Gemeinnütziger Bereich (C ₈₃) ●	Individuum (C ₈₄) ○		
	Schlüsselpartner (D ₉)	Unternehmen (C ₉₁) ●	Forschungseinrichtung (C ₉₂) ○	Gemeinnützige Organisation (C ₉₃) ○	Individuen (C ₉₄) ○		
	Nutzer*innen (D ₁₀)	Business (C ₁₀₁) ●	Konsument*innen (C ₁₀₂) ○	Entwickler*innen (C ₁₀₃) ○	Gemeinnützige Organisation (C ₁₀₄) ○	Staat (C ₁₀₅) ○	
Governance (MD ₃)	Entscheidungskontrolle (D ₁₁)	Zentral (C ₁₁₁) ●		Ausgeglichen (C ₁₁₂) ○	Dezentral (C ₁₁₃) ○		
	Community-beziehung (D ₁₂)	Symbiotische Beziehung (C ₁₂₁) ●		Kommensalistische Beziehung (C ₁₂₂) ●	Parasitäre Beziehung (C ₁₂₃) ○		
	Onboarding (D ₁₃)	Einfach (C ₁₃₁) ○		Mittel (C ₁₃₂) ●	Schwierig (C ₁₃₃) ○		

7.1.6. FIWARE

Das Ziel von FIWARE ist die Bereitstellung einer standardisierten Open Source-Plattform für die Entwicklung neuer Internetdienste und -anwendungen. Das FIWARE-Projekt wurde 2011 gestartet und erhielt verschiedene EU-Förderungen. Die gesamte Plattform soll als eine lizenzfreie, eigenständige Open Source Cloud-Plattform für künftige intelligente Städte dienen – auch, um Abhängigkeiten von kommerzieller Software und damit verbundene Lizenzgebühren zu vermeiden.¹⁴³

Kern der FIWARE-Plattform sind die sogenannten FIWARE Generic Enabler, die aus Open Source-Komponenten bestehen und einen entsprechenden Standard erfüllen müssen. Die zentrale Einheit ist der sogenannte FIWARE ContextBroker Orion.

Die Ausrichtung von FIWARE wird durch einen technischen Lenkungsausschuss bestimmt und die FIWARE-Komponenten werden auf GitHub unter verschiedenen Lizenzen (MIT, AGPL-3.0) als Open Source entwickelt.¹⁴⁴ Technologisch sind die einzelnen Softwarekomponenten den Bereichen „Konnektivität und Kommunikation“ und/oder „Daten und Analytik“ zuzuordnen.

Tabelle 11: Einordnung von FIWARE in die Open Source-Taxonomie für Industrie 4.0

Dimension		Dimension				
Geschäftsmodell (MD ₁)	Technologiebereich (D ₁)	Konnektivität und Kommunikation (C ₁₁) ●	Prozessautomation und -flexibilisierung (C ₁₂) ○	Daten und Analytik (C ₁₃) ●	Sicherheit und Vertrauen (C ₁₄) ○	Mensch-Technik-Interaktion (C ₁₅) ○
	Produktangebot (D ₂)	Commodity (C ₂₁) ○	Werkzeug (C ₂₂) ○	Infrastruktur (C ₂₃) ●	Support und Service (C ₂₄) ○	Standards und Schnittstellen (C ₂₅) ●
	Wertversprechen (D ₃)	Funktioneller Wert (C ₃₁) ●	Ökonomischer Wert (C ₃₂) ●	Kollaborativer Wert (C ₃₃) ●	Markenwert (C ₃₄) ○	Sozialer Wert (C ₃₅) ○
	Lizenzierung (D ₄)	Duale Lizenzierung (C ₄₁) ○	Permissive Lizenzierung (C ₄₂) ○	Copyleft-Lizenzierung (C ₄₃) ○	Nicht spezifiziert (C ₄₄) ●	
	Finanzielle Ausrichtung (D ₅)	Gewinnorientiert (C ₅₁) ○		Gemeinnützig (C ₅₂) ●		Unklar (C ₅₃) ○
	Ertragsstrategie (D ₆)	Direkte Bepreisung (C ₆₁) ○	Indirekte Bepreisung (C ₆₂) ○	Förderung (C ₆₃) ●	Nicht monetär (C ₆₄) ○	
Ökosystem (MD ₂)	Projektbetreiber (D ₇)	Getrieben durch zentralen Akteur (C ₇₁) ○		Konsortium (C ₇₂) ○	Peer-to-Peer-Netzwerk (C ₇₃) ○	
	Betreiberhintergrund (D ₈)	Industrie (C ₈₁) ●	Forschung (C ₈₂) ○	Gemeinnütziger Bereich (C ₈₃) ○		Individuum (C ₈₄) ○
	Schlüsselpartner (D ₉)	Unternehmen (C ₉₁) ●	Forschungseinrichtung (C ₉₂) ●	Gemeinnützige Organisation (C ₉₃) ●		Individuen (C ₉₄) ○
	Nutzer*innen (D ₁₀)	Business (C ₁₀₁) ●	Konsument*innen (C ₁₀₂) ○	Entwickler*innen (C ₁₀₃) ○	Gemeinnützige Organisation (C ₁₀₄) ○	Staat (C ₁₀₅) ●
Governance (MD ₃)	Entscheidungskontrolle (D ₁₁)	Zentral (C ₁₁₁) ●		Ausgeglichen (C ₁₁₂) ○	Dezentral (C ₁₁₃) ○	
	Community-beziehung (D ₁₂)	Symbiotische Beziehung (C ₁₂₁) ○		Kommensalistische Beziehung (C ₁₂₂) ●	Parasitäre Beziehung (C ₁₂₃) ○	
	Onboarding (D ₁₃)	Einfach (C ₁₃₁) ○		Mittel (C ₁₃₂) ○	Schwierig (C ₁₃₃) ●	

143 Vgl. FIWARE 2022.

144 Vgl. GitHub 2022e.

7.1.7. IndustryFusion

Mit IndustryFusion wird eine herstellerübergreifende Open Source-Lösung zur Vernetzung von intelligenten Fabriken (Smart Factories) und smarten Produkten (Smart Products) entwickelt.¹⁴⁵ Das Konsortium um IndustryFusion wird primär durch deutsche Unternehmen geprägt. IndustryFusion fällt in den Technologiebereich

„Konnektivität und Kommunikation“, wobei primäre Motivation das Anbieten einer Lösung für den Bereich „Prozessautomation und -flexibilisierung“ ist. Die Hauptkomponente wird seit Januar 2021 öffentlich entwickelt und steht unter der Apache 2.0 Lizenz zur Verfügung. Die Entwicklung erfolgt primär durch das deutsche Unternehmen iteratec GmbH.

Tabelle 12: Einordnung von IndustryFusion in die Open Source-Taxonomie für Industrie 4.0

Dimension		Dimension				
Geschäftsmodell (MD ₁)	Technologiebereich (D ₁)	Konnektivität und Kommunikation (C ₁₁) ●	Prozessautomation und -flexibilisierung (C ₁₂) ○	Daten und Analytik (C ₁₃) ○	Sicherheit und Vertrauen (C ₁₄) ○	Mensch-Technik-Interaktion (C ₁₅) ○
	Produktangebot (D ₂)	Commodity (C ₂₁) ○	Werkzeug (C ₂₂) ○	Infrastruktur (C ₂₃) ●	Support und Service (C ₂₄) ○	Standards und Schnittstellen (C ₂₅) ●
	Wertversprechen (D ₃)	Funktioneller Wert (C ₃₁) ●	Ökonomischer Wert (C ₃₂) ●	Kollaborativer Wert (C ₃₃) ○	Markenwert (C ₃₄) ●	Sozialer Wert (C ₃₅) ○
	Lizenzierung (D ₄)	Duale Lizenzierung (C ₄₁) ○	Permissive Lizenzierung (C ₄₂) ●	Copyleft-Lizenzierung (C ₄₃) ○	Nicht spezifiziert (C ₄₄) ○	
	Finanzielle Ausrichtung (D ₅)	Gewinnorientiert (C ₅₁) ○		Gemeinnützig (C ₅₂) ●		Unklar (C ₅₃) ○
	Ertragsstrategie (D ₆)	Direkte Bepreisung (C ₆₁) ○	Indirekte Bepreisung (C ₆₂) ○	Förderung (C ₆₃) ○	Nicht monetär (C ₆₄) ●	
Ökosystem (MD ₂)	Projektbetreiber (D ₇)	Getrieben durch zentralen Akteur (C ₇₁) ○		Konsortium (C ₇₂) ●	Peer-to-Peer-Netzwerk (C ₇₃) ○	
	Betreiberhintergrund (D ₈)	Industrie (C ₈₁) ●	Forschung (C ₈₂) ●	Gemeinnütziger Bereich (C ₈₃) ●		Individuum (C ₈₄) ○
	Schlüsselpartner (D ₉)	Unternehmen (C ₉₁) ●	Forschungseinrichtung (C ₉₂) ●	Gemeinnützige Organisation (C ₉₃) ●		Individuen (C ₉₄) ○
	Nutzer*innen (D ₁₀)	Business (C ₁₀₁) ●	Konsument*innen (C ₁₀₂) ○	Entwickler*innen (C ₁₀₃) ○	Gemeinnützige Organisation (C ₁₀₄) ○	
Governance (MD ₃)	Entscheidungskontrolle (D ₁₁)	Zentral (C ₁₁₁) ●		Ausgeglichen (C ₁₁₂) ○		Dezentral (C ₁₁₃) ○
	Community-beziehung (D ₁₂)	Symbiotische Beziehung (C ₁₂₁) ○		Kommensalistische Beziehung (C ₁₂₂) ●		Parasitäre Beziehung (C ₁₂₃) ○
	Onboarding (D ₁₃)	Einfach (C ₁₃₁) ○		Mittel (C ₁₃₂) ●		Schwierig (C ₁₃₃) ○

145 Vgl. GitHub 2022f, IndustryFusion 2022.

7.1.8. International Data Spaces

Die Initiative International Data Spaces (IDS) hat zum Ziel, ein Referenzarchitekturmodell und zugehörige Referenzimplementierungen für industrielle Datenräume (englisch: Data Spaces) zu entwerfen. Die Basis bilden die sogenannten Konnektoren. Wenn auch nicht explizit für Industrie 4.0 entwickelt, lassen sich diese im Kontext von Industrie 4.0 verwenden. Die Konnektoren ermöglichen den Aufbau von sicheren Verbindungen für den Austausch von Daten und erlauben, die Datennutzung zu kontrollieren. Die Initiative hat sich Grundprinzipien gesetzt. Eines davon ist der Erhalt der Souveränität über die eigenen Daten. Dieses Prinzip schließt

die Übertragung von Eigentumsrechten an zentrale Instanzen oder Anbieter aus. Die IDS-Entwicklungen sind quelloffen und werden aktuell auf GitHub unter Koordination der International Data Spaces Association unter der Apache 2.0 License veröffentlicht.¹⁴⁶

Die Entwicklung des Eclipse Dataspace Connector erfolgt seit einem Jahr unter dem Dach der Eclipse Foundation und mit den Partnern Microsoft, Daimler, SAP, BMW, Amadeus und der Fraunhofer Gesellschaft (Stand März 2022).¹⁴⁷ Der Eclipse Data Space Connector steht ebenfalls unter der Apache 2.0 License. Die Data Space-Konnektoren sind den beiden Technologiebereichen „Konnektivität und Kommunikation“ und „Sicherheit und Vertrauen“ zuzuordnen.

Tabelle 13: Einordnung des Eclipse Data Space Connector der Initiative International Data Spaces in die Open Source-Taxonomie für Industrie 4.0

Dimension	Dimension					
Geschäftsmodell (MD ₁)	Technologiebereich (D ₁)	Konnektivität und Kommunikation (C ₁₁) ●	Prozessautomation und -flexibilisierung (C ₁₂) ○	Daten und Analytik (C ₁₃) ●	Sicherheit und Vertrauen (C ₁₄) ○	Mensch-Technik-Interaktion (C ₁₅) ○
	Produktangebot (D ₂)	Commodity (C ₂₁) ○	Werkzeug (C ₂₂) ○	Infrastruktur (C ₂₃) ○	Support und Service (C ₂₄) ○	Standards und Schnittstellen (C ₂₅) ●
	Wertversprechen (D ₃)	Funktioneller Wert (C ₃₁) ●	Ökonomischer Wert (C ₃₂) ●	Kollaborativer Wert (C ₃₃) ○	Markenwert (C ₃₄) ○	Sozialer Wert (C ₃₅) ○
	Lizenzierung (D ₄)	Duale Lizenzierung (C ₄₁) ○	Permissive Lizenzierung (C ₄₂) ●	Copyleft-Lizenzierung (C ₄₃) ○	Nicht spezifiziert (C ₄₄) ○	
	Finanzielle Ausrichtung (D ₅)	Gewinnorientiert (C ₅₁) ○		Gemeinnützig (C ₅₂) ●		Unklar (C ₅₃) ○
	Ertragsstrategie (D ₆)	Direkte Bepreisung (C ₆₁) ○	Indirekte Bepreisung (C ₆₂) ○	Förderung (C ₆₃) ●	Nicht monetär (C ₆₄) ○	
Ökosystem (MD ₂)	Projektbetreiber (D ₇)	Getrieben durch zentralen Akteur (C ₇₁) ○		Konsortium (C ₇₂) ●	Peer-to-Peer-Netzwerk (C ₇₃) ○	
	Betreiberhintergrund (D ₈)	Industrie (C ₈₁) ●	Forschung (C ₈₂) ●	Gemeinnütziger Bereich (C ₈₃) ○	Individuum (C ₈₄) ○	
	Schlüsselpartner (D ₉)	Unternehmen (C ₉₁) ●	Forschungseinrichtung (C ₉₂) ●	Gemeinnützige Organisation (C ₉₃) ○	Individuen (C ₉₄) ○	
	Nutzer*innen (D ₁₀)	Business (C ₁₀₁) ●	Konsument*innen (C ₁₀₂) ○	Entwickler*innen (C ₁₀₃) ●	Gemeinnützige Organisation (C ₁₀₄) ○	Staat (C ₁₀₅) ○
Governance (MD ₃)	Entscheidungskontrolle (D ₁₁)	Zentral (C ₁₁₁) ○		Ausgeglichen (C ₁₁₂) ●	Dezentral (C ₁₁₃) ○	
	Community-beziehung (D ₁₂)	Symbiotische Beziehung (C ₁₂₁) ●		Kommensalistische Beziehung (C ₁₂₂) ○	Parasitäre Beziehung (C ₁₂₃) ○	
	Onboarding (D ₁₃)	Einfach (C ₁₃₁) ○		Mittel (C ₁₃₂) ●	Schwierig (C ₁₃₃) ○	

146 Vgl. GitHub 2022g.

147 Vgl. Eclipse Dataspace Connector 2022.

7.1.9. open62541

open62541 ist eine Open Source-Implementierung des in der International Electrotechnical Commission Serie IEC 62541 standardisierten OPC-UA-Protokolls für die industrielle Kommunikation.¹⁴⁸ Der Standard wird durch Forschungseinrichtungen (Fraunhofer-Gesellschaft, Rheinisch-Westfälische Technische Hochschule Aachen, Technische Universität Dresden, Landesforschungsinstitut des Freistaats Bayern für softwareintensive Systeme) seit dem Jahr 2014 in der Programmiersprache C entwickelt und unter der Mozilla

Public License 2.0 veröffentlicht. Insgesamt haben seit 2014 193 Personen an der Entwicklung im GitHub-Projekt mitgewirkt.¹⁴⁹ Des Weiteren lieferten sechs Industrieunternehmen Beiträge. Kommerzieller Support kann von den beiden Unternehmen Kalycito und basysKom GmbH bezogen werden. Die Entwicklung steht außerdem als Bibliothek (libopen62541) für die Anwendungsentwicklung zur Verfügung. Das Projekt stellt Komponenten zum Herunterladen bereit und wird in den Technologiebereich „Konnektivität und Kommunikation“ eingeordnet.

Tabelle 14: Einordnung von open62541 in die Open Source-Taxonomie für Industrie 4.0

Dimension		Dimension				
Geschäftsmodell (MD ₁)	Technologiebereich (D ₁)	Konnektivität und Kommunikation (C ₁₁) ●	Prozessautomation und -flexibilisierung (C ₁₂) ○	Daten und Analytik (C ₁₃) ○	Sicherheit und Vertrauen (C ₁₄) ○	Mensch-Technik-Interaktion (C ₁₅) ○
	Produktangebot (D ₂)	Commodity (C ₂₁) ○	Werkzeug (C ₂₂) ●	Infrastruktur (C ₂₃) ○	Support und Service (C ₂₄) ○	Standards und Schnittstellen (C ₂₅) ○
	Wertversprechen (D ₃)	Funktioneller Wert (C ₃₁) ●	Ökonomischer Wert (C ₃₂) ○	Kollaborativer Wert (C ₃₃) ○	Markenwert (C ₃₄) ○	Sozialer Wert (C ₃₅) ○
	Lizenzierung (D ₄)	Duale Lizenzierung (C ₄₁) ○	Permissive Lizenzierung (C ₄₂) ○	Copyleft-Lizenzierung (C ₄₃) ○	Nicht spezifiziert (C ₄₄) ○	
	Finanzielle Ausrichtung (D ₅)	Gewinnorientiert (C ₅₁) ○		Gemeinnützig (C ₅₂) ●	Unklar (C ₅₃) ○	
	Ertragsstrategie (D ₆)	Direkte Bepreisung (C ₆₁) ○	Indirekte Bepreisung (C ₆₂) ○	Förderung (C ₆₃) ●	Nicht monetär (C ₆₄) ○	
Ökosystem (MD ₂)	Projektbetreiber (D ₇)	Getrieben durch zentralen Akteur (C ₇₁) ○		Konsortium (C ₇₂) ●	Peer-to-Peer-Netzwerk (C ₇₃) ○	
	Betreiberhintergrund (D ₈)	Industrie (C ₈₁) ○	Forschung (C ₈₂) ●	Gemeinnütziger Bereich (C ₈₃) ○	Individuum (C ₈₄) ○	
	Schlüsselpartner (D ₉)	Unternehmen (C ₉₁) ●	Forschungseinrichtung (C ₉₂) ●	Gemeinnützige Organisation (C ₉₃) ○	Individuen (C ₉₄) ○	
	Nutzer*innen (D ₁₀)	Business (C ₁₀₁) ●	Konsument*innen (C ₁₀₂) ○	Entwickler*innen (C ₁₀₃) ○	Gemeinnützige Organisation (C ₁₀₄) ○	Staat (C ₁₀₅) ○
Governance (MD ₃)	Entscheidungskontrolle (D ₁₁)	Zentral (C ₁₁₁) ○		Ausgeglichen (C ₁₁₂) ●	Dezentral (C ₁₁₃) ○	
	Community-beziehung (D ₁₂)	Symbiotische Beziehung (C ₁₂₁) ●		Kommensalistische Beziehung (C ₁₂₂) ○	Parasitäre Beziehung (C ₁₂₃) ○	
	Onboarding (D ₁₃)	Einfach (C ₁₃₁) ●		Mittel (C ₁₃₂) ○	Schwierig (C ₁₃₃) ○	

148 Vgl. GitHub 2021e, Open62541 2022.

149 Vgl. ebd.

7.1.10. Open Manufacturing Platform

Die Open Manufacturing Platform (OMP) ist eine Initiative innerhalb der Linux Foundation, um die Entwicklungen von Industrie 4.0-Lösungen für das produzierende Gewerbe zu beschleunigen.¹⁵⁰ Gegründet 2019 von Microsoft und der BMW Group, haben sich mittlerweile über 30 Unternehmen in dieser Allianz zusammengeschlossen. In Arbeitsgruppen organisiert werden Herausforderungen wie IoT-Konnektivität und die semantische Datenmodellierung bearbeitet. Die OMP treibt vor allem die Erstellung technischer Spezifikationen und

die Nutzung offener Industriestandards voran, entwickelt aber auch Software, beispielsweise den OMP OPC UA Edge Connector, der seit August 2021 öffentlich entwickelt wird.¹⁵¹ Gelöst werden soll mit dieser Entwicklung das Problem fehlender industrietauglicher Konnektoren für den Datenaustauschstandard OPC Unified Architecture. Mithilfe des OPC UA Edge Connector können über diesen Standard kommunizierende Geräte an IT-Systeme angebunden werden, die diesen Standard nicht unterstützen, sondern über Formate wie MQTT oder Kafka kommunizieren. Das Projekt ist dem Technologiebereich „Konnektivität und Kommunikation“ zuzuordnen.

Tabelle 15: Einordnung der Open Manufacturing Platform in die Open Source-Taxonomie für Industrie 4.0

Dimension		Dimension				
Geschäftsmodell (MD ₁)	Technologiebereich (D ₁)	Konnektivität und Kommunikation (C ₁₁) ●	Prozessautomation und -flexibilisierung (C ₁₂) ○	Daten und Analytik (C ₁₃) ○	Sicherheit und Vertrauen (C ₁₄) ○	Mensch-Technik-Interaktion (C ₁₅) ○
	Produktangebot (D ₂)	Commodity (C ₂₁) ○	Werkzeug (C ₂₂) ●	Infrastruktur (C ₂₃) ○	Support und Service (C ₂₄) ○	Standards und Schnittstellen (C ₂₅) ○
	Wertversprechen (D ₃)	Funktioneller Wert (C ₃₁) ●	Ökonomischer Wert (C ₃₂) ○	Kollaborativer Wert (C ₃₃) ○	Markenwert (C ₃₄) ○	Sozialer Wert (C ₃₅) ○
	Lizenzierung (D ₄)	Duale Lizenzierung (C ₄₁) ○	Permissive Lizenzierung (C ₄₂) ●	Copyleft-Lizenzierung (C ₄₃) ○	Nicht spezifiziert (C ₄₄) ○	
	Finanzielle Ausrichtung (D ₅)	Gewinnorientiert (C ₅₁) ○		Gemeinnützig (C ₅₂) ●		Unklar (C ₅₃) ○
	Ertragsstrategie (D ₆)	Direkte Bepreisung (C ₆₁) ○	Indirekte Bepreisung (C ₆₂) ○	Förderung (C ₆₃) ●	Nicht monetär (C ₆₄) ○	
Ökosystem (MD ₂)	Projektbetreiber (D ₇)	Getrieben durch zentralen Akteur (C ₇₁) ○		Konsortium (C ₇₂) ●	Peer-to-Peer-Netzwerk (C ₇₃) ○	
	Betreiberhintergrund (D ₈)	Industrie (C ₈₁) ●	Forschung (C ₈₂) ○	Gemeinnütziger Bereich (C ₈₃) ○		Individuum (C ₈₄) ○
	Schlüsselpartner (D ₉)	Unternehmen (C ₉₁) ●	Forschungseinrichtung (C ₉₂) ○	Gemeinnützige Organisation (C ₉₃) ○		Individuen (C ₉₄) ○
	Nutzer*innen (D ₁₀)	Business (C ₁₀₁) ●	Konsument*innen (C ₁₀₂) ○	Entwickler*innen (C ₁₀₃) ○	Gemeinnützige Organisation (C ₁₀₄) ○	
Governance (MD ₃)	Entscheidungskontrolle (D ₁₁)	Zentral (C ₁₁₁) ●		Ausgeglichen (C ₁₁₂) ○	Dezentral (C ₁₁₃) ○	
	Community-beziehung (D ₁₂)	Symbiotische Beziehung (C ₁₂₁) ○		Kommensalistische Beziehung (C ₁₂₂) ●	Parasitäre Beziehung (C ₁₂₃) ○	
	Onboarding (D ₁₃)	Einfach (C ₁₃₁) ○		Mittel (C ₁₃₂) ●	Schwierig (C ₁₃₃) ○	

150 Vgl. Open Manufacturing Platform 2020.

151 Vgl. GitHub 2021f.

7.1.11. Open Source Automation Development Lab

Das Open Source Automation Development Lab (OSADL) ist ein genossenschaftlich organisiertes Kompetenznetzwerk, das die Pflege und Anpassung von OSS für den industriellen Einsatz zum Ziel hat. Mitglieder des Netzwerks schließen sich in OSADL-Projekten zusammen, um so die Entwicklung bestimmter OSS zu unterstützen. Neben Projekten, die sich um Rechtsfragen drehen oder mit Compliance-Aspekten befassen, werden auch Projekte unterstützt, in denen OSS entwickelt wird. So finanziert OSADL über Mitgliedsbeiträge unter anderem die Entwicklung und Pflege der Realtime-Fähigkeiten des Linux-Kernels durch Anpassungen des PREEMPT_RT Patches.¹⁵² Mit diesen Anpassungen kann der Linux-Kernel als Betriebssystemkern eines Echtzeitbetriebssystems genutzt werden, wodurch es

möglich ist, Echtzeitanforderungen von Anwendungen zu erfüllen. Ebenso finanziert OSADL die Erweiterung der OPC-UA-Open-Source-Implementierung open62541 um die zwei Protokollergänzungen Publish (Pub) und Subscribe (Sub).¹⁵³ Diese Mechanismen werden verwendet, um Nachrichten gleichzeitig an eine Reihe von Empfängern zu senden beziehungsweise um sich auf bestimmte Ereignisse zu registrieren, wobei entsprechende Nachrichten beim Auftreten des Ereignisses an die Abonnenten gesendet werden. Die stattfindende Kommunikation wird über den in der Entwicklung befindlichen Time Sensitive Networking (TSN) Standard abgesichert. Das Projekt OPC UA PubSub over TSN kann dem Technologiebereich „Konnektivität und Kommunikation“ zugeordnet werden. Die Entwicklungsergebnisse beider genannter Projekte fließen direkt in die jeweiligen Hauptprojekte ein, die sie erweitern.

