

Alexander Ziegler

Das Tech-Unternehmen

Zum Fundament eines neuen Unternehmenstypus

Zuerst online veröffentlicht am 15. Juli 2021

Arbeitssoziologische Forschungsergebnisse 01/2021

Research Findings in Sociology of Work 01/2021

Herausgegeben vom Institut für Sozialwissenschaftliche Forschung e. V. – ISF München

Die Arbeitssoziologischen Forschungsergebnisse sind eine in loser Folge erscheinende Reihe des Instituts für Sozialwissenschaftliche Forschung – ISF München. Sie dient dazu, aktuelle Forschungsergebnisse aus dem Institut der Öffentlichkeit vorzustellen. Die Beiträge werden in einem Repositorium auf der Website www.isf-muenchen.de dauerhaft vorgehalten. Rückmeldungen und Diskussion mit den Autorinnen und Autoren sind ausdrücklich erwünscht und können an die Redaktion adressiert werden.

Herausgeber:

Institut für Sozialwissenschaftliche Forschung e. V. – ISF München

Jakob-Klar-Str. 9, 80796 München, Tel. +49 (0)89 272921-78

www.isf-muenchen.de

Redaktion: Frank Seiß, Lektorat am ISF München, frank.seiss@isf-muenchen.de

© 2021 ISF München

Zitierempfehlung:

Ziegler, Alexander (2021): Das Tech-Unternehmen. Zum Fundament eines neuen Unternehmenstypus. Arbeitssoziologische Forschungsergebnisse 01/2021. München: ISF München.

<https://www.isf-muenchen.de/pdf/arbeitssoziologische-forschungsergebnisse/2021-ziegler-tech-unternehmen-fundament-unternehmenstypus.pdf>

Alexander Ziegler

Das Tech-Unternehmen

Zum Fundament eines neuen Unternehmenstypus¹

<https://doi.org/10.36194/ArbSozForschung-2021-001>

Zusammenfassung: Tech-Unternehmen wie Amazon, Google oder Stripe gelten als Schlüsselakteure der digitalen Transformation. Während ihre Analyse in den letzten Jahren in der soziologischen Digitalisierungsforschung zunehmend an Bedeutung gewonnen hat, konnte eine Definition von Tech-Unternehmen bisher noch nicht vorgelegt werden. Vor diesem Hintergrund zielt der Aufsatz darauf, die Grundlage für die Ausarbeitung eines soziologischen Begriffs des Tech-Unternehmens als neuen Unternehmenstypus zu schaffen. In Anlehnung an den „Münchener Betriebsansatz“ lautet die Grundthese, dass das Tech-Unternehmen als eine historisch neuartige Strategie der (Einzel-)Kapitalverwertung zu begreifen ist. Diese Strategie zeichnet sich wesentlich dadurch aus, dass sie um die kontinuierliche Entwicklung, den Betrieb und die Monetarisierung von Internetanwendungen zentriert. Darauf aufbauend rückt die Frage, welche spezifischen Fähigkeiten in den Tech-Unternehmen zur Realisation dieser Strategie entwickelt wurden, ins Zentrum des Aufsatzes. In einer historischen Rekonstruktion der Herausbildung der ersten Tech-Unternehmen wird gezeigt, dass mit der Entwicklung IT-basierter Maschinensysteme aus verteilten Serverclustern und der Modularisierung der Softwarearchitekturen ein neues technologisch-organisatorisches Fundament geschaffen wurde, auf dem Internetanwendungen ins Zentrum von Verwertungsstrategien rücken können.

Schlüsselwörter: Tech-Unternehmen, digitale Transformation, Unternehmensstrategie, Tech-Arbeiter, neue Maschinensysteme, Cloud-Computing

Dr. Alexander Ziegler ist Wissenschaftler am ISF München e.V.

Kontakt: alexander.ziegler@isf-muenchen.de

¹ Für Impulse und kritische Anregungen zu den in diesem Text vorgestellten Überlegungen danke ich den Expert:innen Wolfgang Fritz Haug, Anja Hendel, Stefan Henze, Johannes Katzan, Martin Schmitt, Patrick Stähler und Sebastian Wohlrapp sowie meinen ISF-Kolleg:innen Andreas Boes, Eckhard Heidling, Konstantin Klur, Thomas Lühr, Sarah Nies, Sabine Pfeiffer, Kurt Rachlitz, Dieter Sauer, Frank Seiß und Elisabeth Vogl.

The Tech Company

On the foundations of a new type of corporation

Abstract: Tech companies such as Amazon, Google or Stripe are seen as key actors of digital transformation. Whereas the analysis of their strategies is gaining traction in sociological research on digitalization, a definition of tech companies has not yet been presented. Against this background, the article aims to establish the basis for the elaboration of a sociological concept of the tech company as a new type of corporation. Following in the footsteps of the “Münchner Betriebsansatz”, the main thesis is that the tech company is to be seen as a historically new strategy for the valorization of single capital. This strategy centers around the continuous development, operation, and monetization of internet applications. Building on this, the paper focuses on the specific capabilities that have been developed among the tech companies to enable this strategy. In a historical reconstruction of the emergence of the first tech companies it is shown that the development of IT-based machine systems consisting of distributed server clusters and modularized software architectures has laid a new technological and organizational foundation for internet applications to move to the center of valorization strategies.

Keywords: tech company, digital transformation, corporate strategy, tech worker, new machine systems, cloud computing

Dr. Alexander Ziegler is researcher at the ISF München e.V.

Contact: alexander.ziegler@isf-muenchen.de

A lot of people still don't realize it,
but we really are a technology company
Andy Jassy, designerter CEO von Amazon

1. Einleitung

Tech-Unternehmen gelten als die prägenden Akteure der digitalen Transformation von Wirtschaft und Gesellschaft. Mystischen Wesen gleich kreuzen sich in ihnen die Faszination angesichts der Fähigkeiten ihrer Produkte und der Rasanz ihres Aufstiegs mit der Furcht vor ihrer wachsenden gesellschaftlichen Macht. Das schillernde Präfix „Tech“ stand ursprünglich für „Technology“ und diente als Bezeichnung für Unternehmen, die Computerhardware- und Softwaretechnologie bereitstellten. Der im Web 2.0 gegründete Blog TechCrunch mit seiner vielgelesenen Berichterstattung zur Start-up-Szene des Silicon Valley machte „Tech“ zu einem populären Lifestyle-Ausdruck, der schnell auf die Internetwirtschaft schlechthin ausgedehnt wurde. Neben Konzernen wie Google, Amazon, Alibaba oder Zalando werden heute auch aufstrebende Unternehmen wie der Entwickler von E-Commerce-Technologie Shopify, der Kommunikationsdienst Twilio, der Zahlungsdienst Stripe oder ByteDance mit seinem sozialen Netzwerk TikTok als Tech-Unternehmen bezeichnet.

Als kleine Start-ups zuerst an der Westküste der USA entstanden, ist es Tech-Unternehmen in sehr kurzer Zeit gelungen, das mit hohen staatlichen Finanzspritzen im US-amerikanischen Militärkomplex entstandene Internet wirtschaftlich zu erschließen und in die erste unmittelbare „weltgesellschaftliche Handlungsebene“ (Boes et al. 2015, 83) der Menschheit zu verwandeln. Ausgestattet mit Risikokapital haben sie, angefangen beim Einzelhandel, dem Werbegeschäft, der Medien-, Film-, Musik- und IT-Industrie, immer weitere Branchen mit internetbasierten Geschäftsmodellen umgestaltet.² Während gegenwärtig auf der einen Seite gerade die „Tech-Giganten“ aufgrund ihrer in der Corona-Pandemie rasant weiter ausgebauten Dominanz über bestimmte Märkte (Galloway 2018; Staab 2019), ihres Strebens nach Kommodifizierung der Privatsphäre (Zuboff 2018) oder ihres Einflusses auf die öffentliche Meinungsbildung und das politische Geschehen (Highfield 2016) immer mehr in den Fokus gesellschaftlicher Debatten rücken, orientieren sich auf der anderen Seite immer neue Start-ups mit ihren Geschäftsideen, aber auch etablierte Unternehmen wie General Electric, Volkswagen oder die Allianz bei der Neuausrichtung ihrer Geschäftsmodelle in der digitalen Transformation an diesen Vorbildern (Ries 2017; Kersten 2018). Im Internet der Dinge rücken Tech-Unternehmen nun in industrielle Wertschöpfungsprozesse vor, versuchen Produkte und Geschäftsmodelle zu transformieren und den etablierten Marktgrößen die Spitzenpositionen streitig zu machen (Ziegler 2020). Vor diesem Hintergrund wird daher auch in den Unternehmen, Gewerkschaften, Politik und Wissenschaft hierzulande immer intensiver darüber diskutiert, was genau diese Tech-Unternehmen auszeichnet, worin ihre spezifischen Fähigkeiten bestehen und wie diese in der Praxis umgesetzt werden.

² Die für solch disruptive Vorhaben günstigen politökonomischen Rahmenbedingungen dieses Innovationsystems mit seinem Zentrum Silicon Valley wurden in mehreren Jahrzehnten etabliert. Siehe dazu bereits Kenney (2000) und Lüthje (2001) sowie aktuell O'Mara (2019).