Tabelle 16: Einordnung des Open Source Automation Development Labs in die Open Source-Taxonomie für Industrie 4.0

Dimension		Dimension					
Geschäftsmodell (MD ₁)	Technologiebereich (D ₁)	Konnektivität und Kommunikation (C ₁₁) ○	Prozessautomation und -flexibilisierung (C ₁₂) ●	Daten und Analytik (C ₁₃) ○	Sicherheit und Vertrauen (C ₁₄) ○	Mensch-Technik-Interaktion (C ₁₅) ○	
	Produktangebot (D ₂)	Commodity (C ₂₁) ○	Werkzeug (C ₂₂) ○	Infrastruktur (C ₂₃) ○	Support und Service (C ₂₄) ●	Standards und Schnittstellen (C ₂₅) ●	
	Wertversprechen (D ₃)	Funktioneller Wert (C ₃₁) ○	Ökonomischer Wert (C ₃₂) ○	Kollaborativer Wert (C ₃₃) ●	Markenwert (C ₃₄) ○	Sozialer Wert (C ₃₅) ○	
	Lizenzierung (D ₄)	Duale Lizenzierung (C ₄₁) ○	Permissive Lizenzierung (C ₄₂) ○	Copyleft-Lizenzierung (C ₄₃) ○	Nicht spezifiziert (C ₄₄) ●		
	Finanzielle Ausrichtung (D ₅)	Gewinnorientiert (C ₅₁) ●		Gemeinnützig (C ₅₂) ○	Unklar (C ₅₃) ○		
	Ertragsstrategie (D ₆)	Direkte Bepreisung (C ₆₁) ○	Indirekte Bepreisung (C ₆₂) ○	Förderung (C ₆₃) ●	Nicht monetär (C ₆₄) ○		
Ökosystem (MD ₂)	Projektbetreiber (D ₇)	Getrieben durch zentralen Akteur (C ₇₁) ○		Konsortium (C ₇₂) ●	Peer-to-Peer-Netzwerk (C ₇₃) ○		
	Betreiberhintergrund (D ₈)	Industrie (C ₈₁) ●	Forschung (C ₈₂) ●	Gemeinnütziger Bereich (C ₈₃) ○	Individuum (C ₈₄) ○		
	Schlüsselpartner (D ₉)	Unternehmen (C ₉₁) ●	Forschungseinrichtung (C ₉₂) ●	Gemeinnützige Organisation (C ₉₃) ○	Individuen (C ₉₄) ○		
	Nutzer*innen (D ₁₀)	Business (C ₁₀₁) ●	Konsument*innen (C ₁₀₂) ○	Entwickler*innen (C ₁₀₃) ○	Gemeinnützige Organisation (C ₁₀₄) ○	Staat (C ₁₀₅) ○	
Governance (MD ₃)	Entscheidungskontrolle (D ₁₁)	Zentral (C ₁₁₁) ●		Ausgeglichen (C ₁₁₂) ○	Dezentral (C ₁₁₃) ○		
	Community-beziehung (D ₁₂)	Symbiotische Beziehung (C ₁₂₁) ○		Kommensalistische Beziehung (C ₁₂₂) ●	Parasitäre Beziehung (C ₁₂₃) ○		
	Onboarding (D ₁₃)	Einfach (C ₁₃₁) ○		Mittel (C ₁₃₂) ○	Schwierig (C ₁₃₃) ●		

152 Vgl. OSADL 2021a.

153 Vgl. OSADL 2021b.

7.1.12. Robotic Operating System

Das seit 2007 entwickelte Robotic Operating System (ROS) umfasst mehrere Bibliotheken, die als Grundlage für die Entwicklung von Robotikanwendungen verwendet werden können. Damit fällt ROS in den Technologiebereich „Prozessautomation und -flexibilisierung“. Neun Projekte, die ROS einsetzen, werden öffentlich vorgestellt.¹⁵⁴ Darüber hinaus existieren jedoch weitere Projekte, die ROS einsetzen. Die Entwicklung erfolgt unter Verwendung der permissiven BSD-Lizenz, sodass wenige Restriktionen bezüglich der Software gesetzt werden und auch kommerzielle Produkte möglich sind.

ROS wird von mehr als tausend Personen, die in Industrieunternehmen oder Forschungseinrichtungen angestellt sind, unter dem

Dach der Open Source Robotics Foundation mit Sitz in den USA vorangetrieben. Hauptnutzer der Bibliotheken sind Unternehmen sowie Entwicklerinnen und Entwickler. Über die Open Source Robotics Corporation, ein Tochterunternehmen der Foundation, werden die Projekte finanziert, wobei auch individuelle Spenden zur Unterstützung beitragen. Das ROS-Projekt wird durch ein Peer-to-Peer-Netzwerk vorangetrieben, dessen Governance auf einer dezentralen und demokratischen Entscheidungsfindung basiert. Prinzipiell können alle an der Entwicklung beteiligten Personen die Rolle eines Core Maintainers einnehmen, wenn sie sich durch hohe Beteiligung in Form von Code-Kontributionen oder Reviews hervorheben. Core Maintainer erhalten mehr Rechte, zum Beispiel Zugang zur Modifikation von Schlüsselkomponenten, und verpflichten sich gleichzeitig, dafür zu sorgen, dass das Projekt vorangetrieben wird. Weiterhin wird die demokratische Struktur durch Foren und

Tabelle 17: Einordnung von ROS in die Open Source-Taxonomie für Industrie 4.0

Dimension		Dimension					
Geschäftsmodell (MD ₁)	Technologiebereich (D ₁)	Konnektivität und Kommunikation (C ₁₁) ○	Prozessautomation und -flexibilisierung (C ₁₂) ●	Daten und Analytik (C ₁₃) ○	Sicherheit und Vertrauen (C ₁₄) ○	Mensch-Technik-Interaktion (C ₁₅) ○	
	Produktangebot (D ₂)	Commodity (C ₂₁) ○	Werkzeug (C ₂₂) ●	Infrastruktur (C ₂₃) ●	Support und Service (C ₂₄) ○	Standards und Schnittstellen (C ₂₅) ○	
	Wertversprechen (D ₃)	Funktioneller Wert (C ₃₁) ●	Ökonomischer Wert (C ₃₂) ○	Kollaborativer Wert (C ₃₃) ●	Markenwert (C ₃₄) ○	Sozialer Wert (C ₃₅) ○	
	Lizenzierung (D ₄)	Duale Lizenzierung (C ₄₁) ○	Permissive Lizenzierung (C ₄₂) ●	Copyleft-Lizenzierung (C ₄₃) ○	Nicht spezifiziert (C ₄₄) ○		
	Finanzielle Ausrichtung (D ₅)	Gewinnorientiert (C ₅₁) ○		Gemeinnützig (C ₅₂) ●	Unklar (C ₅₃) ○		
	Ertragsstrategie (D ₆)	Direkte Bepreisung (C ₆₁) ○	Indirekte Bepreisung (C ₆₂) ○	Förderung (C ₆₃) ●	Nicht monetär (C ₆₄) ○		
Ökosystem (MD ₂)	Projektbetreiber (D ₇)	Getrieben durch zentralen Akteur (C ₇₁) ○		Konsortium (C ₇₂) ○	Peer-to-Peer-Netzwerk (C ₇₃) ●		
	Betreiberhintergrund (D ₈)	Industrie (C ₈₁) ○	Forschung (C ₈₂) ○	Gemeinnütziger Bereich (C ₈₃) ●	Individuum (C ₈₄) ●		
	Schlüsselpartner (D ₉)	Unternehmen (C ₉₁) ●	Forschungseinrichtung (C ₉₂) ●	Gemeinnützige Organisation (C ₉₃) ●	Individuen (C ₉₄) ●		
	Nutzer*innen (D ₁₀)	Business (C ₁₀₁) ●	Konsument*innen (C ₁₀₂) ○	Entwickler*innen (C ₁₀₃) ●	Gemeinnützige Organisation (C ₁₀₄) ○	Staat (C ₁₀₅) ○	
Governance (MD ₃)	Entscheidungskontrolle (D ₁₁)	Zentral (C ₁₁₁) ○		Ausgeglichene (C ₁₁₂) ○	Dezentral (C ₁₁₃) ●		
	Community-Beziehung (D ₁₂)	Symbiotische Beziehung (C ₁₂₁) ●		Kommensalistische Beziehung (C ₁₂₂) ○	Parasitäre Beziehung (C ₁₂₃) ○		
	Onboarding (D ₁₃)	Einfach (C ₁₃₁) ●		Mittel (C ₁₃₂) ○	Schwierig (C ₁₃₃) ○		

154 Vgl. Open Robotics 2021.

-diskussionen gestützt. Das Onboarding wird durch eine ausführliche Dokumentation und Diskussionsforen erleichtert. Durch die symbiotische Beziehung ergibt sich ein funktionierendes Ökosystem, sodass ROS neben seinem funktionellen Wert auch einen kollaborativen Mehrwert bietet. ROS ist ein gutes Beispiel dafür, dass im Open Source-Bereich nicht nur das eigentliche Produkt zählt, sondern auch die Entwicklung einer funktionierenden Community zentral ist. Die Bereitstellung von Kommunikationskanälen (Foren, Dokumentation, E-Mail-Verteiler), aber auch die bewusste Abgabe gewisser Entscheidungsrechte an die Community können zu einem gesunden Ökosystem beitragen.¹⁵⁵

7.2. Bewertung des Stands von Open Source im Kontext von Industrie 4.0

Die vorgestellten OSS-Projekte stellen nur eine Auswahl dar, geben jedoch den allgemeinen Stand von Open Source im Kontext von Industrie 4.0 treffend wieder. Deutlich wird, dass lediglich eine kleine Zahl von Projekten einen klaren Industrie 4.0-Fokus haben, beispielsweise Eclipse Vorto oder open62541. Viel größer ist die Zahl der Projekte, deren Ergebnisse Lösungsbausteine (unter anderem Frameworks und Bibliotheken) oder Tools für die Software-Entwicklung im Industrie 4.0-Kontext darstellen. So sind beispielsweise ROS, die Projekte der Apache Foundation und auch die Mehrheit des Eclipse Top-Level-IoT-Projekts nicht entwickelt worden, um ausschließlich im Kontext von Industrie 4.0 eingesetzt zu werden. Noch deutlicher wird dies anhand des Linux-Kernels, der sich auf einer

Vielzahl an eingebetteten Systemen findet, somit auch in Industrie 4.0. Speziell für den Einsatz in Industrie 4.0 entwickelt wurde er nicht und ist dennoch ein wesentlicher Lösungsbaustein.

Unter den wenigen Projekten mit klarer Industrie 4.0-Motivation lässt sich nur eine geringe Anzahl mehr als einem Technologiebereich zuordnen. Allerdings fällt die Zuordnung zu den Technologiebereichen auch nicht immer leicht. Ein Projekt, das unverkennbar zwei Technologiebereiche tangiert, ist BaSyx, das „Konnektivität und Kommunikation“ explizit für „Prozessautomation und -flexibilisierung“ implementiert. Projekte, die mehr als zwei Technologiebereiche ansprechen und damit klar technologieübergreifende Software-Lösungen für Industrie 4.0 bereitstellen, sind den Autorinnen und Autoren nicht bekannt.

Die vorgestellte Auswahl an OSS-Projekten umfasst nur aktiv gepflegte Projekte. Abschließend muss jedoch angemerkt werden, dass man immer wieder auch auf inaktive Projekte stößt. Solche Projekte sollten im Kontext von Industrie 4.0 nur mit großer Vorsicht eingesetzt werden, da keine Weiterentwicklung erfolgt und nicht auf die Unterstützung einer Entwicklungs-Community zurückgegriffen werden kann. Unter anderem bedeutet dies auch, dass Sicherheitslücken eigenständig behoben werden müssten, wenn sie bekannt werden. Unter den inaktiven Projekten finden sich immer wieder auch die Entwicklungsergebnisse ehemals öffentlich geförderter Forschungsprojekte, die nach Auslaufen der Finanzierung nicht weiter gepflegt werden, da der Aufbau einer Community nicht forciert wurde oder nicht gelungen ist.

8. Expertenperspektiven: Chancen und Herausforderungen von Open Source Software im Kontext von Industrie 4.0

Neben der Literaturlauswertung und Analyse bestehender Industrie 4.0-Open-Source-Projekte auf Basis der erstellten Taxonomie wurden im Rahmen der vorliegenden Expertise 22 Expertinnen und Experten zu ihrer Einschätzung von OSS im Kontext von Industrie 4.0 befragt. Die ausgewählten Personen wurden im Zeitraum von September bis November 2021 in circa einstündigen, virtuell durchgeführten Terminen von je zwei Mitgliedern des Forschungsteams mithilfe eines Leitfadens, der das Gespräch grob strukturierte, interviewt. Die Expertinnen und Experten sind allesamt mit der Thematik von OSS vertraut. Vertreten sind folgende Perspektiven: Entwicklungs-Community, Standardisierungsorganisation, unternehmerisches Management und Forschungsinstitution. Mithilfe der Befragung von Expertinnen und Experten, die Open Source aktiv nutzen, zu Open Source-Projekten beitragen, selbst Projekte initiieren oder Open Source auf politischer beziehungsweise unternehmerischer Ebene fördern, wurden Best Practices und etablierte Erfahrungswerte im Umgang mit OSS erhoben. Sie stellen das zentrale Ergebnis der vorliegenden Expertise dar.

Im Folgenden werden die wesentlichen Erkenntnisse und Kernaussagen aus den 22 Interviews zusammengefasst. In den Interviews wurden dabei folgende Schwerpunkte gesetzt:

8.1. Bedeutung, aktuelle Trends und Eignungsbereiche von Open Source Software im Kontext von Industrie 4.0

Die befragten Expertinnen und Experten nutzen OSS und tragen aktiv zur Weiterentwicklung von Open Source-Projekten bei. Dieser Beitrag äußert sich sowohl in der Bereitstellung von Entwicklungskapazitäten, als auch in der finanziellen Förderung von Projekten. Letzteres ist insbesondere bei großen Unternehmen der Fall. Auffällig ist in diesem Zusammenhang, dass die interviewten Personen aus Start-ups zwar betonen, dass sie ohne OSS ihre Geschäftsmodelle nicht realisieren könnten, aufgrund der wenigen verfügbaren Ressourcen ein „Zurückgeben“ an die Community jedoch herausfordernd finden. Die Start-ups stehen im Spannungsfeld der Entwicklung und Etablierung ihres zentralen geistigen Eigentums und der Frage, wie und in welchem Umfang diese Entwicklungen mit der Community geteilt werden sollen. Der Wunsch, etwas zurückzugeben, könnte perspektivisch dem Geschäftsmodell dienen. Im Hinblick auf Nutzungsoptionen und Vorteile von OSS bestätigen die Befragten in ihren Äußerungen die in den vorangegangenen Analysen bereits identifizierten Mehrwerte.



„Open Source ist in erster Linie ein **Lizenz- und Kooperationsmodell**. Als solches birgt es viele Vorteile und lässt sich nicht auf einzelne Technologiebereiche beschränken.“

*Rafael Laguna de la Vera, Direktor der Bundesagentur für Sprunginnovationen SPRIND
Bildrechte: SPRIND GmbH*



„Als wir uns entschieden haben, unsere Software quelloffen bereitzustellen, hat dies zu einer Verbesserung der **Code-Qualität** geführt. Aber das war und ist kein Selbstläufer, denn durch die Lizenz alleine verändert sich der Code nicht und natürlich gibt es auch schlechten offenen, genauso wie schlechten proprietären Code.“

„Unseren Sourcecode zu veröffentlichen war eine **unternehmensstrategische Entscheidung**. Gepaart mit einer Adaption unserer **Geschäftsmodelle** hat es uns Märkte eröffnet, die wir sonst aufgrund unserer Unternehmensgröße und **namhafter Konkurrenz** viel schwerer erreicht hätten.“

*Peter Ganten, CEO Univention GmbH und Vorsitzender des Vorstands der Open Source Business Alliance
– Bundesverband digitale Souveränität e. V.
Bildrechte: Univention GmbH*

Die Befragten sehen Open Source primär als Kollaborationsmodell, das erlaubt, Ideen und Entwicklungen zu teilen und gemeinsam mit Partnern oder externen Entwicklerinnen und Entwicklern voranzutreiben. Dabei schätzen die Expertinnen und Experten auch den Nebeneffekt, dass bestimmte Vorgehensmodelle und Arbeitsweisen der Software-Entwicklung innerhalb der Open Source Community gefestigt werden. Insofern trägt Open Source auch zu einer verbesserten Organisation und strukturierteren Durchführung von Entwicklungsvorhaben bei und kann deshalb auch als **Entwicklungsmodell** von Software verstanden werden.

Kollaborationen und gemeinschaftliche Entwicklungen sind dabei nicht auf bestimmte Industrie 4.0-Bereiche beschränkt. Die Expertinnen und Experten nutzen in ihren jeweiligen Branchen und Anwendungsdomänen verschiedene OSS. Sie sehen eine Vielzahl unterschiedlicher Trends im Bereich Industrie 4.0. Als Beispiele wurden unter anderem Software Stacks auf Geräten, Softwareprotokolle sowie Sensorkommunikation über entsprechende Protokolle genannt. Ferner wurden Referenzimplementierungen und Verwaltungsschalen angeführt. Ebenso wurden Smart und Open Data Models als aufstrebender Trend identifiziert. Neben den doch eher allgemein gehaltenen Funktionen und Basisanwendungen wurde des Weiteren vor allem der Bereich der Fertigungsautomatisierung genannt. Der Bereich der Autonomisierung ist dagegen, nach Ansicht der Expertinnen und Experten, noch unterentwickelt. Als weitere Einsatzbereiche, die sich vor allem gut für die unternehmensübergreifende Zusammenarbeit eignen, nannten die Befragten: Kryptographie, Vertrauens- und Identitätsmanagement sowie Entwicklungen im Software-Infrastrukturbereich. Einen ausgeprägten Trend sehen die Expertinnen und Experten aktuell in der Open Hardware-Entwicklung. Nicht nur Code wird immer öfter quelloffen bereitgestellt, sondern zunehmend auch Open Hardware verfügbar gemacht. Hier betonen die Expertinnen und Experten den Wunsch nach Unabhängigkeit und Souveränität und sehen die Gefahr von Lock-in-Effekten.

Grundsätzlich hängt die Eignung von OSS laut den Befragten weniger vom Technologiefeld ab als von der Frage, ob eine kritische Masse von Personen an der jeweiligen Entwicklung interessiert ist. Hier zeigt sich noch einmal deutlich, dass OSS grundsätzlich technologieneutral ist und von den Expertinnen und Experten in erster Hinsicht als effektive und effiziente Kollaborationsmethode, als leitendes Entwicklungsmodell und erst im letzten Schritt als strategisches Element im Geschäftsmodell von Unternehmen betrachtet wird. Open Source als Teil der Unternehmensstrategie

eignet sich unter anderem dann, so die Expertinnen und Experten, wenn die Langlebigkeit von Entwicklungen gewährleistet werden soll, zum Beispiel um die firmenunabhängige Wartung langlebiger Softwareprodukte in der Luftfahrt sicherzustellen. Als konkrete Entwicklungen, die sich für OSS-Projekte eignen, nennen die Befragten vor allem Software-Komponenten, die Basisfunktionen abdecken. Ebenso eignet sich ihrer Ansicht nach die gemeinschaftliche Entwicklung dort, wo Daten ausgetauscht, Schnittstellen geschaffen und Sicherheit und Vertrauen in die Entwicklung gestärkt werden sollen. Entsprechend könnten auch Ressourcen gebündelt und gemeinschaftliche Standardlösungen für die jeweilige Branche entwickelt werden. Die Grundlage für eine erfolgreiche gemeinschaftliche Entwicklung stellt laut den Expertinnen und Experten vor allem das passende Ökosystem dar. Im Folgenden werden Aspekte genannt und skizziert, die aus Sicht der Befragten Attraktivität und Nachhaltigkeit von OSS fördern, indem sie zur Gesundheit von Open Source Communities und ihren Ökosystemen beitragen.

8.2. Erfolgsfaktoren und Attraktivität von Open Source-Ökosystemen und Entwicklungs-Communities

Die Community und das Ökosystem, das sich im Rahmen einer Projektentwicklung und Fortführung formt, sind laut den befragten Expertinnen und Experten entscheidend für den Erfolg einer Open Source-Lösung. Dabei betonen sie, dass es schwierig ist, aktive Communities aufzubauen. Ist dies jedoch geglückt, so die Expertinnen und Experten, sind auch die Projekte oftmals von hoher Qualität und Langlebigkeit.