Dieser Aufsatz ist in der Überzeugung geschrieben, dass die soziologische Forschung sowohl konzeptionell als auch empirisch einen wichtigen Beitrag zur Beantwortung dieser Fragen leisten kann. Er zielt darauf, eine Grundlage für die Ausarbeitung eines soziologischen Begriffs des Tech-Unternehmens als eines neuen, die digitale Transformation prägenden Unternehmenstypus zu schaffen und eine Diskussion zur Bestimmung seiner Dimensionen und Spezifika anzustoßen.

Im ersten Schritt wird dafür in Auseinandersetzung mit den vorliegenden soziologischen Forschungsergebnissen eine *Basisdefinition* des Tech-Unternehmens vorgelegt. In Anlehnung an den „Münchener Betriebsansatz“ lautet die Grundthese, dass das Tech-Unternehmen als eine historisch neuartige Strategie der (Einzel-)Kapitalverwertung zu begreifen ist. Diese Strategie zeichnet sich wesentlich dadurch aus, dass sie um die kontinuierliche Entwicklung, den Betrieb und die Monetarisierung von Internetanwendungen zentriert. Darauf aufbauend rückt im zweiten Schritt die Frage, welche spezifischen Fähigkeiten zur Realisation dieser Strategie in den Tech-Unternehmen entwickelt wurden, ins Zentrum des Aufsatzes. In einer historischen Rekonstruktion der Herausbildung der ersten Tech-Unternehmen wird gezeigt, dass mit der Entwicklung IT-basierter Maschinensysteme aus verteilten Serverclustern und der Modularisierung der Softwarearchitekturen ein neues technologisch-organisatorisches Fundament geschaffen wurde, auf dem Internetanwendungen ins Zentrum von Verwertungsstrategien rücken können. Abschließend werden im dritten Schritt die zentralen Ergebnisse zusammengefasst und weitergehende Forschungsbedarfe skizziert.

Die Grundlage für diesen Aufsatz bildet eine umfassende Analyse empirischen Materials, welche kontinuierlich erweitert und fortgesetzt wird. Sie gründet auf der einen Seite auf den Auswertungen von zwei Forschungsaufenthalten in den Jahren 2015 und 2017 im Silicon Valley im Kontext des vom Bundesministerium für Bildung und Forschung geförderten Verbundvorhabens „digit-DL – digitale Dienstleistung in modernen Wertschöpfungssystemen – neue Produktivitätspotenziale nachhaltig gestalten“, gemeinsam mit Kolleg:innen am ISF München. In diesem konnten zahlreiche qualitative Beschäftigten- und Experten:inneninterviews in Start-ups, Tech-Unternehmen sowie den im Silicon Valley ansässigen Forschungseinheiten etablierter Unternehmen (z.B. Automobilindustrie und IT-Industrie) geführt und durch Hintergrundgespräche mit verschiedenen Akteuren aus dem Innovationssystem (z.B. Risikokapitalfonds) flankiert werden.³ Auf der anderen Seite gründet sie auf der Auswertung eines umfangreichen Korpus von Sekundärliteratur und Primärquellen, wie Blogbeiträge, öffentlich zugängliche Interviews, Kommunikation in Foren (z.B. Github) und Podcasts, Oral-History-Dokumente und Fachvorträge zur Thematik. Das Ziel dieses langfristig angelegten Forschungsprozesses besteht darin, die Entwicklung des Tech-Unternehmens als neuen Unternehmenstypus im „Flusse der Bewegung“ (MEW 23, 28) zu rekonstruieren – von der Genese bis hinein in die Gegenwart – und die wesentlichen Elemente dieser neuen Strategie freizulegen.

³ Siehe dazu Boes et al. 2018.

2. Das Tech-Unternehmen. Ein neuer Unternehmenstypus

Die soziologische Forschung im deutschsprachigen Raum hat anders als z.B. die Managementforschung in den USA⁴ vergleichsweise spät begonnen, sich mit dem Phänomen der Tech-Unternehmen zu befassen. Erste Analysen wurden Mitte der 2010er Jahre vorgelegt und konzentrierten sich seither überwiegend auf die US-amerikanischen GAFA.⁵ Sie konnten viele Aspekte der Strategien dieser Konzerne herausarbeiten. Dazu zählen „die radikale Infragestellung der Funktionsweise etablierter Märkte“ in „Strategie[n] der Disruption“ (Nachtwey/Staab 2015, 60f), die Etablierung einer „Plattformlogik“ (Kirchner/Beyer 2016, 337) zur Organisation von Märkten als „integrierte soziotechnische Ökosysteme“ (Dolata 2015, 511) sowie zur Erzielung von „Netzwerkeffekten“ (Nachtwey/Staab 2015, 71), die Verwertung „der Daten der Nutzer“ (Staab 2016, 9) und die Entwicklung „informationsbasierter Geschäftsmodelle“ (Boes et al. 2019, 126), die Etablierung neuer Formen der „Wertrealisierung“ (Pfeiffer 2019, 393ff), die Verwendung „agiler Konzepte der Organisation von Arbeit“ (Langes/Vogl 2019, 158), aber auch der Einsatz „digitaler Kontrolltechnologien“ (Nachtwey/Staab 2016, 29) zur Steuerung von Arbeitsprozessen, konsequente Unternehmensübernahmen zur Verteidigung und zum Ausbau gefestigter Marktpositionen (Dolata 2015, 512ff) und neuartige Steuervermeidungsstrategien (Staab 2019, 267).