Als wesentliche Merkmale erfolgreicher beziehungsweise attraktiver Communities und Projekte nennen die Expertinnen und Experten unter anderem die Art der Kommunikation beziehungsweise die Geschwindigkeit und Professionalität, mit der auf Anfragen oder Änderungswünsche etc. reagiert wird. Zu den weiteren Merkmalen gehören allgemeingültige Aspekte, wie eine gute Dokumentation der Software-Entwicklung und der damit verbundenen Prozesse, ferner geringe Eintrittshürden und eine klare Erkennbarkeit des Projektzwecks und Ziels, zum Beispiel kommuniziert über entsprechende Projekt-Webseiten. Auch die Aktualität der Projekte und die Intensität der Aktivitäten sind für die Expertinnen und Experten wichtige Indikatoren für die Kraft und Reife der Community. Ein weiteres Gütemerkmal ist, wenn die Beitragenden nicht alle aus derselben Organisation kommen. Erfolgen nur Beiträge durch ein



„ **Building vibrant communities** is a task that must be seriously tackled. You need an office; you need a clear champion who is responsible. You need to be responsive and communicate openly. Do not let issues unsolved for a long period – just to mention a few. **All this takes time, money and effort.**“

Frédéric Desbiens, Program Manager IoT and Edge Computing Eclipse Foundation

Unternehmen, könnte dies, so die Expertinnen und Experten, zu Abhängigkeiten führen oder die Attraktivität des Projekts schmälern, die mit zunehmender Diversität der Community steigt. Daher bevorzugen die Befragten vor allem breit und divers aufgestellte Communities. Ist jedoch das Ziel, eine sehr spezifische Entwicklung effizient voranzutreiben, kann eine stärker fachlich getriebene und weniger diverse Community von Vorteil sein. Insofern beschreiben die Befragten das Spannungsfeld zwischen schnellen Entwicklungserfolgen und konsensorientierten Mehrheiten. Als weiteres Spannungsfeld wird gesehen, dass große Konzerne in meritokratisch organisierten Communities ihr Selbstverständnis als einflussnehmender Geldgeber nur schlecht realisieren können und deshalb mitunter von Finanzierungen absehen. Grundsätzlich gilt für die Befragten, dass die Community über dem Code steht („Community over Code“). Dieser Leitsatz bringt zum Ausdruck, dass eine gute Community entscheidender für den nachhaltigen Erfolg eines Projekts ist als die Qualität des Codes. Ferner betonen die Befragten, dass sie von den etablierten Stiftungen (Foundations) lernen können und die dortigen Governance-Strukturen als Blaupausen für den Aufbau eigener Communities dienen sollten. Darüber hinaus wurde angeführt, dass es sinnvoll sein kann, eigene Projekte unter der Schirmherrschaft bestehender Stiftungen zu entwickeln, um so von den bereits existierenden Strukturen und Communities zu profitieren. Ebenso bestätigen die Expertinnen und Experten mehrheitlich die in Kapitel 5.3 genannten Messgrößen und Einflussfaktoren für gesunde Open Source-Projekte und Open Source-Ökosysteme.

8.3. Strategische Gründe für Open Source und ihre Relevanz in Abhängigkeit von der Unternehmensgröße

Die Expertinnen und Experten wurden nicht nur gefragt, welche Bedeutung OSS auf technischer Ebene für Industrie 4.0 hat, sondern auch, welche strategische Rolle Open Source in und für Unternehmen und deren Geschäftsmodelle spielen kann. Je nachdem, ob es sich um Start-ups, KMU oder Konzerne handelt, kann sich die strategische Bedeutung von Open Source unterscheiden.

Im Hinblick auf die Realisierung von Geschäftsmodellen auf Basis oder im Zusammenhang mit OSS stellen einige Expertinnen und Experten fest, dass vor allem große, monopolistische Konzerne

erfolgreich Geschäftsmodelle auf OSS aufbauen konnten, obwohl Open Source im öffentlichen Bereich (zum Beispiel in der kommunalen Verwaltung) einen viel besseren Ruf genießt. Die Nutzung und Verbreitung von Open Source im industriellen Bereich insgesamt halten viele Expertinnen und Experten noch für ausbaufähig. Gleichzeitig weisen einige auf einen Widerspruch hin: Trotz des Erfolgs, den Konzerne mit OSS-basierten Geschäftsmodellen erreichen konnten, sind Open Source-Lösungen dort, wo sich starke, monopolistische Anbieter etabliert haben, besonders im Softwarebereich, kaum vertreten. Zudem sehen die Expertinnen und Experten, dass vor allem Anbieter von Lösungen aus dem Bereich Maschinen- und Anlagenbau vor einer Vielzahl von Herausforderungen stehen: Auf der einen Seite verstärkt sich ihre Abhängigkeit von internationalen Softwareanbietern, auf der anderen Seite gerät zum Beispiel im Maschinenbau das eigentliche Kerngeschäft immer stärker unter Druck. Derzeit, so die Einschätzung, können sich Maschinen- und Anlagenbauer noch nicht über die Bereitstellung eigener Software-Lösungen differenzieren, während gleichzeitig die Konstruktion von Anlagen und Hardware nicht mehr alleinige beziehungsweise ausreichende Kernkompetenz ist. Daher stehen Branchen wie der Maschinen- und Anlagenbau unter einem massiven Transformationsdruck.

Obwohl es den Unternehmen schwerfällt, aus eigener Kraft innovative Software-Lösungen am Markt zu etablieren, sind sie nach Ansicht einiger Expertinnen und Experten zu zögerlich bei der Öffnung für Open Source und die gemeinschaftliche Entwicklung. Darin sehen sie ein Hemmnis für die Etablierung deutscher oder europäischer Open Source-Lösungen am Markt sowie für die Digitalisierung im Allgemeinen und Industrie 4.0 „Made in Germany“ im Speziellen. Nach Ansicht der Expertinnen und Experten bremst die Sorge vor der Öffnung und Weitergabe des geschäftsdifferenzierenden geistigen Eigentums. Es dominiert die Furcht, dass internationale Wettbewerber diese Offenheit ausnutzen könnten, um ihre Marktstellung zu forcieren. Das Dilemma, bestehend aus fehlenden eigenen Ressourcen oder Marktmacht für die Etablierung innovativer Soft- und Hardwarelösungen einerseits und fehlendem Mut, Open Source als Entwicklungs- und Kooperationsmodell einzusetzen, andererseits, sehen die Expertinnen und Experten als dringend aufzulösen. In diesem Kontext wünschen sie sich europäische Souveränität – nicht im Sinne von Abschottung und dem Aufbau eigener monopolistischer Strukturen, sondern im Sinne von Diversität,



„ Gerade zu Beginn tun sich Start-ups häufig schwer, eine aktive Rolle in **Open-Source-Communitys** zu spielen. **IP-Aufbau** und die eigenen Produkte skalierfähig zu machen, stehen im Vordergrund.“

„ **Start-ups** im Allgemeinen und auch wir als MotionMiners im Speziellen profitieren von Open-Source-Software. Nicht **bei null zu starten** hat unsere Anwendungs- und KI-Entwicklungen enorm beschleunigt.“

*Dr.-Ing. Sascha Feldhorst, CEO und Co-Founder MotionMiners GmbH
Bildrechte: MotionMiners GmbH*



„ Gerade im **mittelständischen Maschinenbau** gibt es noch **viel Potenzial** für den Einsatz von Open Source. Damit hier entsprechende Projekte gemeinschaftlich als Open Source realisiert werden können, hilft es, wenn sich die **Beteiligten schon im Vorfeld kennen, vertrauen** und für sich eine gemeinsame Aufgabenstellung identifiziert haben.“

*Dr.-Ing. Konstantin Kernschmidt, Technical Lead Industry Fusion Foundation
Bildrechte: Dr.-Ing. Konstantin Kernschmidt*



„ Open Source ist ein **deutlicher Motivationstreiber** für viele unserer Entwickler. Allein schon aus diesem Grund ist es **für Arbeitgeber zunehmend wichtig**, Open Source strategisch anzugehen.“

Anonym, Konzern (Software)

Wettbewerb und Anbietervielfalt. Sie empfehlen daher eine Öffnung in den Bereichen, die nicht wettbewerbsdifferenzierend sind, aber durch Gemeinschaftsentwicklungen einen Mehrwert für die Branche liefern können.

Besonders KMU könnten nach Ansicht der Expertinnen und Experten von gemeinschaftlichen Open Source-Entwicklungen, aber auch von deren Nutzung profitieren. Bei KMU, die Schwierigkeiten bei der digitalen Transformation erleben, kann der Einsatz von OSS unterstützen. Weniger innovative Unternehmen können mit OSS einen risikoarmen und einfachen Zugang zu neuen Technologien erschließen. Die Expertinnen und Experten sehen den Vorteil, dass quelloffene einfacher und schneller Proof of Concepts entwickelt werden können. Um das Potenzial von Open Source zu erschließen, sehen sie vor allem in konservativen Unternehmen, die bisher das Prinzip der Abgrenzung und internen Innovationsentwicklung verfolgen, die Notwendigkeit eines Umdenkens hin zu einer partizipativen Innovationskultur und einer Kultur der Öffnung und des Teilens. Ziel muss nach Ansicht der Befragten sein, Open Source nicht nur zu nutzen, sondern aktiv OSS der Community zur Verfügung zu stellen. Sie sehen als Gewinner einer solchen Öffnung unter anderem kleine Anbieter im Maschinenbau, die von der Vernetzung ihrer Produkte auf Software-Ebene profitieren könnten. Wünschenswert wäre eine erhöhte Kompatibilität und Standardisierung, damit sich die Realisierung von digitalen Geschäftsmodellen vereinfacht. Hierbei kann, so die Expertinnen und Experten, die gemeinschaftliche Entwicklung von OSS-Lösungen einen wertvollen Beitrag leisten. Als Herausforderung gerade für KMU sehen die Befragten neben dem Schließen von Kenntnislücken das Gewinnen von IT-Fachkräften. Angemerkt wurde, dass gerade der Einsatz von OSS zur Steigerung der Attraktivität eines Unternehmens als Arbeitgeber im Bereich Informatik führen kann. Programmiererinnen und Programmierer, so erleben es die Expertinnen und Experten, bewerten die Mitwirkung an Open Source-Projekten im Rahmen ihres Arbeitsverhältnisses als sehr positiv.

Auch die befragten Personen aus dem Bereich der Start-ups bestätigen, dass Open Source eine strategisch bedeutsame Rolle für ihre Unternehmen spielt. Oft ist OSS die Grundlage, auf der sie ihre Geschäftsmodelle aufbauen. Aus diesem Grund ist die freie Verfügbarkeit und der Zugang zu Software entscheidend für den Unternehmenserfolg.

Strategisch scheinen derzeit vor allem große Unternehmen Open Source entdeckt zu haben und in ihre Unternehmensstrategien zu integrieren. Gründe dafür sind einerseits der damit verbundene Kostenvorteil, zum anderen ist mit der Unterstützung von Communities – sei es finanziell oder personell – die Hoffnung auf Mitgestaltung aktueller technologischer Trends verbunden. Eher große Unternehmen als KMU nutzen das OSPO-Konzept, um Open Source strategisch innerhalb des Unternehmens voranzutreiben und Open Source-Prozesse strategisch auszurichten. KMU sehen für sich jedoch den Vorteil, über kürzere und flexiblere Entscheidungswege OSS-Projekte schneller realisieren zu können.

8.4. Bedeutung von Lizenzfragen

In den Gesprächen mit den Expertinnen und Experten wurde um eine Einschätzung im Hinblick auf verschiedene OSS-Lizenzen und ihren Einfluss auf Geschäftsmodelle, die Entwicklungszusammenarbeit und die Nutzung der bereitgestellten Projekte gebeten. Die Mehrheit der Expertinnen und Experten gab an, vor allem permissive Lizenzen wie Apache 2.0 zu bevorzugen. Aus ihrer Sicht sind Copyleft-Lizenzen unattraktiv und befinden sich aufgrund der zunehmenden Verbreitung der Cloud-Technologie auf dem Rückzug. Die Befragten sehen die Möglichkeit, durch ein Cloud-basiertes Geschäftsmodell Lizenzfragen zu umgehen: Die Verwendung von gehosteter Software stellt rechtlich eine interne Verwendung dar und keine Weitergabe der Software. Die Sorge vor Lizenzverletzungen verbunden mit Rechtsstreitigkeiten schreckt nach Ansicht der Befragten Unternehmen ab, Projekte oder Entwicklungen zu



„ Auch Software, die unter einer **Copyleft-Lizenz** wie der GPL steht, kann im industriellen Umfeld eingesetzt werden und sollte **nicht per se ausgeschlossen** werden.“

Dr. Till Jaeger, Partner JBB Rechtsanwälte



„ Wir haben Open Source zu einer **strategischen Managementaufgabe** gemacht und arbeiten daran, das volle Potenzial für Daimler zu erschließen. Ein wichtiger Schritt war dabei der Aufbau eines professionellen und interdisziplinären **Open Source Program Office** als zentrale Anlaufstelle und zur Bündelung aller Open Source-Aktivitäten.“

Jonas von Malottki, Joint Priority Lead FOSS Daimler AG

verwenden, die unter einer Copyleft-Lizenz stehen. Aus dem juristischen Bereich wurde geäußert, dass die meisten Rechtsstreite im Bereich von Linux beziehungsweise GPL-Lizenzen geführt werden. Open Source Compliance-Verstöße betreffen nicht nur die Verwendung von OSS im eigenen Unternehmen, sondern auch den Einkauf über Zulieferer. Vereinzelt gab es jedoch auch Fürsprecher der Copyleft-Lizenz, die unter anderem dann als sinnvoll erachtet wird, wenn das Geschäftsmodell an eine duale Lizenzierung geknüpft ist, um zum Beispiel verschiedene Zielgruppen anzusprechen.

Die befragten Expertinnen und Experten sind sich jedoch in einem Punkt einig: Der „Lizenzdschungel“ stellt ein Hemmnis für die Verwendung und Bereitstellung von OSS-Lösungen dar. Insbesondere KMU oder Start-ups ohne eigene juristische Expertise oder externen rechtlichen Beistand könnten bei der Bewertung von Lizenzfragen schnell überfordert sein. Folglich geht die Nutzung und vor allem die Bereitstellung eigener Software unter einer Open Source-Lizenz mit großen Unsicherheiten einher. Als Herausforderung für KMU wird weiterhin gesehen, dass eine Prüfung der Lizenzkompatibilitäten zum Beispiel bei der eigenen Bereitstellung von OSS sich oftmals nicht realisieren lässt. Dies führt unter anderem dazu, dass Unternehmen die Verwendung von Copyleft-Lizenzen pauschal ausschließen. Größere Unternehmen verfügen laut der Befragten in der Regel über die entsprechenden Kompetenzen, die durch OSPO-Prozesse oder interne Rechtsabteilungen abgedeckt sind. Die Expertinnen und Experten merken an, dass zwar oftmals auf fachlicher Ebene an Standardisierungen gearbeitet wird, Standardisierungsbemühungen im lizenzrechtlichen Bereich jedoch träge sind. Positiv bewertet werden wichtige Zusammenschlüsse wie die Genossenschaft Open Source Automation Development Lab (OSADL), die neben dem Ziel, OSS-Entwicklungen im Maschinen- und Anlagenbau zu koordinieren und voranzutreiben, auch das Ziel verfolgt, Transparenz und Rechtssicherheit in Bezug auf die Verwendung von Lizenzen zu unterstützen und somit auch standardisierte Vorgehensweisen zu etablieren. Als Problem wird gesehen,

dass große Unternehmen oftmals gehemmt sind, solchen Organisationen beziehungsweise Vereinigungen beizutreten. Die Befragten sehen generelle Vorbehalte seitens der juristischen Abteilungen großer Unternehmen gegenüber Mitgliedschaften in Organisationen, Vereinen oder Genossenschaften. Es fehlt hier Vertrauen in die rechtlichen Bewertungen anderer Organisationen, so beispielsweise bezüglich der Bewertung der Kompatibilität unterschiedlicher Open Source-Lizenzen.

Im Hinblick auf eigene Geschäftsmodelle spielen Lizenzfragen bei den meisten Unternehmen eine eher untergeordnete Rolle. Die Expertinnen und Experten sind sich einig, dass Open Source weniger ein Geschäftsmodell als vielmehr ein Kooperations- und Entwicklungsmodell ist. Nur wenige Unternehmen streben nach Ansicht der Befragten beispielsweise das Geschäftsmodell der dualen Lizenzierung an. Größer wird die Bedeutung von OSS-basierten Geschäftsmodellen im Bereich Services und Support gesehen. Dennoch hemmen auch hier lizenzrechtliche Fragen neben Sicherheitsbedenken die Verbreitung von OSS.

8.5. Open Source Software im Kontext von Standardisierung

Die Expertinnen und Experten vertreten unterschiedliche Industriezweige beziehungsweise öffentliche Einrichtungen und Institutionen, die allesamt in ihren jeweiligen Bereichen von Standardisierungen profitieren können. Die Befragten waren sich einig, dass OSS einen wichtigen Beitrag leisten kann, um De-facto-Standards zu etablieren. Auch stellt OSS, wie in Abschnitt 4.2.3 dargestellt, eine wichtige Ergänzung zu De-jure-Standards dar. Insbesondere die Befragten aus der Industrie betonen, dass Open Source als Referenzimplementierung dazu beitragen kann, bestimmte Standards in die Praxis zu überführen. Auch scheint die Bedeutung von Open Source bei Standardisierungsorganisationen angekommen zu



„Das Potenzial von Open Source Software für Industrie 4.0 ist riesig. Es ist ein unverzichtbarer Baustein, um **Standards – seien es Datenmodelle oder Schnittstellen** – deutlich schneller in die Anwendung zu bringen. Open Source-Implementierungen helfen bei der Verbreitung, der Akzeptanz und einer **De-facto-Standardisierung**. Nur so erreichen wir wirkliche Interoperabilität und eine **Reduktion von Integrationsaufwänden**.“

Ulrich Ahle, CEO FIWARE Foundation, e. V.

sein, auch wenn diese Open Source tendenziell nicht als Enabler für De-jure-Standards sehen. Die Expertinnen und Experten betonen, dass sich De-facto-Standards oder Referenzimplementierungen auf Basis von OSS einer höheren Akzeptanz erfreuen als De-jure-Standards. Dies liegt ihrer Ansicht nach an der gemeinschaftlichen, community-getriebenen Entwicklung, die zwar mitunter zeitintensiv ist, aber letztendlich zu einer breiten Anwendung und Diffusion beiträgt und damit auch zu einer De-facto-Standardisierung führen kann.

Als Profiteure von De-facto-Standards sehen die Befragten vor allem Start-ups und junge Unternehmen, da sie ihre Geschäftsmodelle und die damit verbundenen Prozesse wesentlich nachhaltiger ausrichten können. Aus fachlicher Sicht eignen sich besonders Basisentwicklungen und -komponenten, Datenmodelle sowie Schnittstellen für Standardisierungen. Die Vorteile von einheitlichen Schnittstellen und Komponenten bestehen darin, dass Unternehmen einfacher neue Kundensegmente erschließen oder in Kooperation mit anderen Partnern und Stakeholdern gemeinsame Wertversprechen realisieren können. Aus unternehmerischer Sicht kann die Bereitstellung eigener OSS auch einen strategischen Beitrag zur Standardisierung darstellen: Unternehmen, die nachgefragtes geistiges Eigentum schon heute Open Source stellen, können sich einen Early Mover-Vorteil sichern, wenn sich durch die Bereitstellung eigenen Codes ein De-facto-Standard etablieren lässt, der die eigene Position am Markt festigt. Dies gelingt zum Beispiel, indem komplementierende Anbieter auf bereitgestellte Schnittstellen zurückgreifen und durch positive Netzwerkeffekte den De-facto-Standard stärken.

Als großes Hemmnis für De-facto-Standardisierungen und die Verbreitung entsprechender Implementierungen sehen die Expertinnen und Experten vorhandene Patente. Somit bestätigen sie die negative Auswirkung von Patenten auf De-facto-Standardisierungen im Industriekontext.

8.6. Bedenken gegen den Einsatz von Open Source Software

Die Organisationen der Befragten sind mehrheitlich Nutzer beziehungsweise Bereitsteller von OSS. Dennoch wurden die Expertinnen und Experten auch zu ihrer Einschätzung der häufigsten Bedenken gegen den Einsatz von OSS befragt. Als wesentliche Hemmnisse für die Nutzung von OSS oder Beteiligung an Open Source-Projekten

nennen die Befragten, wie bereits dargestellt, lizenzrechtliche Fragestellungen und mangelnden rechtlichen Support. Lizenzrechtliche Bedenken verhindern besonders bei komplexen Produkten mit einer Vielzahl an Gewerken und Modulen die Nutzung von OSS beziehungsweise die Veröffentlichung von Quellcode. Als weitere potenzielle Einwände gegen Open Source nennen die Befragten die Befürchtung, dass sich Lizenzen über die Zeit ändern könnten oder die genutzte Software nicht nachhaltig beziehungsweise langfristig entwickelt wird. Außerdem hemmen nach Ansicht der Expertinnen und Experten fehlender Support und fehlende Unterstützungsdienstleistungen im Bereich der OSS. Möglicherweise wird häufig nur deshalb auf proprietäre Systeme gesetzt, weil der Service durch IT-Dienstleister hier gewährleistet scheint. Sie empfehlen deshalb, besonders KMU, die wenig technologisiert sind, gezielt bei der Verwendung von Open Source-Komponenten zu unterstützen.

Die Befragten nennen als weiteres potenzielles Hemmnis die Sorge, eigene Entwicklungsergebnisse beziehungsweise eigene Leistungen, die einer Community frei zur Verfügung gestellt werden, könnten ausgenutzt werden. Ängste existieren zum einen vor wenig reziproken Ökosystemen und dem „Free-Rider-Problem“ (Trittbrettfahrer), zum anderen vor dem Abgreifen von Know-how und erfolgreichem Kommerzialisieren fremder Leistungen. Die Expertinnen und Experten berichten, dass kleine Unternehmen und Start-ups zwar die Chance einer breiten Diffusion ihrer Entwicklungen durch OSS sehen, jedoch häufig befürchten, dass Unternehmen mit einer höheren Marktmacht oder Bekanntheit die Entwicklungsergebnisse oder das offengelegte geistige Eigentum kommerziell ausnutzen könnten, sodass der Urheber der Software „leer“ ausgeht. Gleichzeitig schildern die Expertinnen und Experten, dass die Durchsetzung einer Open Source-Lösung in Unternehmen auch am fehlenden Bekanntheitsgrad oder Renommee des jeweiligen Bereitstellers scheitern kann. Die Vertrauenswürdigkeit eines Urhebers in der „echten Welt“ ist ein wichtiger Faktor bei der Entscheidung über Lösungen und Ansätze. Insofern sind Projekte manchmal darauf angewiesen, dass große und bekannte Partner sich in den formierenden Communities engagieren und die Projekte durch Entwicklerressourcen oder monetäre Beiträge fördern. Die Expertinnen und Experten sehen hier die Herausforderung, die Diversität und Pluralität der Community zu erhalten, damit keine Abhängigkeiten von einzelnen Partnern entstehen, die wiederum hemmend wirken können.

Ein weiteres großes Hemmnis sehen die Expertinnen und Experten darin, dass Unternehmen oftmals über wenige Informationen



„**„Public money, public code.“** Es muss gute Gründe geben, nicht Open Source und offene Technologien zu entwickeln – nicht nur bei öffentlichen, sondern auch bei privatwirtschaftlichen Entwicklungen.“

*Rafael Laguna de la Vera, Direktor der Bundesagentur für Sprunginnovationen SPRIND
Bildrechte: SPRIND GmbH*



„ Vor der Entscheidung, mit eigener Software zu einem Open Source-Projekt beizutragen, sollte analysiert werden, inwieweit das Projekt Wissens- und Erfahrungsschatz betrifft, der nicht preisgegeben werden sollte. Dies sollte keine einmalige Entscheidung sein, sondern eine kontinuierliche Abschätzung. Denn was heute unternehmenseigene IP ist, kann morgen schon Commodity sein. Außerdem sollte es einen Prozess im Unternehmen geben, mit dem Beiträge zu Open Source-Projekten geregelt und Mitarbeitern entsprechende Befugnisse erteilt werden.“

„ Nicht alle Unternehmensbereiche eignen sich gleichermaßen für Open Source. Vor allem diejenigen Bereiche, in denen allgemein verfügbare Basistechnologien eingesetzt werden, sind gut geeignet.“

*Dr. Carsten Emde, Geschäftsführer Open Source Automation Development Lab (OSADL) e. G.
Bildrechte: OSADL*

verfügen, von wem die Software bereitgestellt wird. Als Beispiel wird ein KMU mit konservativer Prägung angeführt, das einer ausländischen Software mit unklarem Ursprung nicht vertraut, auch wenn diese aus fachlicher Sicht geeignet scheint.

Neben den fehlenden Vertrauensankern nennen die Expertinnen und Experten als weiteres Hemmnis eine grundsätzliche unternehmerische Risikoaversion. Sie schildern, dass juristische Fachabteilungen oft nicht ausreichend mit der Thematik OSS vertraut sind und deshalb lieber proprietäre Lizenzen empfehlen. Auch sehen die Befragten hemmende Defizite bei der Bewertung von OSS-Kontributionen: Der durch gemeinschaftliche Entwicklungen entstehende immaterielle Unternehmenswert lässt sich in gängigen Controlling-Verfahren oft nicht abbilden mit der Folge, dass OSS ein zu geringer Stellenwert beigemessen wird. Da die Bereitstellung und Nutzung von OSS beziehungsweise die Entwicklung einer unternehmerischen Open Source-Strategie eine Managementfrage ist, spielen nach Ansicht der Expertinnen und Experten Bildungshintergründe auf Führungsebene eine große Rolle – die oftmals nicht im IT-Bereich liegen. Die Befragten beklagen zudem eine unzureichende Vermittlung von Open Source als Konzept in der universitären Lehre. Diese Kritik bezieht sich vor allem auf ingenieurwissenschaftliche Studiengänge, in denen noch vermittelt wird, dass Patente nicht nur ein Indikator für die Innovationskraft von Unternehmen sind, sondern sich auch durch Patentnutzungsgebühren und Prämien für Unternehmen wie Patentinhaber auszahlen. Zu wenig wird nach Ansicht der Expertinnen und Experten in Studium und Ausbildung vermittelt, dass OSS – als Konzept der

Open Innovation – ein Zeichen für die Innovationskraft eines Unternehmens sein kann. In fehlenden Vorkenntnissen und „veralteten“ Innovationsmanagementperspektiven sehen die Expertinnen und Experten damit Hemmnisse für die Verbreitung von OSS.

8.7. Expertenstimmen: Handlungsoptionen zur Stärkung von Open Source Software

Die Expertinnen und Experten wurden im Rahmen der Interviews abschließend gefragt, in welchen Bereichen sie persönlich Handlungsbedarf sehen, um OSS im Kontext von Industrie 4.0 zu stärken. Die Expertinnen und Experten bezogen sich häufig auf Best Practices und nannten zum einen Handlungsfelder, die auf der Unternehmensebene realisiert werden können, zum anderen wurden auch Handlungsempfehlungen für die Politik formuliert.

Die befragten Expertinnen und Experten betonen, dass Open Source eine Managementaufgabe ist und sehen auf Ebene des Führungspersonals mit betriebswirtschaftlichem oder ingenieurwissenschaftlichem Hintergrund Wissensdefizite hinsichtlich des Konzepts und der positiven Effekte von OSS. Aus diesem Grund empfehlen die Expertinnen und Experten, verstärkt Aufklärung und Schulungen in diesem Bereich anzubieten. Die Formate müssten sich dabei speziell an Personen ohne Software- und Programmierkenntnisse richten. Auch sollten Ausbildungsbetriebe und Hochschulen sowie Universitäten das Konzept von OSS verstärkt in ihre Lehrpläne integrieren.



„ Das Potenzial von Open Source zu heben, ist eine **zentrale Herausforderung** im Rahmen europäischer Digitalisierungsbestrebungen.“

*Peter Ganten, CEO Univention GmbH und Vorsitzender des Vorstands der Open Source Business Alliance
– Bundesverband digitale Souveränität e. V.*

Bildrechte: Univention GmbH



„ Für eine **digitale Souveränität Europas** ist Open Source unerlässlich.“

Rafael Laguna de la Vera, Direktor der Bundesagentur für Sprunginnovationen SPRIND

Bildrechte: SPRIND GmbH

Ist das Wissen über die positiven Effekte von OSS in Unternehmen vorhanden und auch auf Seiten des Managements oder der Geschäftsführung für strategisch relevant befunden worden, gilt es innerhalb der Unternehmen, entsprechende Prozesse zu etablieren. Wie der Aufbau eines OSPO gelingen kann und worauf Unternehmen bei der Bereitstellung und Nutzung von OSS achten sollten, könnten zentrale, öffentlich geförderte Anlaufstellen kommunizieren. Die Aufbereitung von Best Practices und der Austausch von Erfahrungswerten durch die Kommunikation von Leuchtturmprojekten kann dabei förderlich sein. Im Zuge der Schaffung eigener Anwendungsfälle müssen Unternehmen – so empfehlen es die Expertinnen und Experten – ein offenes, Open Source-freundliches Mindset etablieren, das Kollaboration und Vernetzung über die proprietäre, interne Entwicklung stellt. Dazu gehören im Rahmen eines Veränderungs-Managements auch Akzeptanz fördernde Maßnahmen für OSS im eigenen Unternehmen. Niederschwellige Weiterbildungsangebote sowie der Besuch entsprechender Lehrveranstaltungen in Hochschulen können hier einen wertvollen Beitrag leisten.

Die Förderung des Aufbaus Open Source-freundlicher, unternehmensinterner Strukturen sehen die befragten Expertinnen und Experten vor allem als staatliche Aufgabe. Insbesondere im Hinblick auf Lizenz- und Compliance-Fragen wünschen sie sich mindestens eine nationale, besser noch europäische Anlaufstelle für Unternehmen. Auch sollte neben den bereits existierenden nationalen und

europäischen Förderangeboten die Förderung von OSS weiter ausgebaut werden. Unternehmen und Forschungseinrichtungen, die ihre Ergebnisse offen zur Verfügung stellen wollen, sollten in Förderentscheidungen mehr als bisher berücksichtigt werden. Als weitere Kriterien für die Zuteilung von Förderung sehen die Befragten die Nachhaltigkeit des Codes und das Interesse der Öffentlichkeit, also die Relevanz. Eine einmalige Veröffentlichung des Quellcodes ohne Community wird dabei nicht als nachhaltig erachtet. Einige Meinungen gehen so weit zu fordern, dass Code, der mit öffentlichen Geldern finanziert wurde, der Öffentlichkeit auch wieder unter einer Open Source-Lizenz zur Verfügung zu stellen ist.

Zusammenfassend lässt sich festhalten, dass die Expertinnen und Experten sich vor allem eine Befähigung der Unternehmen wünschen: Unternehmen müssen sowohl auf rechtlicher Seite in Lizenz-, Patent- und Compliance-Fragen gestärkt werden als auch auf organisatorischer Ebene, bei der Durchführung von Open Source-Projekten. Auch wenn Open Source einen alternativen Ansatz zu proprietären Lizenzen und der Erteilung von Patenten darstellt, müssen Unternehmen hier Sicherheit im Umgang mit urheberrechtlichen Fragestellungen erlangen, um zu prüfen, ob eben jenes durch bereits bestehende Patente und Urheberrechte im Zuge einer Open Source-Veröffentlichung verletzt wird. KMU lassen sich nach Ansicht der Befragten am besten mit niedrighschwelligem Angeboten und leicht zu transferierenden Best Practices unterstützen.

9. SWOT-Analyse von Open Source Software im Kontext von Industrie 4.0

Die vorliegende Expertise bearbeitet die Frage, welche Chancen und Herausforderungen mit OSS im Kontext von Industrie 4.0 verbunden sind. Hierfür wurde nicht nur die einschlägige Fachliteratur intensiv analysiert, sondern auch die Erfahrungen und Einsichten verschiedener Expertinnen und Experten aus Unternehmen, Start-ups und Verbänden gesammelt und strukturiert. Im Folgenden werden im Rahmen einer SWOT-Analyse die Stärken, Schwächen, Chancen und Risiken (Strengths, Weaknesses, Opportunities, Threats) entlang der gängigen PESTL-Dimensionen (Political, Economic, Social, Technological, Legal) zusammengefasst. Diese Art der Betrachtung erlaubt es, Handlungsoptionen abzuleiten, die in Kapitel 10 dargestellt werden.

9.1. Stärken: positive Einflussfaktoren für den Ausbau von Open Source Software im industriellen Umfeld

Aus **politischer** Sicht ist eine wesentliche Stärke für die Verbreitung und den Ausbau von OSS das bereits vorhandene Bewusstsein für die Potenziale von Open Source. Schon heute können Unternehmen, die ihre Ergebnisse im Rahmen eines Förderprojekts quelloffen zur Verfügung stellen, höhere Förderquoten als vergleichbare Antragsteller erzielen.¹⁵⁶ öffentlich geförderte Studien über das wirtschaftliche Potenzial von Open Source für die Industrie¹⁵⁷ sind ebenfalls als Stärke zu bewerten, da sie dieses Bewusstsein weiter schärfen. Open Source steht nicht nur auf der Agenda der Europäischen Kommission, sondern auch im Koalitionsvertrag der neuen Bundesregierung. Dort heißt es zum Beispiel: „*Entwicklungsaufträge werden in der Regel als Open Source beauftragt, die entsprechende Software wird grundsätzlich öffentlich gemacht.*“¹⁵⁸ Der Ausbau einer Open Source-freundlichen Verwaltung und eines Staats, der den freien Zugang zu Quell-Code aktiv fördert, wird angestrebt.

Aus **ökonomischer** Sicht besteht für Deutschland eine entscheidende Stärke in der hohen Kompetenz und dem Know-how in der industriellen Produktion sowie dem Maschinen- und Anlagenbau. Die zunehmenden Software-Applikationen in diesem Bereich stellen eine gute Ausgangsposition für Open Source-Entwicklungen dar und erlauben somit eine Stärkung von Industrie 4.0. Zudem sind die deutschen „Player“ und „Hidden Champions“ am Markt etabliert, kennen bestehende Strukturen und können zum Teil auf jahrzehntelange Expertise zurückgreifen. Dies stellt einen wesentlichen Vorteil gegenüber „digitalen Newcomern“ und „digitalen Größen“ mit weniger Branchenwissen dar – insbesondere im Hinblick auf die Bildung eigener Konsortien und Communities. Entscheidende Stärke

deutscher Maschinen- und Anlagenbauer ist dabei, dass oftmals nicht die Software das marktdifferenzierende geistige Eigentum darstellt, sondern das Know-how der Entwicklung und Konstruktion von Anlagen und Hardware. Die Kombination aus Hardware, Konstruktionsexpertise und zunehmender Digitalisierung der vorhandenen Maschinen und Anlagen als Geschäftsmodell ist als starke Ausgangsposition deutscher Unternehmen zu bewerten.

Hinzu kommt, dass die Nutzung von OSS bereits verbreitet ist – das bedeutet, dass Unternehmen das Konzept von OSS nicht grundsätzlich fremd ist. Der zunehmende Einsatz des OSPO-Konzepts in großen Unternehmen ist dabei eine zusätzliche Stärke: Denn dies zeigt, dass Unternehmen nicht nur OSS nutzen und einsetzen, sondern zunehmend sukzessive daran arbeiten, eigene Entwicklungsergebnisse zur Verfügung zu stellen. Insofern ist Open Source als Teil der Unternehmensstrategie bereits in einigen Unternehmen angekommen. Diese beginnende Professionalisierung mithilfe des OSPO-Konzepts, aber auch die unter dem Dach von Stiftungen bereits vorhandene professionelle Durchführung von Open Source-Projekten bietet großes Potenzial für einen vereinfachten Zugang zu OSS. Die immer stärkere Professionalisierung der Open Source-Ökosysteme trägt zu einer verbesserten Akzeptanz und gesteigerten Implementierung der Lösungen bei. Für Unternehmen bestehen die wirtschaftlichen beziehungsweise ökonomischen Chancen dabei vor allem in der Tatsache, dass OSS als Kollaborationsmodell und Entwicklungsvorgehen Transaktionskosten abbauen kann. Das gilt sowohl für die gemeinschaftliche Entwicklung neuer Software als auch für die Nutzung bestehender Software. Die Verwendung und Weiterentwicklung dessen, was andere zuvor schon erschaffen haben, erlaubt die Einsparung wertvoller Ressourcen und stärkt Synergieeffekte.

Aus **sozialer** Sicht liegt eine Stärke in der positiven Wahrnehmung von Open Source. Zwar wurden primär Expertinnen und Experten interviewt, die OSS selbst verwenden oder bereitstellen – doch bestätigen die Befragten, dass insbesondere Fachkräfte aus dem Informatikbereich Arbeitgeber bevorzugen, die ihnen die Mitarbeit in Open Source-Projekten ermöglichen. Insofern ist eine unternehmerische Ausrichtung auf Open Source für IT-Fachangestellte ein wichtiges Kriterium in der Wahl des Arbeitgebers. Eine weitere Stärke im sozialen Bereich ist die zunehmende Internationalisierung der Belegschaft von Unternehmen. Open Source-Projekte kennen keine Ländergrenzen und Nationalstaaten. Folglich ist die gesellschaftliche Öffnung und Internationalisierung eine gute Grundvoraussetzung für die Schaffung diverser, internationaler Entwicklungs-Communities. Eine weitere Stärke aus sozialer Sicht ist, dass es bereits viele Industrie 4.0-Initiativen, unternehmerische Verbände und Organisationen

156 Vgl. Europäische Kommission 2020a.

157 Vgl. Blind et al. 2021.

158 Vgl. Koalitionsvertrag 2021.

gibt, wie die Plattform Industrie 4.0, Bitkom oder der Verein Deutscher Ingenieure (VDI), die das Thema OSS vorantreiben.

Aus **technologischer** Sicht liegen viele der Stärken in den guten ökonomischen Grundvoraussetzungen begründet, die dazu geführt haben, dass neue Industrie 4.0-Projekte auf eine Vielzahl an guten Vorarbeiten zurückgreifen und auf den Ergebnissen vorangegangener Projekte aufsetzen können. Projekte wie open62541 (siehe Kapitel 6.1.9) zeigen, dass eine Zusammenarbeit für eine gemeinschaftliche Stärkung von Industrie 4.0 gelingen kann. Ebenso kann als Stärke aus technologischer Sicht eine weitere Erkenntnis der vorliegenden Expertise angeführt werden: OSS ist in erster Linie ein Kooperations- beziehungsweise Kollaborationsmodell und erst in zweiter Linie ein Lizenzmodell. Aus diesem Grund ist OSS „technologieneutral“. Das bedeutet, dass es theoretisch keine Bereiche gibt, die sich aus technologischer Sicht nicht für eine gemeinsame Entwicklung eignen würden.

Aus **rechtlicher** Sicht ist das Aufkommen erster Anlaufstellen, zum Beispiel die eingetragene Genossenschaft OSADL, die das Ziel hat OSS-Projekte zu fördern, zu koordinieren und dabei auch rechtlicher Ansprechpartner zu sein, als Stärke zu bewerten. Auch Initiativen wie der OpenChain Standard ISO/IEC 5230 stärken die standardisierte Vorgehensweise in der OSS-Entwicklung und die Rechtssicherheit. Eine weitere Stärke aus rechtlicher Sicht besteht zudem in der zunehmenden Konvergenz im Lizenzbereich. Es kristallisiert sich zunehmend eine Verwendung bestimmter permissiver Lizenzen beziehungsweise von „Lizenz-Standards“ heraus. Unternehmen verwenden einheitliche Lizenzen oder bevorzugt permissive Lizenzen, um ihre OSS bereitzustellen und somit eine hohe Lizenzkompatibilität zu gewährleisten. Dies führt zu einer erhöhten Akzeptanz von OSS und stärkt die gemeinsame Weiterentwicklung von Software unter einheitlichen Lizenzen.

9.2. Schwächen: Ungenutzte Potenziale von Open Source Software für Industrie 4.0

Neben begünstigenden Faktoren stehen aber auch eine Vielzahl an Schwächen einer größeren Verbreitung des Open Source-Ansatzes in Industrie 4.0 entgegen. Als **politische** Schwäche ist derzeit zu beobachten, dass zwar OSS-Kontributionen explizit in Ausschreibungen von Fördermittelgebern verlangt, aber Verstärkungskonzepte und der Aufbau von Ökosystemen bisweilen nicht oder nur selten berücksichtigt werden. So können Fehlanreize entstehen, der Community-Ausbau wird vernachlässigt und eine nachhaltige Entwicklung nicht realisiert.

Eine weitere Schwäche auf **ökonomischer** Ebene bzw. auf Unternehmensebene ist, dass nach Ansicht der Expertinnen und Experten die strategische Bedeutung von OSS für den Unternehmenserfolg auf der Managementebene nicht genug wahrgenommen wird. Dies ist ein Grund dafür, dass OSS sich nicht vollends im Kontext von Industrie 4.0 etablieren kann. Zwar sind Entwicklerinnen und Entwickler die Konzepte und Vorteile von OSS zum Teil bekannt, doch verfügen diese zum oftmals über zu wenig Wissen beziehungsweise

Möglichkeiten sich aktiv in die Gestaltung von Geschäftsmodellen oder Unternehmensstrategien einzubringen. Der fehlende Austausch und das fehlende, bereichsübergreifende, interdisziplinäre Verständnis der vielseitigen Aspekte von Open Source Software stellen große Schwächen innerhalb von Unternehmen dar.

Aus **sozialer** beziehungsweise gesellschaftlicher Sicht besteht eine Schwäche darin, dass sich von den vielen Open Source-Projekten nur ein Bruchteil tatsächlich behaupten und etablieren kann. Dies ist oft auf eine fehlende Community, mangelnde Ressourcen oder mangelndes Engagement der Partner zurückzuführen. Ebenso bewerten die Erstellerinnen und Ersteller der vorliegenden Expertise deutsche Open Source-Projekte im internationalen Vergleich als relativ unbedeutend. Insofern fehlen hier gegebenenfalls Leuchtturmprojekte und Positivbeispiele, die auch international Aufmerksamkeit erregen können.

Eine weitere Schwäche, die OSS im Kontext von Industrie 4.0 hemmt, ist die Art und Weise, wie tradierte Industrieunternehmen Projekte entwickeln. Dies kann für die **technologische** Entwicklung herausfordernd sein: Auf der einen Seite wurden das Know-how, Branchenkenntnisse und Fachkompetenz als große Stärke und gute Ausgangsposition bewertet – auf der anderen Seite stehen traditionelle Produktentwicklungszyklen und Vorgehensweisen im Gegensatz zur agilen Software-Entwicklung, einer kontinuierlichen Integration (Continuous Integration) und einer ständigen Aktualisierung von Software (Continuous Deployment) im Open Source-Bereich. Die Software-Entwicklung zeichnet sich heute durch Schnelllebigkeit aus und braucht mitunter informelle Entscheidungswege. Hierarchische Strukturen, ein konservatives Projektmanagement und starre Projektplanung auf einen langfristigen Zeitraum können eine agile, ergebnisoffene Software-Entwicklung hemmen.

Die Unsicherheit in Bezug auf Lizenzfragen stellt bei eigenen Software-Kontributionen ein großes Hemmnis dar. Diese **rechtlichen** Hürden sind für Unternehmen oftmals eine Herausforderung – nicht nur im nationalen, sondern zunehmend auch im internationalen Zusammenspiel mit Partnern. Die Unsicherheiten beziehen sich vor allem auf das Patentrecht: Unternehmen zögern, OSS bereitzustellen aus Sorge, bestehende Patente (unwissentlich) zu verletzen. Eine weitere Schwäche aus rechtlicher Sicht im weiteren Sinne besteht darin, dass es zwar Synergie-Effekte zwischen Standardisierungsorganisationen und Open Source Communities gibt, diese aber nicht ausreichend dahingehend genutzt werden, aus der Erfahrung von Open Source Communities heraus De-jure-Standards zu entwickeln. Hier kann die wertvolle Rolle von OSS als komplementierender Referenzimplementierungsgeber oftmals nicht vollends erschlossen beziehungsweise genutzt werden.

9.3. Chancen: Förderliche Rahmenbedingungen zur Verbreitung von Open Source Software für Industrie 4.0

Die vorliegende Expertise zeigt, dass eine große Chance für die Stärkung von Industrie 4.0 darin besteht, dass Unternehmen Open

Source gegenüber grundsätzlich aufgeschlossen sind. Wie Studien belegen (siehe Abschnitt 4.2), kann Open Source einen erheblichen Beitrag zu einem positiven Wirtschaftswachstum leisten. Des Weiteren verbinden die im Rahmen dieser Expertise befragten Expertinnen und Experten mit Open Source die Hoffnung auf digitale Souveränität Europas. Das Bewusstsein für makroökonomische Zusammenhänge ist vorhanden: Durch OSS können die Anbietervielfalt und eine freie Auswahl an Lösungen gesichert werden. Folglich kann der Marktmacht monopolistischer Software-Anbieter mithilfe von Open Source ein breites Produktangebot entgegengesetzt werden. In dieser Erkenntnis liegt die große **politische** Chance Deutschlands und der ganzen Europäischen Union: Diese Zusammenhänge müssen transparent dargestellt und politische Open Source-Strategien konsequent verwirklicht werden.

Eine **ökonomische** Chance für OSS könnte in der wirtschaftlichen beziehungsweise buchhalterischen Bewertung von Unternehmen liegen. Weniger Patente und Schutzrechte als vielmehr die Qualität von Quellcode und die Intensität der Community-Beteiligungen könnten zukünftig für die Bewertung der Innovationskraft eines Unternehmens entscheidend sein. Für Unternehmen ergeben sich über die Realisierung erfolgreicher Geschäftsmodelle hinaus mit OSS weitere ökonomische Chancen, die zur Stärkung der unternehmerischen Position beitragen können.

Aus gesellschaftlicher beziehungsweise **sozialer** Sicht kann eine weitere Chance für OSS darin bestehen, dass bereits eine „Sharing-Kultur“ etabliert ist. Die Sharing-Ökonomie stärkt das Bewusstsein für Kollaborationen und geteilte Ressourcen – zwar primär im Privatsektor, aber auch zunehmend im industriellen Umfeld. Die neue Generation von Entwicklerinnen und Entwicklern ist durch diese Kultur des Teilens von Wissen und Ressourcen geprägt. Auch OSS kann als Teil einer reziproken unternehmerischen Ökonomie des Teilens verstanden werden. Eine weitere Chance zur Stärkung von OSS besteht in der aufkommenden sogenannten Purpose Economy. Immer mehr Unternehmen hinterfragen kurzfristige Profitmaximierung und zielen auf Nachhaltigkeit und Sinnstiftung ab. Es geht darum, ein wertorientiertes und motivierendes Umfeld für Mitarbeitende zu schaffen, aber auch den sinnhaften gesellschaftlichen Mehrwert der eigenen Produkte und Dienstleistungen für den Kunden in den Fokus zu rücken. Neue Unternehmensformen wie Gesellschaften in Verantwortungseigentum – also Unternehmen, die den Mitarbeitenden „gehören“ – sind Ausdruck dieser Bewegung. Auch OSS kann für die an der Entwicklung beteiligten Personen als sinnstiftend betrachtet werden, denn man gibt einer Community oder Gemeinschaft ein Produkt zurück, das dann allen gehört. Dieses neue Unternehmertum mit Ausrichtung auf Sinnstiftung und Nachhaltigkeit kann OSS eine neue Kultur der Offenheit und des Teilens stärken. Letztendlich kann Open Source auch als barriere- und diskriminierungsfreier Zugang zu Software für alle Personen und Gesellschaftsschichten verstanden werden.