In der Zusammenschau entsteht aus diesen Arbeiten eine umfassende Übersicht wichtiger Aspekte der Strategien von Tech-Unternehmen. Die Frage, was die verschiedenen Tech-Unternehmen miteinander verbindet und worin sie sich wesentlich von anderen Unternehmen unterscheiden, wurde jedoch in der soziologischen Forschung kaum explizit gestellt.⁶ In den bisherigen Auseinandersetzungen mit ihren Strategien standen stattdessen häufiger übergeordnete Fragestellungen im Zentrum, etwa nach einer neuen Form kapitalistischen Wirtschaftens (Boes et al. 2019, 122; Nachtwey/Staab 2020, 285), die in den Aktivitäten einzelner Vorreiter- und Leitunternehmen der Tech-Branche sichtbar wird. Nur selten haben Analysen von Tech-Unternehmen bisher dezidiert die Unternehmensperspektive eingenommen (z.B. Dolata 2015) und nur cursorisch wurde der Blick auf ihre innere Betriebsweise gerichtet.

In Auseinandersetzung mit den vorliegenden soziologischen Arbeiten zu Tech-Unternehmen fällt darüber hinaus auf, dass ihre Strategien überwiegend *in geronnener Form* als *Strategiemuster* analysiert wurden – zu einem Zeitpunkt, als Tech-Unternehmen wie Apple, Google oder Amazon bereits in die Riege der wertvollsten Unternehmen der Welt aufgestiegen waren. Dabei gerät in Vergessenheit, dass in der Anfangszeit des World Wide Web Muster für die

⁴ Für einen ersten Überblick zur Forschung über Tech-Unternehmen im strategischen Management siehe Ziegler (2020, 87ff).

⁵ Das Kürzel GAFA steht für Google, Apple, Facebook und Amazon.

⁶ Im Silicon Valley hingegen wurde, wenn auch eher selten dokumentiert, über diese Frage diskutiert. Als einer der ersten explizit formulierte sie der Entwickler Alex Payne, der eine Schlüsselrolle beim Aufbau der IT-Infrastruktur von Twitter gespielt hatte. In seinem Blog definiert Payne (2012) ein Unternehmen als Tech-Unternehmen, wenn sein Produkt „consists of applied scientific knowledge that solves concrete problems and enables other endeavors“. Einer solchen ahistorischen Definition folgend könnten z.B. auch Maschinenbauunternehmen als Tech-Unternehmen bezeichnet werden. Eine einheitliche Definition hat sich bislang auch dort nicht herauskristallisiert. Zur weiteren Diskussion um die Bestimmung von Tech-Unternehmen im Silicon Valley siehe auch: Dash 2016; Guzzetta 2016; Heath 2017.

Gestaltung von Verwertungsstrategien allenfalls in Ansätzen existierten.⁷ Keines dieser Start-ups konnte vorab exakt bestimmen, ob und wie genau mit dem Web Geld verdient werden kann (McCullough 2018, 35; O'Mara 2019, 302ff). Außerdem lässt sich durch eine solche Ex-post-Betrachtungsweise kaum erklären, warum sich z.B. Google mit seiner Anwendung zur Websuche gegenüber konkurrierenden Anwendungen von Altavista, LookSmart, Inktomi oder Yahoo durchgesetzt hat oder warum das soziale Netzwerk Facebook andere soziale Netzwerke wie Friendster oder MySpace verdrängte, obwohl diese zunächst über deutlich höhere Nutzerzahlen und somit First-Mover-Vorteile sowie ein höheres Kapitalaufkommen verfügten. Insgesamt entsteht in den Analysen bisweilen der Eindruck, dass jedes Unternehmen mit genügend finanzieller Ausstattung die Strategiemuster mehr oder weniger ohne Reibungsverluste umsetzen könnte. Nicht zuletzt der Blick auf die vielfältigen Bemühungen zur Adaption der Strategiemuster in etablierten Unternehmen und deren Initiativen etwa zum Aufbau digitaler Plattformen für die Industrie oder die Finanzbranche zeigen, dass dem keineswegs so ist (Ziegler 2020, 284ff; Butollo/de Paiva Lareiro 2020, 88ff).