Eine solche Kultur des Teilens, Partizipierens und der Offenheit führt letztendlich zur Erschließung eines großen **technologischen** Potenzials. Die Öffnung von Quellcode und die kollaborative Entwicklung bergen die Chance, neue Technologiefelder gezielter

voranzutreiben. Da in vielen Industrie 4.0-Bereichen Potenzial für Open Source-Projekte vorhanden ist, besteht hier vor allem die Chance, relevante Themenfelder einfach und zügig weiterzuentwickeln, um Industrie 4.0 „Made in Germany“ oder „Made in Europe“ zu stärken. Die vorliegende Expertise hat gezeigt, dass spezifische Industrie 4.0-Fachanwendungen, beispielsweise aus den Bereichen Produktionsplanung oder Bestandsmanagement, bisher selten als OSS bereitgestellt werden. Hier liegt eine Chance im Ausbau von (nicht wettbewerbsdifferenzierenden) Fachanwendungen im Bereich von Industrie 4.0. Eine weitere Chance aus technologischer Sicht ist die zunehmende Verbreitung von OSS über Hardware. Unternehmen, die Anlagen und Hardware fertigen, stellen zunehmend offene Schnittstellen bereit beziehungsweise öffnen ihren Quellcode sogar darüber hinaus. Dies stärkt die Akzeptanz und nachhaltige Nutzung der Anlagen und kann gleichzeitig zu einer Verbesserung der Software-Entwicklung führen. Gleichzeitig können hier weiter De-facto-Standards etabliert und gestärkt werden.

Der bereits erwähnte **rechtliche** Aspekt der zunehmenden Lizenzkonvergenz und der damit verbundene Trend zu permissiven Lizenzen bedeutet, dass Unternehmen im Bereich Industrie 4.0 vereinfacht proprietäre und kommerzielle Geschäftsmodelle auf Basis von OSS realisieren können. Anders als bei Copyleft-Lizenzen erlauben die meisten permissiven Lizenzen eine proprietäre Nutzung und stellen somit eine einfache Möglichkeit für Unternehmen dar, die entsprechenden Softwarekomponenten zu verwenden und relativ rechtssicher zu kommerzialisieren.

9.4. Risiken: Negative Einflussfaktoren für die Verbreitung von Open Source Software im Bereich Industrie 4.0

Neben den bereits genannten Schwächen gibt es auch eine Reihe von Faktoren, die nicht nur Schwächen, sondern tatsächliche Risiken für eine erfolgreiche Unterstützung von Industrie 4.0 durch OSS darstellen können. Auf **politischer** Seite gilt es vor allem, wie bereits dargestellt, Fehlanreize zu vermeiden und Förderrichtlinien in Bezug auf OSS nachhaltig zu gestalten. Bei der Ausgestaltung und Bewertung von Projekten ist dabei zu berücksichtigen, dass Open Source nur ein Kriterium und nicht alleiniges Merkmal sein kann – die Relevanz des Vorhabens sollte dabei ebenso entscheidend sein. Zudem muss sichergestellt werden, dass ein komplementierendes Ökosystem erschaffen wird. Andernfalls hat OSS keinen nennbaren positiven, skalierenden Effekt. Werden diese Aspekte der inhaltlichen Zielsetzung und nachhaltigen Verstetigungsinfrastrukturen nicht entsprechend adressiert und berücksichtigt, sehen sich nationale und europäische Förderinitiativen mit dem Risiko des Scheiterns konfrontiert. Zudem gilt es Marktverzerrungen zu vermeiden, das heißt es sollten primär Software beziehungsweise Entwicklungsergebnisse gefördert werden, die einen wohlfahrtssteigernden Ansatz haben und bestehende Geschäftsmodelle nicht schädigen.

Aus **ökonomischer** Perspektive besteht ein großes Defizit beziehungsweise Risiko in der grundlegenden Struktur von OSS: Open Source-Kontributionen und -Projekte können aus

wirtschaftswissenschaftlicher Perspektive als private Bereitstellung öffentlicher Güter betrachtet werden. Öffentliche Güter sind Güter, deren Verfügbarkeit durch den Konsum beziehungsweise die Nutzung nicht abnimmt und von deren Nutzung niemand ausgeschlossen werden kann. Oftmals stellt der Staat öffentliche Güter wie Infrastruktur (Beleuchtung, Straßen etc.) bereit. Im Falle von Open Source handelt es sich jedoch um eine Bereitstellung durch Privatpersonen und Firmen. Folglich kann diese Bereitstellung im weiteren Sinne auch als „Spende“ an die Gemeinschaft betrachtet werden. Insofern kann grundsätzlich davon ausgegangen werden, dass die Bereitstellung von Software zur freien Nutzung für Unternehmen ökonomisch keinen Sinn macht, solange nicht von allen Code für ein „volkswirtschaftliches“ Optimum bereitgestellt wird.¹⁵⁹ Derzeit wird OSS wirtschaftspolitisch nicht als Spende betrachtet, sodass hier gegebenenfalls nicht genug Anreize geschaffen werden. Die Free-Rider-Problematik der Nutzung ohne Gegenwert kann eine Unterversorgung mit OSS verstärken.

Ein weiteres ökonomisches Risiko stellen monopolistische Plattformanbieter wie Amazon und Google für die bestehende Anbietervielfalt und deren Kerngeschäfte dar: Die großen Software- und Technologieanbieter dringen zunehmend in den B2B-Bereich vor und etablieren eigene Standards. Amazon ist bereits mit Amazon Web Services (AWS) ein marktbestimmender Anbieter von Cloud-Diensten und auch Google, beziehungsweise der Mutterkonzern Alphabet, bringt im Bereich der KI-Entwicklung, zum Beispiel mit TensorFlow, einem Framework für datenstromorientierte Programmierung, eigene Lösungen voran. Ein weiteres Beispiel ist der Konzern Meta (ehemals Facebook), der mit PyTorch neue Maßstäbe im Bereich des maschinellen Lernens setzt. Diese Plattformunternehmen bergen zudem durch ihre Netzwerkeffekte und das breite Angebot an oftmals kostenfrei bereitgestellten komplementierenden Diensten und Services die Gefahr von Vendor Lock-ins, das heißt die Abhängigkeit von ausschließlich einem Anbieter. Aus Sicht der Industrie schwächen die Tech-Giganten mit ihren Unternehmenspraktiken die Produktvielfalt und eine wettbewerbsgetriebene Innovation. OSS-Projekte können dabei helfen, Alternativen zu den bestehenden Angeboten monopolistischer Konzerne anzubieten und eigene, unabhängige Entwicklungen voranzutreiben. Durch die Anbietervielfalt und den freien Wettbewerb können wesentlich höhere Wohlfahrtsgewinne realisiert werden als in monopolistischen Märkten. Eine Unterrepräsentation eigener europäischer (Open Source-)Lösungen stellt folglich ein Risiko dar. Hier gilt es zu berücksichtigen, dass sowohl die Sorge vor der Vereinnahmung von Lösungen durch monopolistische Anbieter wie auch fehlende Ökosysteme und Communities die Bereitstellung europäischer Open Source-Lösungen hemmen können. Aber auch fehlende Anreize aufgrund der bereits etablierten Lösungen von Alphabet und gleichwertigen Technologiekonzernen hemmen die Bereitstellung solcher Lösungen.

Die Sorge, in diesem Spannungsfeld zwischen Wettbewerb und Kollaboration – auch mit großen Technologie-Anbietern – eigene

marktdifferenzierende Ergebnisse zu teilen, führt aus **sozialer** Sicht zu einem hohen Risiko für eine starke Industrie 4.0-Community. Unsicherheiten bezüglich des eigenen Geschäftsmodells, wie auch eine hohe Risikoaversion und entsprechende Sorgen etwas Wettbewerbsdifferenzierendes zu teilen, das von großen monopolistischen Plattformen vereinnahmt werden könnte, wirken sich negativ auf eine pluralistische, community-getriebene Entwicklung von Industrie 4.0-Lösungen aus. Dies stellt insofern ein Risiko dar, als dass die Bereitschaft etwas zu teilen unmittelbar von der Bereitschaft anderer, etwas zu teilen abhängt. Wird keine kritische Masse erreicht, sind entsprechende Communities und Weiterentwicklungen gehemmt. Ebenso ist anzumerken, dass Open Source als Konzept grundsätzlich gesamtgesellschaftlich wenig bekannt ist. In anderen Berufsgruppen als der Informatik gibt es große Wissenslücken bis hin zur gänzlichen Unbekanntheit – auch dies steigert die Risikoaversion seitens des Managements, das sich im Zweifelsfall eher gegen eine Open Source-Stellung entscheidet.

Neben diesen Risiken, birgt aber auch schlichtweg schlechte, fehlerhafte oder schädliche OSS aus **technologischer** Sicht das Risiko, OSS insgesamt in ein schlechtes Licht zu stellen. Trotz der positiven Effekte von OSS ist es unbestritten, dass auch viele qualitativ minderwertige Projekte existieren und dass Sicherheitslücken in weit verbreiteten Komponenten, wie in jüngerer Vergangenheit beispielsweise bei Apache Struts (CVE.2017-5638) oder Apache Log4j (CVE-2001-44228) geschehen, weitreichende Auswirkungen auf die Sicherheit der diese Komponenten verwendenden Systeme haben können. Insofern kann das Bekanntwerden solcher Schwächen zum einen das Image von OSS insgesamt schädigen, aber eben auch das Heben der Potenziale verhindern. Fehler führen dazu, dass OSS weniger akzeptiert ist, genutzt wird und auch weniger Fachkräfte bereit sind, zu Projekten beizutragen.

Aus **rechtlicher** Sicht stellen vor allem Patentverletzungen oder Missachtungen entsprechender Urheberrechte wie auch die Verwendung nicht kompatibler Lizenzen ein hohes Risiko dar. Insbesondere die Missachtung lizenzrechtlicher Fragestellungen sowie die Weiterverbreitung von Ergebnissen, die Lizenzrecht missachten, kann ein erhebliches Risiko bedeuten – nicht nur für das Unternehmen, sondern für alle Nutzer der bereitgestellten OSS-Lösung. Zudem ist die mangelnde Harmonisierung unterschiedlicher Rechtsrahmen (zum Beispiel amerikanisches Recht und europäisches Recht in Bezug auf Haftung) ein rechtliches Risiko für die Verbreitung von OSS.

9.5. Zusammenfassung Stärken, Schwächen, Chancen und Risiken von Open Source Software für Industrie 4.0

In der folgenden Übersicht (siehe Tabelle 18) werden die wesentlichen Kernerkenntnisse der Bedeutung und Rolle von OSS für Industrie 4.0 zusammengefasst.

¹⁵⁹ Vgl. Blind et al. 2021.

Tabelle 18: SWOT-Analyse Open Source Software im Kontext von Industrie 4.0

	Stärken	Schwächen
Politisch	<ul style="list-style-type: none"> • Politisches Bewusstsein für Open Source Software ist vorhanden, sowohl auf EU-Ebene als auch auf nationaler Ebene • Berücksichtigung in Förderbekanntmachungen 	<ul style="list-style-type: none"> • Förderbekanntmachungen und Ausschreibungen adressieren den Aufbau eines Open Source-Verstetigungsökosystems nicht ausreichend
Wirtschaftlich	<ul style="list-style-type: none"> • Hohe Kompetenz und Know-how in der industriellen Produktion sowie dem Maschinen- und Anlagenbau als Ausgangsbasis für Industrie 4.0 • Software oft nicht das marktdifferenzierende geistige Kerneigentum, daher Kombination aus Soft- und Hardware als gute Ausgangsbedingung • Sukzessiver Ausbau und Professionalisierung von Open Source durch OSPOs in großen Unternehmen steigert Akzeptanz • Nutzung von Open Source Software minimiert Transaktionskosten, stärkt Synergieeffekte und schont Ressourcen 	<ul style="list-style-type: none"> • Open Source Software als Managementthema noch nicht ausgereift beziehungsweise noch nicht „angekommen“ • Fehlendes, bereichsübergreifendes Verständnis für OSS
Sozial	<ul style="list-style-type: none"> • Positive Wahrnehmung von Open Source Software • Zunehmende Internationalisierung des Arbeitsumfelds fördert Offenheit für internationale Entwickler-Communities • Viele vorhandene Industrie 4.0-Initiativen 	<ul style="list-style-type: none"> • Vielzahl an Projekten, aber ohne Communities, dadurch geringe Ausschöpfung von Potenzialen • Deutsche Projekte im internationalen Kontext relativ unbedeutend, dadurch fehlende Bekanntheit und Aufmerksamkeit
Technologisch	<ul style="list-style-type: none"> • Viele vorhandene Projekte bieten die Chance für erfolgreiche Industrie 4.0-Implementierungen • Open Source Software per se technologieoffen und neutral 	<ul style="list-style-type: none"> • Traditionelle Produktentwicklungszyklen und Vorgehensweisen in Unternehmen stellen einen Gegensatz zur agilen und schnelllebigen Softwareentwicklung im Open Source-Bereich dar
Rechtlich	<ul style="list-style-type: none"> • Erste Bemühungen und Initiativen, Rechtssicherheit zu stärken: OSADL, OpenChain • Erster Standard (ISO/IEC 5230) trägt zu Rechtssicherheit beziehungsweise Professionalisierung bei • Konvergenz im Lizenzbereich 	<ul style="list-style-type: none"> • Unsicherheiten in Bezug auf Patent- und Lizenzrecht stellen Unternehmen insbesondere bei eigenen Kontributionen vor Herausforderungen • Fehlende Synergieeffekte zwischen Standardisierungsorganisationen und Open Source-Communities
	Chancen	Risiken
Politisch	<ul style="list-style-type: none"> • Vorhandene Ansätze und Verständnis für OSS stärker in die politische Agenda integrieren: OSS als Baustein technologischer Souveränität 	<ul style="list-style-type: none"> • Es gilt, Marktverzerrungen zu vermeiden, d. h. es sollte primär nicht wettbewerbsdifferenzierende Software gefördert werden • Fehlende Gestaltung von Förderrichtlinien im Hinblick auf ein Verstetigungsökosystem und inhaltliche Sinnhaftigkeit können eine nachhaltige Förderung verhindern
Wirtschaftlich	<ul style="list-style-type: none"> • Open Source Software-Kontributionen als zukünftige Bewertungsmaßstäbe für Innovationskraft (als alternativer Indikator bei der Unternehmensbewertung zu Patenten) 	<ul style="list-style-type: none"> • OSS als öffentliches Gut, das privat bereitgestellt wird. Dadurch keine Anreize für Unternehmen Code bereitzustellen, wenn nicht ausreichend andere Unternehmen auch Code „spenden“ um ein volkswirtschaftliches Optimum zu erreichen • Aufkommende Free-Rider-Problematiken hemmen die Verstetigung von OSS • Vendor Lock-ins bestehender monopolistischer Konzerne als Risiko für neue Open Source-Kontributionen und -Projekte
Sozial	<ul style="list-style-type: none"> • Zunehmende Sharing-Kultur und Purpose Economy stellen eine Chance für den Open Source-Gedanken dar und können somit Industrie 4.0-Entwicklungen stärken • OSS als barrierefreier Zugang zu Software für alle Gesellschaftsschichten 	<ul style="list-style-type: none"> • Hohe Risikoaversion in Bezug auf Offenheit und Open Source Software hemmt die Verbreitung und Akzeptanz
Technologisch	<ul style="list-style-type: none"> • Großes Entwicklungspotenzial für Fachanwendungen in der Industrie 4.0 	<ul style="list-style-type: none"> • Fehlerhafte OSS beziehungsweise qualitativ minderwertige Software kann zu schlechten oder zum Teil sicherheitsbedenklichen Entwicklungen führen
Rechtlich	<ul style="list-style-type: none"> • Trend hin zu permissiven Lizenzen bedeutet eine vereinfachte Verwendung und ggfs. Kommerzialisierung von OSS für Unternehmen durch erhöhte Rechtssicherheit 	<ul style="list-style-type: none"> • Patentverletzungen und Verletzungen des Urheberrechts als Risiko für eigene Open Source-Kontributionen • Mangelnde Harmonisierung internationaler Rechtsrahmen erhöht Rechtsunsicherheiten z. B. in Bezug auf Haftungsfragen

10. Handlungsoptionen zur Stärkung von Open Source Software im Industrie 4.0-Kontext

Die vorliegenden Ergebnisse dieser Expertise erlauben die Herleitung verschiedener Handlungsoptionen. Insbesondere die Schwächen und Risiken zeigen auf, in welchen Bereichen politische, forschungsgetriebene, aber auch privatwirtschaftliche Maßnahmen zur Stärkung von OSS zur Umsetzung von Industrie 4.0 beitragen können. Die Handlungsoptionen sind nicht nur Ergebnisse der vorliegenden Analysen und Experteninterviews, sondern auch durch den iterativen Diskurs der Erstellenden dieser Expertise entstanden. Im Wesentlichen lassen sich die Handlungsoptionen in vier übergeordnete Bereiche gliedern, die im Folgenden vorgestellt werden:

1. Aufklärungskampagnen und Wissenstransfer
2. Forschungsbedarfe und fehlende Bildungsangebote
3. Nachhaltige Ausgestaltung von Fördermaßnahmen und Leuchtturmprojekten
4. Stärkung von Organisationsstrukturen und Koordination sowie inhaltliche Ausrichtung von Open Source-Initiativen im Industrie 4.0-Kontext

10.1. Aufklärungskampagnen und Wissenstransfer

Für das Bereitstellen benötigter Informationen lassen sich Handlungsoptionen wie folgt zusammenfassen:

1. Eine wesentliche Erkenntnis der vorliegenden Expertise ist, dass vor allem im Bereich der indirekten Vorteile und Mehrwerte Wissenslücken bestehen, die eine **grundlegende Aufklärungsarbeit und Kampagnen** über die positiven Effekte von OSS erforderlich machen. Hier gilt es zu betonen, dass OSS viel mehr ist als lediglich „kostenfreie Software“ – nämlich ein Kooperations- und Entwicklungsmodell, das Unternehmen hilft, neue Geschäftsmodelle und Innovationen zu realisieren. Die Aufklärungsarbeit zielt darauf ab, gängige Vorurteile und Ängste abzubauen und darzustellen, dass geistiges Kerneigentum nicht bereitgestellt werden muss.
2. Neben der niederschweligen Aufklärung und Heranführung von Unternehmen an das Thema OSS bedarf es auch einer konkreten, auskunftsfähigen **Anlaufstelle im Bereich der Lizenz- und Patentfragen auf europäischer Ebene**. Unternehmen, insbesondere Start-ups und KMU, müssen einen einfachen (möglichst kostenfreien) Zugang zu Informationen in Bezug auf Rechts- und Patentfragen erhalten. So können Hemmnisse und Risiken minimiert werden.

3. Maßgeschneiderter **Wissenstransfer für KMU**, aber auch für Start-ups, können diese Zielgruppen besser erreichen. Neben den Aufklärungskampagnen sollten besondere Angebote und Transferformate speziell für diese Zielgruppen, zum Beispiel Workshops und Sprechstunden, sowie zusätzliche Experimentierräume geschaffen werden. Die Vermittlung von Open Source-Strategien und Wissen über den Aufbau eines OSPO sollte im Vordergrund stehen. Die Schaffung gemeinsamer Konsortien oder die Einführung in Entwicklungsumgebungen kann für KMU eine große Unterstützung bieten. Eine Zusammenarbeit mit bestehenden Initiativen wie den Zentren im Netzwerk Mittelstand-Digital oder Multiplikatoren kann zielführend sein. Das Aufzeigen von Best Practices und die Darstellung bereits vorhandener Lösungen können zudem sensibilisieren und neue Anreize für Unternehmen schaffen.

10.2. Forschungsbedarfe und fehlende Bildungsangebote

Die vorliegende Expertise hat einige spannende Ergebnisse im Kontext OSS und Industrie 4.0 erbracht. Diese gilt es, im Rahmen von mittelfristigen Forschungsschwerpunkten innerhalb der angewandten, aber auch der universitären und außeruniversitären Grundlagenforschung zu vertiefen. Konkrete Bedarfe sehen die Erstellerinnen und Ersteller der Expertise in den folgenden Bereichen:

1. Die Entwicklung neuer Konzepte und Vorgehensmodelle zur Stärkung von **Open Source im Management** könnte eine Aufgabe für betriebswirtschaftlich geprägte Forschungsrichtungen sein. Hier steht vor allem die Schaffung von Synergien und die effiziente Nutzung von Ressourcen bei der gemeinsamen Innovations- und Software-Entwicklung im Vordergrund. Ebenso sollten Open Source-Geschäftsmodelle für „traditionelle“ Industrieunternehmen in den Fokus gerückt werden und es sollte untersucht werden, wie sich aus der Kombination von Hardware, Open Source und Industrie 4.0 neue innovative Erlös- und Ertragsmodelle realisieren lassen.
2. Neben der Weiterentwicklung von Management-Ansätzen für eine auf Open Source basierende Industrie 4.0 gibt es auch konkreten Forschungsbedarf im Bereich der **Innovationskennzahlen für die Unternehmensbewertung und des Controllings**. Hier stellt sich insbesondere die Frage, wie der Erfolg von Open Source-Kontributionen und der Mehrwert von Open Source Communities in ein modernes Controlling Einzug halten kann.
3. Damit verbunden sind auch Fragen der **steuerrechtlichen Anreize**: Motivation für OSS-Kontributionen könnte geschaffen werden, wenn diese steuerrechtlich begünstigt wären. Wie eine

gerechte und faire Deklaration, beispielsweise als Spende, aussehen könnte – nicht nur der Umfang des Codes, sondern auch die Qualität ist entscheidend – stellt aus Sicht der Autorinnen und Autoren eine spannende Forschungsfrage dar.

4. Ingenieurwissenschaftliche und betriebswirtschaftliche Studiengänge sollten neben der Forschung auch die **Lehre zu Open Source-Themen vertiefen**. Vermittelt werden muss der Open Source-Ansatz im Rahmen der Innovationstheorie als Kollaborations- und Entwicklungsmodell. Zukünftige Führungskräfte sollten lernen, dass nicht nur Patente ein Indikator für Innovationsfähigkeit sind, sondern auch die gemeinschaftliche Entwicklung nach dem Open Source-Ansatz und die Schaffung dringend benötigter OSS in vitalen Communities. Hier gilt es, ein neues Mindset zu etablieren. Dies kommt einer stärkeren Vernetzung von Industrie 4.0-Unternehmen zugute, denn zukünftig werden Unternehmen nicht mehr oder kaum noch in der Lage sein, Entwicklungen eigenständig und schnell genug voranzutreiben. Außer in den einschlägigen Studiengängen sollte dieses innovationstheoretische Wissen und die Vermittlung eines neuen Mindsets aber auch in **Weiterbildungsangebote für Fach- und Nachwuchskräfte** integriert werden. Hier stehen Industrieverbände und -kammern in der Pflicht, entsprechende Angebote einzurichten.

10.3. Nachhaltige Ausgestaltung von Fördermaßnahmen und Leuchtturmprojekten

Damit zukünftige Fördermaßnahmen nachhaltig sind, gilt es, bestimmte in dieser Expertise identifizierte Fehler zu vermeiden und die richtigen Anreize zu schaffen. Dies gilt auch für geförderte Leuchtturmprojekte und deren Verstetigung. Daher formulieren die Erstellerinnen und Ersteller folgende Handlungsoptionen:

1. Der Aufbau von nachhaltigen Communities und die Schaffung eines Transferökosystems (zum Beispiel Technologiepartner, die für KMU Software implementieren können) muss fester Bestandteil von Förderbekanntmachungen sein. Eine geeignete Maßnahme scheint hier das **Anpassen der Bewertungsmetriken** für die Auswahl von Förderanträgen: Antragssteller, die entsprechende Verstetigungskonzepte mit relevanten Partnern vorlegen, könnten bevorzugt berücksichtigt werden. Antragstellende Unternehmen und Forschungseinrichtungen sollten darstellen, wie sie eine Open Source Community aufbauen und pflegen möchten und welche Relevanz die entwickelte OSS für Industrie 4.0 darstellt. Dabei ist zu prüfen und vom Antragssteller darzulegen, inwieweit das Vorhaben anderen Marktteilnehmern schaden kann.
2. Neben Verstetigungskonzepten sollten **interdisziplinäre Teams** gefördert werden. Wir brauchen interdisziplinäre Expertise aus den Bereichen Informatik, Wirtschaftswissenschaften und Ingenieurwissenschaften, um Open Source aus der „Nische“ der Informatik zu holen und für den Kontext Industrie 4.0 fruchtbar zu machen. Insbesondere die Entwicklung von Geschäftsmodellen

und neuen Managementansätzen beziehungsweise deren Erprobung sollte im Vordergrund stehen.

3. Für öffentlich geförderte Forschungsprojekte kann die Überlegung angestellt werden, ob generell der entwickelte Quellcode der Öffentlichkeit als Rückgabe für die Finanzierungsleistung unter einer Open Source-Lizenz zur Verfügung zu stellen ist. Hier würde die Devise **„Public Money, Public Code!“** greifen.
4. Bei der Beschaffung von Software durch die öffentliche Hand sind **lokale und kleinere Anbieter, die auf Open Source setzen, zu bevorzugen**. Das fördert die Diversität und Pluralität von Software und stärkt die Souveränität im Sinne der Anbietervielfalt.

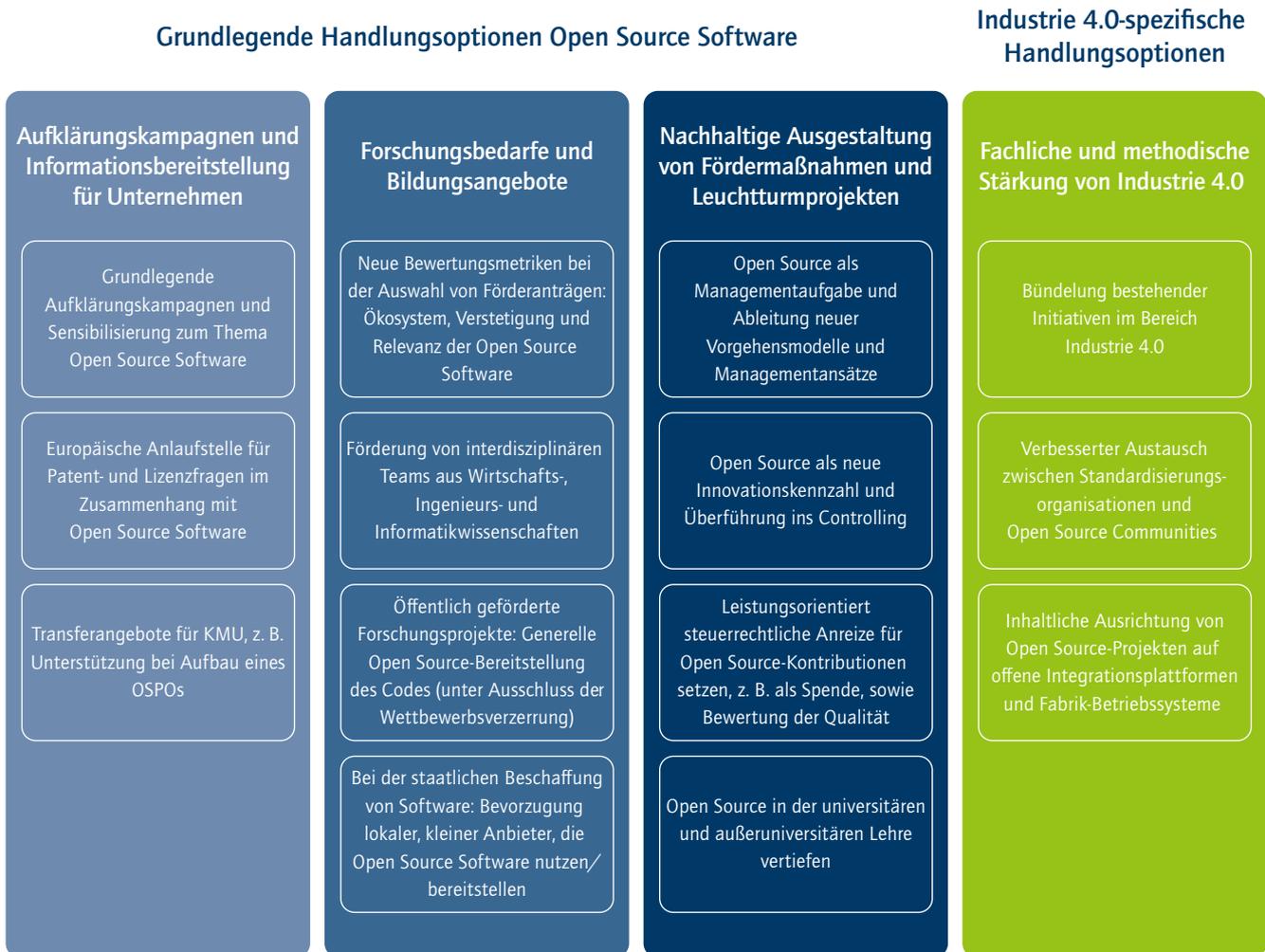
10.4. Stärkung von Organisationsstrukturen und Koordination sowie inhaltliche Ausrichtung von Open Source-Initiativen im Industrie 4.0-Kontext

Neben den eher strukturellen Handlungsoptionen lassen sich für Industrie 4.0 auf fachlicher beziehungsweise methodischer Ebene auch konkrete Handlungsoptionen identifizieren:

1. Eine **Harmonisierung bestehender Industrie 4.0-Initiativen** und eine gemeinschaftliche Fokussierung auf Open Source-Potenziale kann Industrie 4.0 und die darin agierenden Unternehmen stärken. Beispielsweise könnte die Plattform Industrie 4.0 hier eine besondere Rolle einnehmen und Open Source-Aktivitäten bündeln. Auch Verbände, zum Beispiel der VDI oder der Bundesverband der Deutschen Industrie, könnten verstärkt Aufklärung im Bereich OSS leisten und eigene Angebote bereitstellen.
2. Ebenso wäre eine stärkere **Koordination zwischen Standardisierungsorganisationen und Open Source-Initiativen** sowie die Intensivierung ihres Austauschs wünschenswert. Hier könnten etablierte Stiftungen entsprechende Projekte lancieren. Auf Seiten der Standardisierungsorganisationen sollten entsprechende Austauschstellen geschaffen werden.
3. Im Fokus von Industrie 4.0 stehen die horizontale Integration über Wertschöpfungsnetzwerke hinweg, die Vernetzung von Produktionssystemen (vertikale Integration) sowie die Durchgängigkeit des Engineerings entlang des gesamten Lebenszyklus. Als Handlungsoption für die inhaltliche Ausrichtung von Open Source-Entwicklungen, sehen die Erstellerinnen und Ersteller der vorliegenden Expertise aus fachlicher Sicht vor allem folgende Bedarfe: **Entwicklungen im technologischen Schwerpunkt „Konnektivität und Kommunikation“** sind für Industrie 4.0 von großer Bedeutung. Gebraucht werden Industrie 4.0-Softwarebausteine, die auf den Architekturebenen „Informationen“ und „Kommunikation“ des Referenzarchitekturmodells Industrie 4.0 (RAMI 4.0) liegen.¹⁶⁰ Dazu zählen zum Beispiel offene Standards, Daten- und Informationsmodelle sowie quelloffene

160 Vgl. VDI et al. 2015.

Abbildung 10: Handlungsoptionen zur Stärkung von Open Source Software im Industrie 4.0-Kontext



Quelle: eigene Darstellung

Referenzimplementierungen für die Maschine-zu-Maschine-Kommunikation. Weiterer zentraler Handlungsbedarf wird in der **Entwicklung offener Integrationsplattformen und Fabrik-Betriebssysteme** als IT-Backbones für Industrie 4.0 und als Enabler für KI-Anwendungen gesehen.

10.5. Abschließende Betrachtungen und Implikationen für Unternehmen

Die in dieser Expertise aus den Ergebnissen der Analysen und Erörterungen abgeleiteten Handlungsoptionen bestätigen Stimmen anderer Expertinnen und Experten verschiedener Fachrichtungen und Hintergründe zum Thema Open Source. Sie alle kommen zu dem gleichen Schluss: Die Steigerung von Open Source-Beiträgen führt zu Wohlfahrtsgewinnen in einer Volkswirtschaft und sollte daher Teil einer gesunden Wirtschafts- und Innovationspolitik sein. Von Open Source-Entwicklung profitieren vor allem Branchen, die

einen starken Technologiefokus haben – und damit auch Branchen, die ihr Geschäftsmodell im Industrie 4.0-Kontext entwickeln und nachhaltig gestalten möchten. Deshalb haben Industrieverbände und Vereine klar und deutlich zur Bundestagswahl im September 2021 formuliert, dass staatliche Bemühungen um Open Source auszubauen sind. Konkret fordert Bitkom die Gründung eines „National Open Source Program Office“ zur Unterstützung aller Bundesbehörden in ihrer Vorbildfunktion hinsichtlich Open Source und fordert den Ausbau von Informations- und Bildungsangeboten, um „Open Source-Methodiken in den föderalen Strukturen Deutschlands [zu] verstetigen“.¹⁶¹ Auch die bereits vielfach zitierte Studie im Auftrag der Europäischen Union kommt im Schluss zu ähnlichen Handlungsempfehlungen wie die hier vorliegende Expertise.¹⁶² Die Autorinnen und Autoren fordern unter anderem ebenfalls den Ausbau der Open Source-Forschung und Transferleistungen in die Praxis für eine insgesamt höhere Wissensdiffusion. Zudem sehen sie beim öffentlichen Gut OSS ein Marktversagen. Aus diesem Grund fordern sie einen verstärkten Ausbau der staatlichen

161 Vgl. Bitkom 2021.

162 Vgl. Blind et al. 2021.

Förderung sowie die Stärkung einer Open Source-orientierten Verwaltung.

Es ist also zu begrüßen, dass sowohl die Europäische Kommission mit ihrer Open Source-Strategie 2020 – 2023 eigenen Open Source Code bereitstellen und Open Source Communities fördern möchte, als auch die neue Bundesregierung die Relevanz von OSS erkannt und in ihrem Koalitionsvertrag festgeschrieben hat.¹⁶³ Neben der vielfach geforderten Stärkung von Open Source durch die öffentliche Hand als Fördermittelgeber oder Vorbild, sind letztlich jedoch auch Unternehmerinnen und Unternehmer gefragt, sich für Open Source-Strategien zu interessieren, sie zu verstehen und unabhängig von Förderungen, eigenmotiviert zu adaptieren und implementieren.

Insbesondere Unternehmen, die eine Industrie 4.0-Transformation verfolgen, müssen auf OSS setzen, um ihre eigene, aber auch die europäische unternehmerische Souveränität zu sichern. Unternehmen sollten OSS nicht nur aus technologischer Sicht, sondern vor allem aus strategischer und geschäftsmodellrelevanter Perspektive in Erwägung ziehen. Bestehende Hemmnisse und Vorurteile sind abzubauen: Die öffentliche Bereitstellung eigener Software-Entwicklungen ist nicht gleichzusetzen mit der Aufgabe eigenen, marktdifferenzierenden geistigen Kerneigentums. Es bedeutet vielmehr, neue Wege mit engagierten Partnern in der Kollaboration und Entwicklung von Innovationen zu gehen. Nur so – davon sind die Erstellerinnen und Ersteller der vorliegenden Expertise überzeugt – kann die Vision Industrie 4.0 nachhaltig und innovativ Gestalt annehmen. Die angestrebte Vernetzung von allem und jedem, die Generierung und Handelbarkeit neuer Werte und damit verbunden die Sicherung von Wohlstand und nachhaltigem Wachstum kann nur gelingen, wenn Unternehmen national und international zusammenarbeiten und beim gemeinsamen Entwickeln von Open Source Software und Open Hardware Ideen, Konzepte und Innovationen teilen. Initiativen wie die vom Bundesministerium für Digitales und Verkehr geförderte Silicon Economy und die hieraus hervorgegangene Open Logistics Foundation, die im Rahmen der Linux Foundation verortete Open Manufacturing Platform und die OSADL eG sind hierfür drei Beispiele.

Die oben beschriebenen Handlungsoptionen sollen die Rahmenbedingungen für Open Source-Ökosysteme fördern. Die praktische Umsetzung von OSS-Projekten liegt jedoch stets bei den Unternehmen. Das beginnt damit, dass OSS als strategisches Thema fest verankert werden muss. Wie das gelingen kann und welche weiteren Schritte notwendig sind, damit Open Source im Unternehmen zum Innovationstreiber werden kann, damit befasst sich dieser letzte Abschnitt. Abschließend möchten wir die wichtigsten Erkenntnisse dieser Expertise in Bezug auf **unternehmerisches Open Source-Handeln** zusammenfassen und als leitlinienartige Handreichung für Führungskräfte zur Verfügung stellen:

1. **Entwicklung und Kommunikation einer Open Source-Strategie:** Die eigene Open Source-Strategie sollte nicht nur die Frage beantworten, wie und welche OSS im eigenen Unternehmen eingesetzt werden soll, sondern auch, ob das Unternehmen eigene OSS-Beiträge leisten möchte und welche das sein könnten. Dazu sollte erarbeitet werden, welche Motivation und Zielstellung eigene Beiträge haben. Neben der rein intrinsischen Motivation, der Community „etwas zurückgeben zu wollen“, kann der Anreiz für die Mitwirkung an bestimmten Projekten beispielsweise auch in der Sicherung von Technologien, Gestaltung von De-facto-Standards oder der eigenen Geschäftsmodellstrategie bestehen. Die Open Source-Strategie sollte mit einem interdisziplinären Team erarbeitet werden, um innerhalb des Unternehmens eine höchstmögliche Akzeptanz und ein breites Verständnis zu erwirken. Im Anschluss sollte die Strategie klar und verständlich aufbereitet und im Unternehmen kommuniziert werden.
2. **Aufbau eines eigenen Open Source Program Office (OSPO):** Je nach Größe und Struktur des Unternehmens bietet sich der Aufbau eines eigenen OSPO an. Das OSPO ist zentraler Ansprechpartner für alle Fragen zu den Open Source-Prozessen und trägt die Verantwortung für die Qualitätssicherung, aber auch für die Prüfung von Lizenzkompatibilitäten und möglicher Verletzungen geistigen Eigentums. Zudem erarbeitet das OSPO Leitlinien und praktische Vorgehensmodelle für die Open Source-Entwicklung und -Bereitstellung. Darüber hinaus ist es sinnvoll, im Rahmen des OSPO und in Abstimmung mit der Geschäftsführung zu definieren, welche Freiheitsgrade und Befugnisse Mitarbeitende und externe Beitragende im Kontext der gemeinschaftlichen Entwicklung haben. Auch kleinere Unternehmen, ohne große oder eigene Entwicklungsabteilungen, sollten feste Zuständigkeiten und Ansprechpersonen für die Open Source-Entwicklung in ihren Organisationen benennen. Parallel beziehungsweise durchaus auch schon vor der Gründung eines OSPO sind Qualifizierungsmaßnahmen für den Kompetenzaufbau der involvierten Beschäftigten entscheidend.
3. **Strategische Auswahl von Ökosystemen und Communities:** Eine entscheidende Aufgabe eines OSPO oder der entsprechenden Fachabteilungen ist die Bewertung und Auswahl geeigneter Projekte und der damit verbundenen Ökosysteme beziehungsweise Communities. Welche wesentlichen Messgrößen und Bewertungskriterien für beispielsweise die Erhaltungsfähigkeit, Nachhaltigkeit und Prozessreife eines Ökosystems zur Verfügung stehen, ist in Tabelle 3 dargestellt. Eine aktive und ausreichend große Community ist der wichtigste Erfolgsfaktor für eine zielgerichtete und nachhaltige Entwicklung. Sie steht für Code-Qualität und die Langlebigkeit des Projekts. Dementsprechend sollten Unternehmen ex ante sehr genau prüfen, welchen Projekten sie beitreten, in welchen Communities sie selbst Beiträge leisten möchten, aber auch welche Entwicklungen sie nutzen wollen.

163 Vgl. Koalitionsvertrag 2021, Europäische Kommission 2020b.

- 4. Schulung und Etablierung eines neuen Mindsets:** Entwicklerinnen und Entwickler sind durch ihre alltägliche Arbeit durchaus mit OSS vertraut. Oftmals jedoch fehlt es an einem tiefgehenden Verständnis für die Sicherstellung der Software-Qualität in der gemeinsamen Entwicklung innerhalb von Open Source-Ökosystemen sowie für die Wahrung der entsprechenden Coding Guidelines. Zudem sind Entwicklerinnen und Entwickler auch nicht mit den möglichen Lizenzproblematiken und -konflikten vertraut. Dementsprechend ist es die Aufgabe eines OSPO oder der Open Source-Verantwortlichen, Entwicklerinnen und Entwickler zu befähigen, eigenständig beziehungsweise in Absprache OSS-Entwicklungsprozesse durchzuführen. Dazu gehören neben der Methodik auch ein grundlegendes juristisches Verständnis für Lizenzen und Urheberrechtsfragen. Auch der Blick über den Tellerrand sollte von den Verantwortlichen initiiert werden: Im Rahmen von Schulungen sollte die Bedeutung und Relevanz von OSS entlang der Unternehmensstrategie fach- und abteilungsübergreifend vermittelt werden. Insbesondere für Industrie 4.0-Unternehmen ist die gemeinsame Entwicklung eines Verständnisses für neue Geschäftsmodelle, Methoden für Innovationsprozesse und Kollaborationsformate entscheidend. Insofern erfordert der Einsatz und die Bereitstellung von OSS ein neues Mindset in allen Unternehmensbereichen. Die Kommunikation von Open Source-Leuchtturmprojekten und Einbindung von erfahrenen Personen als Multiplikatoren innerhalb des Unternehmens können dabei helfen. Eigene Pilotprojekte können zum Beispiel anfänglich mit vertrauten Partnerunternehmen oder betrieblichen Kunden im kleinen Rahmen realisiert werden.
- 5. Kontinuierliche Anpassung und Hinterfragung des Geschäftsmodells:** Unternehmen stehen heutzutage oftmals in einem schnelllebigem und anspruchsvollem Wettbewerbsumfeld. Das trifft in besonderem Maße auch auf Industrie 4.0-Unternehmen zu. Innerhalb kurzer Abstände treten neue, oft internationale Unternehmen mit ihren innovativen Technologien in den Markt. Um wettbewerbsfähig zu bleiben, müssen Unternehmen kontinuierlich ihre Geschäftsmodelle hinterfragen und an neue Kundenbedürfnisse anpassen. Dies hat einen direkten Einfluss auf die Open Source-Strategie, denn geistiges Kerneigentum sollte nicht ohne guten Grund quelloffen gestellt werden. Produkte, Prozesse und Code, die nicht marktdifferenzierend sind, eignen sich hingegen für Open Source-Entwicklungen (siehe Abschnitt 8.1) Das gilt auch für Prozesse, Services und Produkte, die das primäre Wertversprechen komplementieren. Darum müssen Unternehmen fortwährend analysieren und verstehen, worin ihr primäres Wertversprechen und geistiges Kerneigentum besteht und wie es am Markt bestehen kann. Entsprechend muss auch die Open Source-Strategie ausgelegt und kontinuierlich und über alle Unternehmensbereiche hinweg angepasst werden. Das Zusammenspiel von betriebswirtschaftlicher, technologischer und unternehmensstrategischer Sicht ist hier entscheidend.

Anhang

Glossar

Begriff	Beschreibung
5G	Ein Mobilfunkstandard der fünften Generation, der die Normen und Regelungen zur mobilen Kommunikation definiert.
Adopter Community	Die Adopter Community stellt eine Spezialform der Gemeinschaft der Anwendenden dar. Darunter subsumiert werden Personen, die die angebotenen Frameworks und Anwendungen nicht nur nutzen, sondern innerhalb ihrer Unternehmen in ihre Unternehmens-IT einbetten und Erweiterungen erstellen.
AGPL	Open Source-Lizenz mit starken Copyleft-Klauseln
AI Augmented Development	Entwicklungsprozesse, die auf der Nutzung von Entwicklertools mit integrierter künstlicher Intelligenz basieren.
AI Developer and Teaching Kits	Entwicklertools und Unterrichtskits, mit denen AI entwickelt beziehungsweise gelehrt werden kann.
Apache 2.0-Lizenz	Open Source-Lizenz ohne Copyleft-Klauseln mit weiter Verbreitung.
Application Programming Interface	Eine Programmierschnittstelle oder Anwendungsschnittstelle, das heißt ein Programmteil, das die Verbindung zwischen zwei Systemen zwecks Datenübergabe herstellt.
Artificial General Intelligence	Künstliche allgemeine Intelligenz, die hypothetische Intelligenz eines Computerprogramms.
Asterisk	Eine Open Source Software, die die Funktionalität einer Telefonanlage bietet.
Augmented Intelligence	Eine Technologie, bei der menschliches Urteilsvermögen und künstliche Intelligenz bei der Bewältigung von Entscheidungsproblemen zusammenarbeiten.
Augmented Reality	Die computergestützte Erweiterung der Realität durch die visuelle Darstellung von Informationen.
BSD-Lizenz	Gruppe freizügiger Open Source-Lizenzen ohne Copyleft-Klauseln.
Closed Source Software	Software, deren Quellcode nicht öffentlich einsehbar ist.
Code of Conduct	Verhaltenskodex, welcher die Umgangsformen einer Open Source Community regelt.
Cognitive Computing	Die Simulation von menschlichen Denkprozessen in einem Computermodell.
Commit	Beitrag zu Open Source-Projekten in Form von Code-Beiträgen
Common Development and Distribution-Lizenz	Open Source-Lizenz, die Copyleft-Klauseln enthält.
Continuous Delivery	Eine Software-Entwicklungsmethode, bei der Code-Änderungen automatisch getestet und für eine Produktionsversion vorbereitet werden, um den Software-Auslieferungsprozess zu verbessern.
Continuous Deployment	Eine weitergehende Form des Continuous Delivery, bei der die Auslieferung der Software kontinuierlich ausgeführt wird.
Continuous Integration	Fortlaufender Prozess des Zusammenfügens von Änderungen verschiedener Kontributionen in ein gemeinsames Projekt.
Copyleft-Lizenz	Lizenz, die die den Lizenznehmer verpflichtet, jegliche Bearbeitung des Werks unter die Lizenz des ursprünglichen Werks zu stellen.
Cyber-physisches System	Die Vernetzung von physischer Produktionsumgebung oder von Produkten mit Softwareprogrammen, die autonom miteinander kommunizieren und interagieren können.
De-facto-Standard	Ein nicht offiziell von der Standardisierungsorganisation definierter Standard, der aber aufgrund seiner Verbreitung weitgehend eingehalten wird.
De-jure-Standard	Ein von der Standardisierungsorganisation rechtlich definierter Standard.
Duale Lizenzierung	Eine Software ist unter zwei Softwarelizenzen lizenziert.
Edge AI	Edge AI Devices beschreibt KI-Algorithmen zur lokalen Datenverarbeitung, die ohne externe Speicher- und Rechenleistungen auskommen.
Embedded Systems	Ein eingebettetes System ist ein Computersystem, das eine spezielle Funktion innerhalb eines größeren Systems erfüllt. Dies kann zum Beispiel eine Überwachungs-, Steuerungs-, oder Regelfunktion sein.
Entwicklungs-Communities	Entwicklungs-Communities sind die Schlüsselkomponente eines jeden Open Source-Projekts. Zu ihnen gehören die Autoren beziehungsweise Eigner von Open Source-Projekten sowie Personen und Unternehmen, die zur Entwicklung des Projekts beitragen.
GNU	GNU ist ein quelloffenes Betriebssystem, das im Rahmen des 1984 gestarteten GNU-Projekts. Es wird seither als Softwareasammlung von Anwendungen und Bibliotheken entwickelt.
GNU General Public Licence	Open Source Softwarelizenz, die ursprünglich für das GNU-Projekt geschrieben wurde. Sie ist die erste Copyleft-Lizenz für den allgemeinen Gebrauch.
Governance Modelle	Regeln, Gepflogenheiten und Prozesse, die festlegen, welche Mitwirkenden die Befugnis haben, bestimmte Aufgaben auszuführen, werden in Governance-Modellen formuliert und formalisiert.
Horizontale Integration	Flexible Integration von IT-Systemen und Akteuren in ein Wertschöpfungsnetzwerk über unterschiedliche Prozessschritte hinweg.