Diese Beobachtungen verdeutlichen, dass die soziologische Forschung zu Tech-Unternehmen bis dato zwar viele Strategiemuster erfassen konnte, im Hinblick auf die eingangs angeführten Fragen nach ihrem Wesenskern, ihren spezifischen Fähigkeiten und ihrer praktischen Umsetzung jedoch noch weitgehend am Anfang steht. Ein Hauptgrund scheint mir darin zu liegen, dass eine entscheidende Dimensionen in der Analyse der Strategien dieser Unternehmen unterbelichtet geblieben ist: Die Strategiemuster der Tech-Unternehmen sind nicht am Reißbrett entstanden, sondern sie wurden in einem *sozialen Aushandlungsprozess* von den Akteur:innen entwickelt, umgesetzt und sie werden fortwährend weiterentwickelt. Ihre Konzeption, praktische Erprobung, Inspektion, Adaption, Konsolidierung und Skalierung ist Resultat des Zusammenwirkens der Arbeit einer Vielzahl von Softwareentwickler:innen und Architekt:innen, Systemingenieur:innen, UI-Designer:innen, Produktmanager:innen, Data Scientists, Techniker:innen, Gründer:innen, Risikokapitalist:innen, Manager:innen, Gig-Worker:innen, Zeitarbeiter:innen u.v.m. Wie die Silicon-Valley-Expert:innen Ben Tarnoff und Moira Weigel (2020, 3) einschärfen, entstehen Plattformen nicht aus sich selbst heraus, „platforms are made by people“. Die dafür erforderlichen Fähigkeiten wurden und werden in Technologien, Architekturen, Organisationskonzepten, dokumentierten Wissensbeständen, Praktiken und Vorgehensmodellen materialisiert, aber auch ins „Erfahrungswissen“ (Böhle/Milkau 1988) und den „Mindset“ (Blank et al. 2020) der Menschen eingeschrieben, im „Arbeits-handeln“ (Böhle 2018) erzeugt und in einem komplexen Innovationsnetzwerk aus Entwickler-Gemeinschaften, Universitäten, Wirtschaftshochschulen, Start-up-Inkubatoren, Risikokapitalfonds, Verlagen, Journalist:innen, Analyst:innen, Wirtschaftskanzleien, Gründer:innen und Manager:innen fortlaufend reflektiert und weiterentwickelt. Sie konstituieren die Praxis in den Tech-Unternehmen.

Vor diesem Hintergrund lautet die forschungsleitende Grundüberlegung, dass es über Strategiemuster wie Plattformkonzepte hinausgehend gerade diese dynamische Dimension ist, die Tech-Unternehmen als Repräsentanten eines neuen Unternehmenstypus im soziologischen Sinne auszeichnet. In Anlehnung an den Münchner Betriebsansatz (Altmann/Bechtel

⁷ Zum Beispiel in den Mustern und Praktiken der Plattformstrategien des „Wintelismus“ (Borras/Zysman 1997) in der PC-Industrie.

1971), der in den 1970er und 1980er Jahren am ISF München eine dynamische Perspektive auf die Analyse der *Strategiebildung* in Industriebetrieben entwickelte⁸, wird daher vorgeschlagen, das Tech-Unternehmen als eine historisch neue Form zu begreifen, „Unternehmen als Strategie“ der (Einzel-)Kapitalverwertung zu betreiben. In dieser Perspektive stellt sich die Frage, ob es eine Gemeinsamkeit zwischen Tech-Konzernen wie Google und Amazon und so heterogenen Tech-Unternehmen wie Snowflake mit seinen Anwendungen für Großdatenanalysen, Netflix⁹ mit seinem Streamingdienst, Atlassian mit seinen Lösungen für Projektmanagement, Klarna mit seinen Bezahldiensten oder Celonis mit seinen Lösungen zur Geschäftsprozessoptimierung besteht: Anders als bei allen bisher bekannten Unternehmen stehen bei all diesen Tech-Unternehmen *Internetanwendungen* im Zentrum der Verwertungsstrategien. Ausgehend davon lautet meine Basisdefinition des Tech-Unternehmens als neuen Unternehmenstypus daher:

Das Tech-Unternehmen zeichnet sich als Strategie dadurch aus, dass es um die kontinuierliche Entwicklung, den Betrieb und die Monetarisierung von Internetanwendungen zentriert.

Die Großdatenanalysen, der Streamingdienst, die Lösungen für Projektmanagement, die Bezahldienste oder die Lösungen zur Geschäftsprozessoptimierung gründen auf Internetanwendungen. Auch sog. Online-to-offline-Tech-Unternehmen wie die Mobilitätsplattform Uber, der Übernachtungsvermittler Airbnb oder der chinesische Lieferdienst Meituan sind letztlich Strategien, welche um Internetanwendungen zentrieren und ohne sie nicht umsetzbar sind. Dass die Strategien um Internetanwendungen *zentrieren*, schließt, wie diese Beispiele zeigen, nicht aus, dass Tech-Unternehmen ihre Wertschöpfungstiefe auf Felder außerhalb des Internets ausdehnen, wie nicht zuletzt Amazons weitverzweigtes Netzwerk an Logistikzentren verdeutlicht. Es bedeutet aber auch nicht, dass all die etablierten Unternehmen, die in den 1990er Jahren eine Webpräsenz aufbauten, dadurch zu Tech-Unternehmen geworden wären. Die *differentia specifica*, die ein Unternehmen mit „Offline-Fußabdruck“ dieser Basisdefinition folgend zum Tech-Unternehmen macht, besteht vielmehr darin, dass ihr „Offline-Fußabdruck“ von der Internetanwendung her gestaltet ist und nicht umgekehrt.

In dieser Hinsicht unterscheiden sich Tech-Unternehmen von Industrieunternehmen, die als Verwertungsstrategien ihren Angelpunkt in der maschinensystembasierten Produktion und dem Verkauf von Sachgütern haben (Ziegler 2020, 15). Darin liegt aber auch ein fundamentaler Unterschied zu den Unternehmen der klassischen Softwareindustrie wie Microsoft, Oracle oder SAP bis hin zu Netscape oder auch dem Geschäft mit Industriesoftware von Siemens. Statt wie diese Software zu entwickeln, dem Kunden paketiert zu übergeben, der sie selbst auf seinem Rechner installiert oder dessen IT-Abteilung sie ggf. mit Hilfe des Softwareunternehmens oder von Systemintegratoren an die spezifischen Unternehmensanforderungen anpasst und im eigenen oder dem Rechenzentrum eines IT-Dienstleisters betreibt, zeichnet Tech-Unternehmen aus, dass sie das Internet als Betriebssystem für ihre Anwendungen

⁸ Für einen zusammenfassenden Überblick über den Münchner Betriebsansatz siehe Nies/Sauer 2010 und Ziegler 2020, 63ff.

⁹ Hingegen besteht der springende Punkt dafür, ein Unternehmen als Tech-Unternehmen zu bezeichnen, für den Analysten und Risikokapitalisten Benedict Evans (2019) darin, dass es zur Verteidigung seiner Marktposition vor allem „tech questions“ beantworten müsse. Demnach sei Netflix nicht mehr als Tech-Unternehmen zu betrachten, da es zwar „tech as a crowbar“ nutze, aber in den aktuellen Diskussionen um Netflix „all of the questions that matter are TV industry questions“.

nutzen. Sie generieren „money *with* software rather than *from* software“ (O’Grady 2015, 27; Herv. d. Verf.). Wie im Folgenden gezeigt wird, verändern sich in dem Moment, wo Unternehmen ihre Kernanwendungen im Internet bereitstellen, sowohl die Anforderungen an ihre Software als auch der Umgang damit grundlegend.¹⁰ Die Herausbildung des Tech-Unternehmens als Unternehmenstypus kann so insgesamt auch als Antwort auf die Frage gelesen werden, wie sich Verwertungsstrategien im Internet realisieren lassen.

„Tech-Unternehmen als Strategie“ zu begreifen bedeutet ferner, dass „Tech-Unternehmen zu sein“ keine Eigenschaft ist, die einem Unternehmen angeboren ist. Es ist vielmehr eine Eigenschaft, die, um mit Marx zu sprechen, „wesentlich praktisch“ (MEW 3, 5) ist und erlernt werden kann. Dies zeigen nicht zuletzt die prominenten Beispiele von Apple, Microsoft oder Adobe, die sich von klassischen Computerhardware- bzw. Softwareunternehmen, die in ihrer Strategie um die Entwicklung und den Verkauf von PCs und Software zentrierten, in Tech-Unternehmen transformiert haben (O’Grady 2015),¹¹ während z.B. dem deutschen Softwarehaus SAP dieser Schritt noch nicht in Gänze gelungen zu sein scheint (Busse/Freiberger 2020). Im Anschluss an die vorgelegte Basisdefinition stellt sich daher die Frage, welche spezifischen Fähigkeiten an der US-amerikanischen Westküste entwickelt wurden, um Tech-Unternehmen als Verwertungsstrategie realisieren zu können. Diese Frage wird im Mittelpunkt der folgenden Ausführungen stehen. Dabei wird es nicht möglich sein, sämtliche Dimensionen dieser Fähigkeiten zu ergründen. Im Fokus liegt die historische Rekonstruktion der Herausbildung eines neuen technologisch-organisatorischen Fundaments, auf dessen Grundlage Internetanwendungen ins Zentrum von Verwertungsstrategien rücken konnten.