Begriff	Beschreibung
Hype Cycle	Der Hype Cycle stellt die Phasen der Aufmerksamkeit, die eine neue Technologie durchläuft, dar. Die vom Anbieter von Marktforschungsergebnissen Gartner herausgegebenen Hype Cycles benennen die Phasen Technologischer Auslöser, Gipfel der überzogenen Erwartungen, Tal der Enttäuschungen, Pfad der Erleuchtung und Plateau der Produktivität.
Incorporation	Unter Incorporation versteht man die Möglichkeit, Teile von OSS in eigene Software zu integrieren.
ISO/IEC-Standards 5230:2020	ISO/IEC 5230 ist ein internationaler Standard zu den Anforderungen an ein Compliance-Programm für Open Source-Projekte.
Keras	Keras ist eine in Python geschriebene Open Source Softwarebibliothek für künstliche Intelligenz.
Linking	Beim Linking wird eine Open Source-Komponente mit einer eigenen Komponente verknüpft.
MIT-Lizenz	Open Source-Lizenz ohne Copyleft-Klauseln mit weiter Verbreitung.
Mehrfachlizenzierung	Eine Software ist unter mehreren Softwarelizenzen lizenziert.
Meritokratie	Prinzip in Open Source-Communities, bei der Einfluss durch ständige und wertvolle Beiträge verdient wird.
Modification	Bei einer Modification wird der ursprüngliche Quellcode einer Open Source-Komponente angepasst beziehungsweise verändert.
Mozilla Public-Lizenz	Open Source-Lizenz, die schwache Copyleft-Klauseln enthält.
MySQL	Ein relationales Datenverwaltungssystem.
OpenChain-Projekt	Das OpenChain-Projekt pflegt den internationalen Standard für die Einhaltung von Open Source-Richtlinien.
Open Innovation	Die Öffnung von Innovationsprozessen, um externe Ideen und Innovationen zu generieren.
Open Source Program Offices	OSPOs sind ein in Unternehmen weit verbreitetes Konstrukt zur Organisation und Koordinierung von Open-Source-Programmaktivitäten
Open Source Repository	Speicherort von beziehungsweise für Open Source-Projekte, zum Beispiel GitHub
Open Standards	Standards, die für alle Marktteilnehmer besonders leicht zugänglich, weiterentwickelbar und einsetzbar sind. Von verschiedenen Open Source-Organisationen wird angestrebt, dass Open Standards nur solche Standards bezeichnen sollten, die ohne Patentverletzung unter einer Open Source-Lizenz implementiert werden können.
Permissive-Lizenz	Lizenzen, die es ermöglichen, abgeleitete oder distribuierte Werke unter eine Lizenz zu stellen, die nicht die ursprüngliche Lizenz sein muss.
Platform as a Service	Eine Cloudumgebung, die eine Plattform für die Entwicklung von Anwendungen bereitstellt.
Proof of Concept	Ein Proof of Concept (PoC) dient der Überprüfung der Umsetzbarkeit einer Idee in der Realität. Ein Proof of Concept soll die Machbarkeit der Idee beweisen oder zeigen, ob die Idee wie geplant funktioniert. Andere geläufige Bezeichnungen sind Proof of Principle und Machbarkeitsnachweis.
PyTorch	Ein Open Source Framework für maschinelles Lernen, das auf der Torch-Bibliothek basiert.
Quantum	Eine Open Source Foundation, die die Entwicklung und Standardisierung von Open Source Tools für die Anwendung in Quantencomputern fördert.
Smart Contracts	Intelligente auf der Blockchain-Technologie basierende Verträge, die ohne menschliche Überwachung ausgeführt werden können.
Software Development Kit	Verschiedene Programmierwerkzeuge und Programmbibliotheken, die bei der Entwicklung einer Software unterstützen.
Source Code Repository	Speicherort oder Archiv, wo Quellcode über eine Versionsverwaltung strukturiert und gespeichert wird.
Sprunginnovation	Innovation, die bestehende Produkte, Dienstleistungen und auch Märkte vollständig verändert und neue Märkte, Netzwerke, Nutzenwelten und damit auch Geschäftsfelder entstehen lässt. Eine Sprunginnovation ist beispielsweise das Internet. Ein anderer Begriff dafür ist „disruptive Innovation“.
Swarm Robotics	Autonom agierende Schwärme von Robotern.
SWOT-Analyse	Die Stärken-Schwächen-Chancen-Risiko-Analyse stellt ein Instrument der strategischen Unternehmensplanung dar. Sie wird zur internen Unternehmens- sowie der externen Umfeldanalyse angewandt.
TensorFlow	Ein Open Source Ende-zu-Ende-Framework für maschinelles Lernen und künstliche Intelligenz.
Translation	Nutzung und Umwandlung von Open Source-Komponenten
Vertikale Integration	Datendurchgängigkeit zwischen IT-Systemen für ganzheitliche Vernetzung von der Produktionsebene bis hin zu höheren Ebenen.
Virtual Reality	Eine computergenerierte Wirklichkeit in einer interaktiven virtuellen Umgebung. Diese wird über Bildschirme, spezielle Räume oder Technik, zum Beispiel VR-Brillen, übertragen.
Wi-Fi 6	Die Bezeichnung der Wi-Fi Alliance für den IEEE 802.11ax Standard, der drahtlose lokale Funknetze standardisiert.

Abbildungen

Abbildung 1:	Open Source und Cloud Computing	6
Abbildung 2:	Open Source durchdringt immer weitere Bereiche	8
Abbildung 3:	Technologiebereiche von Industrie 4.0	11
Abbildung 4:	Möglichkeiten der Verwendung von Open Source-Komponenten	16
Abbildung 5:	Basisfunktionen besitzen eine hohe Open Source-Eignung	19
Abbildung 6:	Im Zeitverlauf ansteigende Open Source-Beteiligung von Organisationen	22
Abbildung 7:	Beispielhafter hierarchischer Aufbau einer Entwicklungs-Community	23
Abbildung 8:	Beispiel des organisationalen Zusammenspiels von Stiftung, Projekten und Unternehmen im Open Source-Kontext	24
Abbildung 9:	Archetypen von Open Source-Projekten mit Beispielen aus Industrie 4.0 und darüber hinaus	32
Abbildung 10:	Handlungsoptionen zur Stärkung von Open Source Software im Industrie 4.0-Kontext	64

Tabellen

Tabelle 1:	Sechsstufiges Vorgehen der Expertise	9
Tabelle 2:	Befragte Expertinnen und Experten (anonymisiert)	10
Tabelle 3:	Gängige Open Source-Messgrößen	28
Tabelle 4:	Taxonomie zur Beschreibung und Analyse von Open Source-Projekten im Kontext von Industrie 4.0	30
Tabelle 5:	Produktangebote von Open Source-Projekten	31
Tabelle 6:	Einordnung von Apache Camel und PLC4X in die Open Source-Taxonomie für Industrie 4.0	35
Tabelle 7:	Einordnung von AWS IoT Device SDKs in die Open Source-Taxonomie für Industrie 4.0	36
Tabelle 8:	Einordnung von BaSyx in die Open Source-Taxonomie für Industrie 4.0	37
Tabelle 9:	Einordnung von Tractus-X in die Open Source-Taxonomie für Industrie 4.0	38
Tabelle 10:	Einordnung von Eclipse IoT in die Open Source-Taxonomie für Industrie 4.0	40
Tabelle 11:	Einordnung von FIWARE in die Open Source-Taxonomie für Industrie 4.0	41
Tabelle 12:	Einordnung von IndustryFusion in die Open Source-Taxonomie für Industrie 4.0	42
Tabelle 13:	Einordnung des Eclipse Data Space Connector der Initiative International Data Spaces in die Open Source-Taxonomie für Industrie 4.0	43
Tabelle 14:	Einordnung von open62541 in die Open Source-Taxonomie für Industrie 4.0	44
Tabelle 15:	Einordnung der Open Manufacturing Platform in die Open Source-Taxonomie für Industrie 4.0	45
Tabelle 16:	Einordnung des Open Source Automation Development Labs in die Open Source-Taxonomie für Industrie 4.0	46
Tabelle 17:	Einordnung von ROS in die Open Source-Taxonomie für Industrie 4.0	47
Tabelle 18:	SWOT-Analyse Open Source Software im Kontext von Industrie 4.0	61

Literatur

acatech 2021

acatech – Deutsche Akademie der Technikwissenschaften (Hrsg.): *Impulse zur Innovationspolitik 2021+*, 2021. URL: www.acatech.de/publikation/impulse-zur-innovationspolitik-2021/ [Stand: 16.12.2021].

AlMarzouq et al. (2005)

AlMarzouq, M./Zheng, L./Rong, G./Grover, V.: *Open Source: Concepts, Benefits, and Challenges*. In: *CAIS*, 16, doi:10.17705/1CAIS.01637, 2005, S. 505–521.

Aniszczuk 2021

Aniszczuk, C.: *OSPO 101 Training Modules*, 2021. URL: <https://github.com/todogroup/ospo101> [Stand: 01.12.2021].

Apache 2021a

The Apache Software Foundation: *Apache PLC4X. The universal protocol adapter for Industrial IoT*, 2021. URL: <https://plc4x.apache.org/> [Stand: 07.01.2022].

Apache 2021b

The Apache Software Foundation: *Current language support for protocols*, 2021. URL: <https://plc4x.apache.org/users/protocols/index.html> [Stand: 07.01.2022].

Apache 2022

The Apache Software Foundation: *What is Camel?*, 2022. URL: <https://camel.apache.org/manual/faq/what-is-camel.html> [Stand: 07.01.2022].

Apache IoTDB 2022

Apache IoTDB: *Apache IoTDB. Database for Internet of Things*, 2022, URL: <https://iotdb.apache.org/> [Stand: 08.03.2022].

Apache StreamPipes 2019

Apache StreamPipes: *Apache StreamPipes (incubating)*, 2019. URL: <https://streampipes.apache.org/> [Stand: 08.03.2022].

Appleyard/Chesbrough 2016

Appleyard, M.M./Chesbrough, H.W.: „The Dynamics of Open Strategy: From Adoption to Reversion“. In: *Long Range Planning*, 50: 3, 2017, S. 310–321.

AWS 2022a

Amazon Web Services: *AWS IoT Device SDKs, Mobile SDKs, and AWS IoT Device Client*, 2022. URL: <https://docs.aws.amazon.com/iot/latest/developerguide/iot-sdks.html> [Stand: 10.01.2022].

AWS 2022b

Amazon Web Services: *AWS IoT: Entsperren Sie Ihre IoT-Daten und beschleunigen Sie das Unternehmenswachstum*, 2022. URL: <https://aws.amazon.com/de/iot/> [Stand: 08.03.2022].

Bagozzi/Dholakia 2006

Bagozzi, R.P./Dholakia, U.M.: „Open Source Software User Communities: A Study of Participation in Linux User Groups“. In: *Management Science*, 52: 7, S. 1099–1115.

BaSys 4.0 2022a

Basissystem Industrie 4.0 (BaSys 4.0): *Eine offene Plattform für die vierte industrielle Revolution*, 2022. URL: www.basys40.de/ [Stand: 10.01.2022].

BaSys 4.0 2022b

Basissystem Industrie 4.0 (BaSys 4.0): *Satellitenprojekte für die Weiterentwicklung von BaSys 4 in der Anwendung*, 2022. URL: www.basys40.de/satellitenprojekte/ [Stand: 10.01.2022].

Benkler 2002

Benkler, Y.: „Coase's Penguin, or, Linux and 'The Nature of the Firm'“. In: *The Yale Law Journal*, 112: 3, 2002, S. 369–446.

Bitkom 2016

Bitkom e.V. (Hrsg.): *Open Source Software 2.0. Leitfaden*, 2016. URL: www.bitkom.org/sites/default/files/file/import/FirstSpirit-1498131485664160229-OSS-Open-Source-Software.pdf [Stand: 09.12.2021].

Bitkom 2019

Bitkom e.V. (Hrsg.): *Open Source Monitor. Research Report 2019*. URL: www.bitkom.org/sites/default/files/2020-04/200420_eng_bitkom_report_openmonitor_2019.pdf [Stand: 06.09.2021].

Bitkom 2021

Bitkom e.V.: *Bundestagswahl 2021: Open Source*, 2021. URL: www.bitkom.org/Bundestagswahl-2021-Open-Source [Stand: 09.12.2021].

Bitkom et al. 2015

Bitkom e.V./VDMA e.V./ZVEI e.V. (Hrsg.): *Umsetzungsstrategie Industrie 4.0: Ergebnisbericht der Plattform Industrie 4.0*, Berlin, 2015. URL: www.its-owl.de/fileadmin/PDF/Industrie_4.0/2015-04-10_Umsetzungsstrategie_Industrie_4.0_Plattform_Industrie_4.0.pdf [Stand: 10.12.2021].

Blind/Böhm 2019

Blind, K./Böhm, M.: *JRC Science for Policy Report. The Relationship Between Open Source Software and Standard Setting*, Thumm, N. (Hrsg.), EUR 29867 EN, Luxembourg: Publications Office of the European Union 2019. URL: https://publications.jrc.ec.europa.eu/repository/bitstream/JRC117836/jrc_the_relationship_between_open_source_software_final_online_compressed_logo.pdf [Stand: 15.02.2022].

Blind et al. 2021

Blind, K./Böhm, M./Grzegorzewska, P./Katz, A./Muto, S./Pätsch, S./Schubert, T.: *The impact of Open Source Software and Hardware on technological independence, competitiveness and innovation in the EU economy, Final Study Report*, 2021. URL: <https://op.europa.eu/en/publication-detail/-/publication/29effe73-2c2c-11ec-bd8e-01aa75ed71a1/language-en> [Stand: 20.10.2021].

Böhm 2019

Böhm, M.: „The emergence of governance norms in volunteer driven open source communities“. In: *Journal of Open Law, Technology, & Society*, 9: 1, 2019, S. 3–39.

Chesbrough 2010

Chesbrough, H.W.: *Open innovation; The new imperative for creating and profiting from technology*, Boston, Mass.: Harvard Business School Press 2010.

Crowston et al. 2006

Crowston, K./Howison, J./Annabi, H.: „Information systems success in free and open source software development: theory and measures“. In: *Software Process: Improvement and Practice (Special Issue on Free/Open Source Software Processes)*, 2006, S. 123–148.

Curucu 2018

Curucu, S.: *The Apache Way: Community over code*, 2018. URL: <http://theapacheway.com/community-over-code/#:~:text=In%20brief%2C%20this%20means%20that,part%20of%20how%20we%20work> [Stand: 10.01.2022].

Dahlander/Magnusson 2005

Dahlander, L./Magnusson, M.: „Relationships between open source software companies and communities: Observations from Nordic firms“. In: *Research Policy*, 34: 4, 2005, S. 481–493.

Dahlander/Magnusson 2008

Dahlander, L./Magnusson, M.: *How do Firms Make Use of Open Source Communities? In: Long Range Planning*, 41, 2008, S. 629–649.

Dahlander/Wallin 2006

Dahlander, L./Wallin, M.W.: „A Man on the Inside: Unlocking Communities as Complementary Assets“. In: *Research Policy*, 35, 2006, S. 1243–1259.

Debian 2022

Debian: *Package mosquito*, 2022. URL: www.eclipse.org/paho/index.php?page=downloads.php [Stand 10.01.2022].

Dudley et al. 2007

Dudley, A./Hong, S./Wang, K.: „The Social and Economical Impact of OSS in Developing Countries“. In: St. Amant, K./Still, B. (Hrsg.): *Handbook of Research on Open Source Software: Technological, Economic, and Social Perspectives*, IGI Global: 2007, S. 102–114.

Eclipse BaSyx 2017

Eclipse Foundation: *Eclipse BaSyx Creation Review*, 2017. URL: <https://projects.eclipse.org/projects/technology.basysx/reviews/creation-review> [Stand:10.01.2022].

Eclipse Dataspace Connector 2022

Eclipse Foundation: *Eclipse Dataspace Connector. Overview*. URL: <https://projects.eclipse.org/projects/technology.dataspaceconnector> [Stand: 10.01.2022].

Eclipse Foundation 2020

Eclipse Foundation: *Eclipse Foundation Development Process*, 2020. URL: www.eclipse.org/projects/dev_process/#2_3_Three_Communities [Stand: 10.01.2022].

Eclipse IoT 2022

Eclipse Foundation: *Eclipse IoT Overview*. URL: <https://projects.eclipse.org/projects/iot/%20charter>. [Stand: 10.01.2022].

Eclipse Paho 2019

Eclipse Foundation: *Eclipse Paho Overview, 2019*. URL: <https://projects.eclipse.org/projects/iot.paho> [Stand: 10.01.2022].

Eclipse Paho 2022

Eclipse Foundation: *Eclipse Paho Downloads*. URL: www.eclipse.org/paho/index.php?page=downloads.php [Stand: 10.01.2022].

Eclipse Vorto 2020a

Eclipse Foundation: *Eclipse Vorto Overview*. URL: <https://projects.eclipse.org/projects/iot.vorto> [Stand: 10.01.2022].

Eclipse Vorto 2020b

Eclipse Foundation: *Eclipse Vorto Governance*. URL: <https://projects.eclipse.org/projects/iot.vorto/governance> [Stand: 10.01.2022].

FIWARE 2022

FIWARE: *Fiware Contribution Requirements*. URL: <https://fiware-requirements.readthedocs.io/en/latest/> [Stand: 10.01.2022].

Elhacham et al. 2020

Elhacham, E./Ben-Uri, L./Grozovski, J./Bar-On, Y.M./Milo, R.: „Global human-made mass exceeds all living biomass“. In: *Nature*, 588, 2020, S. 442–444.

European Commission 2019

European Commission: *100 Radical Innovation Breakthroughs for the future*, 2019. URL: [ec_rtd_radical-innovation-breakthrough_052019.pdf](http://ec.rtd.radical-innovation-breakthrough_052019.pdf) (europa.eu) [Stand: 07.12.2021].

Europäische Kommission 2020a

Europäische Kommission: *Allgemeine Gruppenfreistellungsverordnung*, 2020. URL: https://eur-lex.europa.eu/legal-content/DE/TXT/HTML/?uri=LEGISSUM:0802_4 [Stand: 10.01.2022].

Europäische Kommission 2020b

Europäische Kommission: *Mitteilung an die Kommission. Open-Source-Software-Strategie 2020–2023. Offen Denken*. Brüssel: 2020. URL: https://ec.europa.eu/info/sites/default/files/de_ec_open_source_strategy_2020-2023.pdf [Stand: 09.12.2021].

FabOS 2022

FabOS: *FabOS – offenes, verteiltes, echtzeitfähiges und sicheres Betriebssystem für die Produktion*, 2022. URL: www.fab-os.org/ [Stand: 08.03.2022].

Finley 2016

Finley, K.: *Open Source Won. So, Now What? Open source now runs the world. But it still faces problems*, 2016. URL: www.wired.com/2016/08/open-source-won-now/?GuidesLearnMore/ [Stand: 07.12.2021].

Fontana/Frey 2000

Fontana, A./Frey, J.H.: „The Interview. From Structured Questions to Negotiated Text“. In: *Handbook of Qualitative Research*, 2, 2000, S. 645–672.

Forschungsunion/acatech 2013

Promotorengruppe Kommunikation der Forschungsunion Wirtschaft – Wissenschaft/acatech – Deutsche Akademie der Technikwissenschaften (Hrsg.): *Deutschlands Zukunft als Produktionsstandort sichern. Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0. Abschlussbericht des Arbeitskreises Industrie 4.0*, 2013. URL: www.acatech.de/wp-content/uploads/2018/03/Abschlussbericht_Industrie4.0_barrierefrei.pdf [Stand: 15.02.2022].

Forschungsbeirat/acatech 2019a

Forschungsbeirat der Plattform Industrie 4.0 / acatech – Deutsche Akademie der Technikwissenschaften (Hrsg.): *Vorstudie zur Entwicklung einer bedarfs- und nutzergerechten Unterstützung von KMU bei der Einführung und Anwendung von Industrie 4.0*, 2019. URL: www.acatech.de/publikation/unterstuetzung-von-kmu-auf-dem-weg-zur-industrie-4-0/ [Stand: 10.12.2021].

Forschungsbeirat/acatech 2019b

Forschungsbeirat der Plattform Industrie 4.0 / acatech – Deutsche Akademie der Technikwissenschaften (Hrsg.): *Schneller zum Markterfolg Memorandum des Forschungsbeirats der Plattform Industrie 4.0 für ein agileres und flexibleres Innovationssystem in Deutschland*, 2019. URL: www.acatech.de/publikation/memorandum-des-forschungsbeirats/ [Stand: 07.12.2021].

Franco-Bedoya et al. 2015

Franco-Bedoya, O./Ameller, D./Costal, D./Franch, X.: „Measuring the Quality of Open Source Software Ecosystems Using QuESo“. In: Holzinger, A./Cardoso, J./Cordeiro, J./Libourel, T./Maciaszek, L.A./van Sinderen, M. (Hrsg.): *Software Technologies*, Cham: Springer International Publishing 2015, S. 39–62.

Free Software Foundation 2021

Free Software Foundation: *What is Copyleft?* 2021. URL: www.gnu.org/licenses/copyleft [Stand: 31.08.2021].

Gartner 2019

Gartner, Inc.: *Technology Insight for Software Composition Analysis*, 2019. URL: www.gartner.com/en/documents/3971011/technology-insight-for-software-composition-analysis [Stand: 10.12.2021].

Gartner 2020a

Gartner, Inc.: *Hype Cycle for Open-Source Software*, 2020. URL: www.gartner.com/en/documents/4004007 [Stand: 10.12.2021].

Gartner 2020b

Gartner, Inc.: *5 Trends bestimmen den Gartner Hype Cycle for Emerging Technologies 2020*. URL: www.gartner.de/de/artikel/5-trends-bestimmen-den-gartner-hype-cycle-for-emerging-technologies-2020 [Stand: 07.12.2021].

GitHub 2021a

GitHub: *Open Source Program Office (OSPO) 2021 Survey*. URL: https://github.com/todogroup/osposurvey/blob/master/2021/results_2021.pdf [Stand: 10.01.2022].

GitHub 2021b

GitHub: *Open Source Program Office (OSPO) Definition and Guide*, 2021. URL: <https://github.com/todogroup/ospodefinition.org> [Stand: 10.01.2022].

GitHub 2021c

GitHub: *Measuring your open source program's success*, 2021. URL: <https://github.com/todogroup/todogroup.org/blob/main/content/en/guides/measuring.md#how-to-define-success> [Stand: 10.01.2022].

GitHub 2021d

GitHub: *Eclipse Tractus-X*, 2021. URL: <https://github.com/eclipse/tractusx> [Stand: 10.01.2022].

GitHub 2021e

GitHub: *Open62541*, 2021. URL: <https://github.com/open62541/open62541> [Stand: 10.01.2022].

GitHub 2021 f

GitHub: *OMP OPC UA Edge Connector*, 2021. URL: <https://github.com/OpenManufacturingPlatform/iotcon-opc-ua-connector-dotnet> [Stand: 10.01.2022].

GitHub 2022a

GitHub: *Guidebook of open source community management best practices*, 2022. URL: <https://github.com/theopensourceway/guidebook/> [Stand: 10.01.2022].

GitHub 2022b

GitHub: *OSPO 101 Training Modules*, 2022. URL: <https://github.com/todogroup/ospo101> [Stand: 10.01.2022].

GitHub 2022c

GitHub: *Contributions to main, excluding merge commits and bot accounts*, 2022. URL: <https://github.com/apache/camel/graphs/contributors> [Stand: 10.01.2022].

GitHub 2022d

GitHub: *Next generation AWS IoT Client SDK for C++ using the AWS Common Runtime*, 2022. URL: <https://github.com/aws/aws-iot-device-sdk-cppv2> [Stand: 10.01.2022].

GitHub 2022e

GitHub: *Fiware*, 2022. URL: <https://github.com/Fiware> [Stand: 10.01.2022].

GitHub 2022f

GitHub: *IndustryFusion Foundation*, 2022. URL: <https://github.com/IndustryFusion> [Stand: 10.01.2022].

GitHub 2022g

GitHub: *International Data Spaces Association*. URL: <https://github.com/International-Data-Spaces-Association> 2022. [Stand: 10.01.2022].

Glass/Vessey 1995

Glass, R.L./Vessey, I.: „Contemporary Application-Domain Taxonomies“. In: *IEEE Software*, 12, 1995, S. 63–76.

Open62541

Open62541: *open62541 is an open source C (C99) implementation of OPC UA licensed under the Mozilla Public License v2.0*. URL: <https://open62541.org/> [Stand: 10.01.2022].

Hecker 1999

Hecker, F.: „Setting up shop: The business of open-source software“. In: *IEEE Software*, 16, 1999, S. 45–51.

Henkel 2006

Henkel, J.: „Selective revealing in open innovation processes: The case of embedded Linux“. In: *Research Policy*, 35, 2006, S. 953–969.

Huber 2016

Huber, W.: *Industrie 4.0 in der Automobilproduktion. Ein Praxisbuch*, Wiesbaden: Springer Vieweg 2016.

IBM 1983

IBM: *Distribution of IBM licensed programs and licensed program materials and modified agreement for IBM licensed programs*, (IBM Announcement Letter Number 283-016 vom 08.02.1983). URL: www.landley.net/history/mirror/ibm/oco.html [Stand: 07.12.2021].

IndustryFusion 2022

IndustryFusion: *Industry Fusion. Die herstellerübergreifende Open-Source Vernetzungslösung für Smart Factories und Smart Products*, 2022. URL: www.industry-fusion.com/ [Stand: 10.01.2022].

IfrOSS 2021

IfrOSS: Lizenz-Center, 2021. URL: <https://ifrOSS.github.io/ifrOSS/Lizenzcenter> [Stand: 10.12.2021].

Iskoujina/Roberts 2015

Iskoujina, Z./Roberts, J.: „Knowledge sharing in open source software communities: motivations and management“. In: *Journal of Knowledge Management*, 19, 2015, S. 791–813.

Izquierdo/Cabot 2018

Izquierdo, J.L.C./Cabot, J.: „The role of foundations in open source projects“. In: Issarny, V. (Hrsg.) *Proceedings of the 40th International Conference on Software Engineering Software Engineering in Society*, New York: ACM 2018, S. 3–12.

Jansen 2014

Jansen, S.: „Measuring the health of open source software ecosystems: Beyond the scope of project health“. In: *Information and Software Technology*, 56, 2014, S. 1508–1519.

Johnson 2021

Johnson, P.: *Open Source Licenses in 2021: Trends and Predictions*, 2021. URL: www.whitesourcesoftware.com/resources/blog/open-source-licenses-trends-and-predictions/ [Stand: 10.12.2021].

Koalitionsvertrag 2021

Mehr Fortschritt wagen. Bündnis für Freiheit, Gerechtigkeit und Nachhaltigkeit. Koalitionsvertrag 2021–2025 zwischen SPD, Bündnis 90/Die Grünen und FDP, 24.11.2021. URL: www.spd.de/fileadmin/Dokumente/Koalitionsvertrag/Koalitionsvertrag_2021-2025.pdf [Stand: 09.12.2021].

Koenig 2004

Koenig, J.: „Seven Open Source Business Strategies for Competitive Advantage“. In: *IT Manager's Journal* 2004, S. 1–6.

Kort/Zaccour 2011

Kort, P.M./Zaccour, G.: „When Should a Firm Open its Source Code: A Strategic Analysis“. In: *Production and Operations Management*, 20, 2011, S. 877–888.

Laguna de la Vera/Ramge 2021

Laguna de la Vera, R./Ramge, T.: *Sprunginnovation; Wie wir mit Wissenschaft und Technik die Welt wieder in Balance bekommen*, Berlin: Econ 2021.

Lerner/Tirole 2002

Lerner, J./Tirole, J.: „Some Simple Economics of Open Source“. In: *The Journal of Industrial Economics*, 50, 2002, S. 197–234.

Möller et al. 2021

Möller, F./Haße, H./Azkan, C./van der Valk, H./Otto, B.: *Design of Goal-Oriented Artifacts from Morphological Taxonomies: Progression from Descriptive to Prescriptive Design Knowledge Proceedings of the 16th International Conference on Wirtschaftsinformatik* 2021.

Mozilla/Open Tech Strategies 2018

The Mozilla Foundation/Open Tech Strategies: *Open Source Archetypes: A Framework For Purposeful Open Source*, 2018. URL: https://blog.mozilla.org/wp-content/uploads/2018/05/MZOTS_OS_Archetypes_report_ext_scr.pdf [Stand: 02.08.2021].

Müller et al. 2019

Müller, M./Vorraber, W./Slany, W.: „Open Principles in New Business Models for Information Systems“. In: *Journal of Open Innovation: Technology, Market, and Complexity*, 5: 1, 2019, S. 6.

Munga et al. 2009

Munga, N./Fogwill, T./Williams, Q.: *The adoption of open source software in business models Proceedings of the 2009 Annual Conference of the South African Institute of Computer Scientists and Information Technologists*, Vanderbijlpark: 2009.

Myers/Newmann 2006

Myers, M.D./Newman, M.: „The qualitative interview in IS research: Examining the craft“. In: *Information and Organization*, 17: 1, 2007, S. 2–26.

Nagle et al. 2020

Nagle, F./Wheeler, D./Lifshitz-Assaf, H./Ham, H./Hoffman, J.: *Report on the 2020 FOSS Contributor Survey*. 2020. URL: www.linuxfoundation.org/wp-content/uploads/2020/FOSSContributorSurveyReport_121020.pdf [Stand: 15.02.2022].

Nickerson et al. 2013

Nickerson, R.C./Varshney, U./Muntermann, J.: „A Method for Taxonomy Development and its Application in Information Systems“. In: *European Journal of Information Systems*, 22, 2013, S. 336–359.

OpenChain 2022

OpenChain: *Welcome to Our Reference Library. Learn How OpenChain Helps Others*. URL: www.openchainproject.org/resources [Stand: 10.01.2022].

Open Manufacturing Platform 2020

Open Manufacturing Platform: *Accelerating manufacturing innovation at scale through the power of open collaboration*, 2020. URL: <https://open-manufacturing.org/> [Stand: 10.01.2022].

Open Robotics 2021

Open Robotics: *Customer Stories*, 2021. URL: www.openrobotics.org/customer-stories [Stand: 10.01.2022].

Open Source 2022

Open Source: What is Open Source? 2022. URL: <https://opensource.com/resources/what-open-source> [Stand 09.03.2022].

Open Source Initiative 2021

Open Source Initiative: *The Open Source Definition*, 2021. URL: <https://opensource.org/osd> [Stand: 31.08.2021].

OSADL 2021a

OSADL: *OSADL Projects: Realtime Linux*, 2021. URL: www.osadl.org/Realtime-Linux.projects-realtime-linux.0.html [Stand: 10.01.2022].

OSADL 2021b

OSADL: *The OSADL OPC UA PubSub over TSN Project*, 2021. URL: www.osadl.org/OPC-UA-PubSub-over-TSN.opcua-tsn.0.html [Stand: 10.01.2022].

Osterwalder 2004

Osterwalder, A.: *The business model ontology; a proposition in a design science approach*. L'Université de Lausanne, 2004.

Peppers et al. 2007

Peppers, K./Tuunanen, T./Rothenberger, M.A./Chatterjee, S.: „A Design Science Research Methodology for Information Systems Research“. In: *Journal of Management Information Systems*, 24: 3, 2007, S. 45–77.

Plattform Industrie 4.0 2015

Plattform Industrie 4.0: *Umsetzungsstrategie Industrie 4.0. Ergebnisbericht der Plattform Industrie 4.0*, 2015. URL: www.bitkom.org/sites/default/files/file/import/150410-Umsetzungsstrategie-0.pdf [Stand: 15.02.2022].

Pimcore 2022

Pimcore: *Why Open Source?* 2022. URL: <https://pimcore.com/en/why-open-source> [Stand: 10.01.2022].

Proffitt et al. 2020

Proffitt, B./McCance, S./Behrenshausen, B./ Dickerson, P./ Wade, K.: *The Open Source Way 2.0. Contributors*, 2020. URL: www.theopensourceway.org/the_open_source_way-guidebook-2.0.pdf [Stand: 08.03.2022]

Rajala et al. 2006

Rajala, R./Nissilä, J./Westerlund, M.: *Determinants of OSS revenue model choices European Conference on Information Systems 2006 Proceedings*, 2006.

Raymond 2001

Raymond, E.S.: *The cathedral and the bazaar; Musings on Linux and open source by an accidental revolutionary*, Beijing: O'Reilly 2001.

Red Hat 2020

Red Hat: *Understanding open source governance models*, 2020. URL: www.redhat.com/en/blog/understanding-open-source-governance-models [Stand: 10.01.2022].

Red Hat 2021

Red Hat: *The State of Enterprise Open Source*, 2021. URL: www.redhat.com/de/enterprise-open-source-report/2021 [Stand: 31.08.2021].

Rhyn/Blohm 2017

Rhyn, M./Blohm, I.: „Combining Collective and Artificial Intelligence: Towards a Design Theory for Decision Support in Crowdsourcing“. In: *European Conference on Information Systems (ECIS)*, 2017, S. 2656-2666.

Riehle 2019

Riehle, D.: „The Innovations of Open Source“. In: *Computer*, 52: 4, 2019, S. 59-63.

ROS 2021

ROS: *ROS – Robot Operating System*. 2021. URL: www.ros.org/ [Stand: 10.01.2022].

Schreieck 2016

Schreieck, M./Wiesche, M./Krcmar, H.: *Design and Governance of Platform Ecosystems – Key Concepts and Issues for Future Research Proceedings of the 24th European Conference on Information Systems, Istanbul: Turkey: 2016*.

Shah 2006

Shah, S.K.: „Motivation, Governance, and the Viability of Hybrid Forms in Open Source Software Development“. In: *Management Science*, 52, 2006, S. 1000-1014.

Shah/Nagle 2020

Shah, S.K./Nagle, F.: „Why Do User Communities Matter for Strategy?“ In: *Strategic Management Review*, 1, 2020, S. 305-353.

Shanker 2012

Shanker, A.: „A Customer Value Creation Framework for Businesses That Generate Revenue with Open Source Software“. In: *Technology Innovation Management Review*, 2, 2012, S. 18-22.

Siepmann/Graef 2016

Siepmann, D./Graef, N.: „Industrie 4.0 – Grundlagen und Gesamtzusammenhang.“ In: Roth, A. (Hrsg.) *Einführung und Umsetzung von Industrie 4.0: Grundlagen, Vorgehensmodell und Use Cases aus der Praxis*, Berlin/Heidelberg: Springer 2016, S. 17-82.

Soder 2014

Soder, J.: „Use Case Production: Von CIM über Lean Production zu Industrie 4.0“. In: Bauernhansl, T./Hompe, M. ten/Vogel-Heuser, B. (Hrsg.): *Industrie 4.0 in Produktion, Automatisierung und Logistik: Anwendung · Technologien · Migration*, Wiesbaden: Springer Fachmedien 2014, S. 85-102.

Sommer 2021

Sommer, S.: *Open Source: Wer Code teilt, wird Zukunft ernten*, 2021. URL: www.handelsblatt.com/technik/digitale-revolution/ada-magazin-open-source-wer-code-teilt-wird-zukunft-ernten/26946866.html?ticket=ST-5564885-EeR3FdUuQR0dbPoSUotr-cas01.example.org [Stand: 07.12.2021].

Sonatype 2016

Sonatype: *State of the Supply Chain*, 2016. URL: www.sonatype.com/hubfs/SSC/Software_Supply_Chain_Inforgraphic.pdf?t=1468857601884 [Stand: 07.12.2021].

Spaeth et al. 2015

Spaeth, S./Krogh, G./He, F.: „Research Note – Perceived Firm Attributes and Intrinsic Motivation in Sponsored Open Source Software Projects“. In: *Information Systems Research*, 26, 2014, S. 224-237.

Spijkerman/Jansen 2018

Spijkerman, Z./Jansen, S.: „The open source software business model blueprint: A comparative analysis of 10 open source companies SiBW“, In: *SiBW Software-intensive Business: Start-ups, Ecosystems and Platforms*, 2018, S. 128-143. URL: <http://ceur-ws.org/Vol-2305/> [Stand: 15.02.2022].

Stallmann 1983

Stallman, R.: *Free Unix!* (Initial Announcement vom 27.09.1983). URL: www.gnu.org/gnu/initial-announcement [Stand: 07.12.2021].

Synopsis 2020

Synopsis: *Open Source Security and Risk Analysis Report, 2020*. URL: <https://tpsc.com/wp3/wp-content/uploads/2020/10/2020-ossra-report.pdf> [Stand: 10.01.2022].

The Linux Foundation 2018

The Linux Foundation: *Corporate Open Source Programs are on the Rise as Shared Software Development Becomes Mainstream for Businesses*, 2018. URL: www.linuxfoundation.org/press-release/corporate-open-source-programs-are-on-the-rise-as-shared-software-development-becomes-mainstream-for-businesses/ [Stand: 07.12.2021].

ten Hompel/Henke 2022

ten Hompel, M./Henke, M.: „Silicon Economy: Technologiebasis“. In: ten Hompel, M./Henke, M./Otto, B. (Hrsg.): *Silicon Economy. Wie digitale Plattformen industrielle Wertschöpfungsnetzwerke global verändern*, Berlin/Heidelberg: Springer 2022.

ten Hompel et al. 2022

ten Hompel, M./Henke, M./Otto, B. (Hrsg.): *Silicon Economy; Wie digitale Plattformen industrielle Wertschöpfungsnetzwerke global verändern*, Berlin/Heidelberg: Springer 2022.

United Nations 2015

United Nations: *Transforming our world: the 2030 Agenda for Sustainable Development*, 2015. URL: <https://sdgs.un.org/2030agenda> [Stand: 07.12.2021].

Välimäki/Oksanen 2005

Välimäki, M./Oksanen, V.: „The impact of free and open source licensing on operating system software markets“. In: *Telematics and Informatics*, 22, 2005, S. 97-110.

VDI et al. 2015

VDI/VDE/ZVEI (Hrsg.): *Referenzarchitekturmodell Industrie 4.0 (RAMI4.0). Statusreport*, 2015. URL: www.zvei.org/fileadmin/user_upload/Themen/Industrie_4.0/Das_Referenzarchitekturmodell_RAMI_4.0_und_die_Industrie_4.0-Komponente/pdf/Statusreport-Referenzmodelle-2015-v10.pdf [Stand: 09.03.2022].

Vom Brocke et al. 2009

Vom Brocke, J./Simons, A./Niehaves, B./Reimer, K./Plattfaut, R./Cleven, A.: *Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature Search Process Proceedings of the 17th European Conference on Information Systems*, Verona, Italy: AIS 2009.

Vom Brocke et al. 2015

Vom Brocke, J./Simons, A./Riemer, K./Niehaves, B./Plattfaut, R./Cleven, A.: „Standing on the Shoulders of Giants: Challenges and Recommendations of Literature Search in Information Systems Research“. In: *Communications of the Association for Information Systems*, 37, 2015, S. 205-224.

Weking et al. 2018

Weking, J./Stöcker, M./Kowalkiewicz, M./Böhm, M./Krcmar, H.: *Archetypes for Industry 4.0 Business Model Innovations*. 2018

West 2003

West, J.: „How open is open enough?“ In: *Research Policy*, 32, 2003, S. 1259-1285.

West/Bogers 2014

West, J./Bogers, M.: „Leveraging External Sources of Innovation: A Review of Research on Open Innovation“. In: *Journal of Product Innovation Management*, 31, 2014, S. 814-831.

West/Gallagher 2006

West, J./Gallagher, S.: „Challenges of open innovation: the paradox of firm investment in open-source software“. In: *R&D Management*, 36, 2006, S. 319-331.

West/O'mahony 2008

West, J./O'mahony, S.: „The Role of Participation Architecture in Growing Sponsored Open Source Communities“. In: *Industry and Innovation*, 15, 2008, S. 145-168.

Wiegmann et al. 2017

Wiegmann, P.M./Vries, H.J. de/Blind, K.: „Multi-mode standardisation: A critical review and a research agenda“. In: *Research Policy*, 46, 2017, S. 1370-1386.

Autorinnen und Autoren

Prof. Dr. Dr. h. c. Michael ten Hompel ist geschäftsführender Leiter des Fraunhofer-Instituts für Materialfluss und Logistik (IML) sowie Ordinarius des Lehrstuhls für Förder- und Lagerwesen (FLW) der TU Dortmund. Er gilt als einer der Väter des Internet der Dinge und wurde 2012 in die Logistics Hall of Fame aufgenommen. Ten Hompel ist Mitglied von acatech – Deutsche Akademie der Technikwissenschaften und Mitglied im Forschungsbeirat der Plattform Industrie 4.0.

Dr.-Ing. Michael Schmidt ist Chief Scientist am Fraunhofer IML. Im Rahmen seiner Tätigkeit ist er verantwortlich für forschungsstrategische Fragestellungen und strategische Initiativen des Instituts. Als Leiter Strategieentwicklung der vom Bundesministerium für Digitales und Verkehr BMDV geförderten Initiative Silicon Economy befasst er sich intensiv damit, wie Open Source Software die B2B-Plattformwirtschaft verändern wird.

Carina Culotta, M. Sc. ist wissenschaftliche Mitarbeiterin am Fraunhofer IML und befasst sich im Rahmen ihrer Forschung mit digitalen Plattformen, der Blockchain-Technologie, Methoden der Geschäftsmodellentwicklung und der wirtschaftlichen Bedeutung von Open Source Software.

Dr.-Ing. Marius Brehler ist Senior Scientist am Fraunhofer IML und an der Entwicklung verschiedener Open Source-Projekte beteiligt. Im Rahmen seiner Forschung befasst er sich unter anderem mit Maschinellem Lernen. Hierbei steht insbesondere die Open Source-Entwicklung von Compilern fürs Maschinelles Lernen im Vordergrund.

Estelle Duparc, M. Sc. ist wissenschaftliche Mitarbeiterin am Lehrstuhl für Industrielles Informationsmanagement der Technischen Universität Dortmund in der Forschungsgruppe Silicon Economy Logistics Ecosystems. Im Forschungsschwerpunkt Digitalisierung in Produktion und Logistik untersucht sie den strategischen Einsatz von Open Source, um digitale Geschäftsmodelle und Plattformen im industriellen Kontext zu entwickeln.

Jens Leveling, M.Sc. ist wissenschaftlicher Mitarbeiter am Fraunhofer IML und bearbeitet Forschungs- und Industrieprojekte mit Fokus auf Machine Learning und Internet der Dinge. Er untersucht neue IT-Entwicklungen und -Technologien und bewertet deren Potentiale für Anwendungen in der Logistik. Im Fokus dabei steht der Einsatz von Open Source-Entwicklungen. Seit Mai 2020 ist er Lead Software Architekt der Silicon Economy.

Dipl.-Inform. Andreas Nettsträter ist verantwortlich für strategische Initiativen und europäische Kooperationen am Fraunhofer IML und Geschäftsführer der Open Logistics Foundation. Sein Fokus liegt auf innovativen IKT-Lösungen für Logistik und Produktion, wie Internet der Dinge, maschinelles Lernen und autonome Systeme. Er hat einen Abschluss in Informatik und Maschinenbau und ist stellvertretender Vorsitzender für innovative IKT-Systeme in ALICE, der europäischen Technologieplattform für Logistik.

Hendrik van der Valk, M. Sc. ist wissenschaftlicher Mitarbeiter am Lehrstuhl für Industrielles Informationsmanagement an der Technischen Universität Dortmund in der Forschungsgruppe Silicon Economy Logistics Ecosystems. Im Forschungsschwerpunkt Digitalisierung in Produktion und Logistik erarbeitet er Referenzmodelle für digitale Zwillinge sowie Datenplattformen für den organisationsübergreifenden Datenaustausch.

Mitglieder des Forschungsbeirats

Vertreterinnen und Vertreter der Wissenschaft

Prof. Reiner Anderl, TU Darmstadt
Prof. Thomas Bauernhansl, Universität Stuttgart / Fraunhofer IPA
Prof. Manfred Broy, TU München
Prof. Angelika Bullinger-Hoffmann, TU Chemnitz
Prof. Claudia Eckert, TU München / Fraunhofer AISEC
Prof. Ulrich Epple, RWTH Aachen
Prof. Alexander Fay, Helmut-Schmidt-Universität Hamburg
Prof. Jürgen Gausemeier, Universität Paderborn
Prof. Hartmut Hirsch-Kreinsen, TU Dortmund
Prof. Gerrit Hornung, Universität Kassel
Prof. Gisela Lanza, KIT – Karlsruher Institut für Technologie
Prof. Peter Liggesmeyer, TU Kaiserslautern / Fraunhofer IESE
Prof. Wolfgang Nebel, Universität Oldenburg / OFFIS
Prof. Sabine Pfeiffer, Friedrich-Alexander-Universität Erlangen-Nürnberg
Prof. Frank Piller, RWTH Aachen
Prof. Thomas Schildhauer, Alexander von Humboldt Institut für Internet und Gesellschaft/Institute of Electronic Business
Prof. Rainer Stark, TU Berlin
Prof. Michael ten Hompel, TU Dortmund / Fraunhofer IML
Prof. Wolfgang Wahlster, DFKI – Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Vertreterinnen und Vertreter der Industrie

Klaus Bauer, TRUMPF Werkzeugmaschinen GmbH & Co. KG
Dr. Jan-Henning Fabian, ABB AG
Dr. Ursula Frank, Beckhoff Automation GmbH & Co. KG
Dr. Christina Franke, Robert Bosch GmbH
Dietmar Goericke, VDMA – Verband Deutscher Maschinen und Anlagenbau e.V.
Prof. Torsten Kröger, Intrinsic
Dr. Uwe Kubach, SAP SE
Prof. Peter Post, Festo AG & Co. KG
Dr. Harald Schöning, Software AG
Dr. Georg von Wichert, Siemens AG
Dr. André Walter, Airbus Operations GmbH