3. Auf der Suche nach dem Fundament von Tech-Unternehmen

In vielen Analysen zur Entwicklung der Start-ups und Unternehmen der Internetwirtschaft wird ein Aspekt dieser Verwertungsstrategien schlicht als gegeben und keiner besonderen Betrachtung für würdig erachtet: Tech-Unternehmen stellen permanent komplexe software- und datenbasierte Anwendungen für Privat- wie für Geschäftskunden im Internet bereit, die beständig weiterentwickelt werden. Diese Anwendungen sind nicht nur in der Lage, dynamisch Spitzenlasten abzufedern, sondern auch mit dem rasanten Geschäftswachstum Schritt zu halten. Dass darin ein wesentlicher Baustein der Verwertungsstrategien liegt, springt zu- meist erst dann ins Auge, wenn dieser ins Wanken gerät. Die Beispiele hierfür sind zahlreich und spannen sich von der Frühphase des World Wide Web bis in die heutige Zeit. Beim Ausbruch der Corona-Pandemie Anfang März 2020 führte etwa der Kollaps der IT-Infrastruktur

¹⁰ In ihrem Paper zur Softwareentwicklung bei Facebook beschreiben Feitelson et al. (2013, 9) einen Aspekt dieses Unterschieds folgendermaßen: „Traditional software products are finite by definition, with delimited scope and a predefined completion date. (...) Sites like Facebook will never be completed. The mindset is that the system will continue to be developed indefinitely.“

¹¹ Zwar generiert Apple noch immer den Großteil seiner Erlöse durch den Verkauf seiner Geräte. Diese Geräte wären allerdings weitgehend unbrauchbar, wenn sie nicht in Internetanwendungen eingebunden sind. Nachtwey/Staab (2016, 25) zufolge „dienen diese Ankerprodukte vor allem als Vehikel des Vertriebs und Verkaufs zahlreicher Applikationen und Services“. In den letzten Jahren konnte Apple zudem seine direkten Erlöse aus Internetanwendungen wie dem App Store oder Diensten wie Apple Music oder Apple TV kontinuierlich steigern. Die strategische Bedeutung, die es seinen Internetanwendungen beimisst, wird nicht zuletzt daran deutlich, dass es immer mehr Workloads z.B. aus der AWS Cloud auf eigene dynamisch-skalierbare IT-Infrastrukturen migriert und die Investitionen in den Aufbau eigener Datenzentren signifikant ausweitet (Schwan 2019).

des US-Fintechs Robinhood, das seinen Kunden über eine App gebührenfreien Aktienhandel ermöglicht, dazu, dass diese, während die Kurse abstürzten, nur noch dem Wertverlust ihrer Depots zuschauen konnten (Popper/Siegel Bernhard 2020).¹² Aufgrund der Unmittelbarkeit des Internets haben selbst kleinste Ausfallzeiten oder Datenverluste sofort signifikante funktionale wie finanzielle Auswirkungen und gefährden das Vertrauen der Kunden in die Lösungen. Doch wie ist es den – zumindest anfänglich noch kleinen – Start-ups gelungen, internetbasierte Anwendungen dynamisch zu skalieren, bei Hunderten Millionen Zugriffen gleichzeitig global hochverfügbar zu halten, permanent weiterzuentwickeln und kosteneffizient zu betreiben? Die Suche nach Antworten auf diese Frage führt zurück in die von Aufbruch, Euphorie und Krise geprägte Anfangszeit des World Wide Web.

3.1 „Warehouse-Scale-Computing“: Der Aufbau großtechnischer verteilter Systeme

Das Bestreben, nicht nur weitgehend statische Websites, sondern auch komplexe interaktive Anwendungen über das Internet bereitzustellen, entstand in der New Economy Mitte der 1990er Jahre. Die Umsetzung solcher Visionen stieß in dieser Zeit noch an die technologischen Grenzen, die durch die „schmalbandigen“ Datenübertragungsraten und Internetzugänge über Telefonmodems auferlegt wurden. Im Zuge der Spekulationseuphorie floss allerdings ein Großteil der enormen Kapitalvorschüsse neben den Dot.com-Start-ups in Telekommunikationskonzerne, die dieses Geld in den Aufbau von Breitbandzugängen investierten (McCullough 2018, 180). In ähnlicher Form, wie im Großbritannien der 1840er Jahre durch die von Karl Marx als „Eisenbahnschwindel“ (MEW 25, 424) bezeichnete Börsenblase der Ausbau eines weitverzweigten Eisenbahnnetzes beschleunigt worden war, trug die Dot.com-Blase dazu bei, dass eine neue Internet-Infrastruktur in den entwickelten Ländern gelegt wurde (Heim 2013, 242; Srnicek 2018, 25). Gegenüber den Internetzugängen mit Telefonmodems, die für die Zeit des Web 1.0 charakteristisch waren, stiegen die Datenübertragungsraten in den Breitbandnetzen sprunghaft. Im Zusammenspiel mit der Verbreitung von drahtlosen Netzwerken (WLANs) ebnete dies den Weg für neue Formen der Internetnutzung. Inmitten des Zusammenbruchs der New Economy entstand ein Möglichkeitsraum zur Umsetzung der Vision komplexer interaktiver Internetanwendungen.

Sowohl die verbliebenen Start-ups als auch neue Start-ups trieben die Erschließung dieser neuen Internetinfrastruktur voran. Hatten die Kosten für die Bereitstellung von Anwendungen über das Internet zur Hochzeit der Spekulationseuphorie für viele Start-ups eine zu vernachlässigende Rolle gespielt, da sie aus dem nach Anlage suchenden Überschuss an privatem Risikokapital bedient werden konnten (Staab 2019, 91ff), war dieser Zustrom an Risikokapital nun zunächst verbbt. Die Start-ups setzten sich daher intensiv damit auseinander, neue Strategien für eine effiziente Gestaltung ihrer IT-Infrastrukturen umzusetzen.

In dieser Konstellation entwickelten Pioniere wie Google oder Amazon eine bedeutende Basisinnovation: Statt steigende Anforderungen¹³ an Internetanwendungen durch die Anschaffung

¹² Selbst die Mitarbeiter „erfahrener“ Tech-Unternehmen wie Facebook kämpfen in der Corona-Pandemie angesichts neuer Rekorde in der Nutzung der sozialen Netzwerke des Unternehmens damit, ihre IT-Infrastrukturen am Laufen zu halten (Isaac/Frenkel 2020).

¹³ Die steigenden Anforderungen können Werner Vogels zufolge aus einer Vielzahl an Entwicklungen resultieren: „larger data sets, faster update rates, more requests, more services, tighter SLAs (service level agreements), more failures, more latency challenges, more service interdependencies, more developers, more documentation, more programs, more servers, more networks, more data centers“ (zit. n. Gray 2006, 14).

immer leistungsfähigerer, kostspieliger „High-End-Server“ zu erfüllen, gingen sie dazu über, sehr billige, aber auch deutlich fehleranfälligere Servermassenware zu gigantischen horizontal verteilten Systemen (Server-Clustern) zusammenzuschalten (McMillan/Metz 2012; O’Grady 2014). In diesen verteilten Systemen tauschen die Server permanent Nachrichten untereinander aus, sodass z.B. Rechenvorgänge auf eine Vielzahl unterschiedlicher Serverknoten aufgeteilt und parallelisiert durchgeführt werden können, ehe die Ergebnisse wieder aggregiert werden. In ihren Verwertungsstrategien trieben die Start-ups das Prinzip des „distributed computing“ software- und hardwaretechnisch auf die Spitze (Burns 2018, 2). Eingesetzt wurden z.B. nicht mehr die Prozessoren mit der höchsten Spitzenleistung, sondern konsequent die Prozessoren, die im Verhältnis zum Preis pro Einheit die höchste Leistung erzielten (Barroso et al. 2003, 24). Durch diese und zahlreiche weitere Maßnahmen gelang es den Start-ups, das Preis-Leistungs-Verhältnis ihrer IT-Infrastrukturen sprunghaft zu verbessern. Beobachter schätzten, dass z.B. Google im Jahr 2005 für jeden ausgegebenen Dollar dreimal so viel Rechenpower erzeugte wie seine Wettbewerber (Vise/Malseed 2005, 88).

Für den Aufbau dieser horizontal verteilten IT-Infrastrukturen existierte keine Blaupause. In den Start-ups begann vielmehr ein umfassender, bis heute anhaltender Lernprozess. Dabei stellten sich diesen Start-ups nicht nur historisch neuartige Engineering-Probleme, sondern auch die für die Computerwissenschaft anspruchsvollsten Herausforderungen im Umgang mit verteilten Systemen (Varley 2016; Killalea 2020, 71). Die Arbeit an diesen Aufgaben zog zahlreiche Wissenschaftler an, zumal sie nicht unter Laborbedingungen simuliert werden musste, sondern mit real vorhandenen Systemen erfolgen konnte. In hoher Regelmäßigkeit galt es, Probleme zu lösen, mit denen niemand jemals zuvor konfrontiert war (Gray 2006, 14). Schlüsselfragen waren, wie aus den fehleranfälligen und unzuverlässigen Einzelkomponenten zuverlässige und leistungsfähige Großsysteme erzeugt werden können (Vogels 2009) oder wie die Latenzzeiten der interaktiven Internetanwendungen möglichst gering gehalten werden können (Dean/Barroso 2013).¹⁴ Während auf der einen Seite die Systeme am Laufen gehalten werden mussten, wurde auf der anderen Seite in einem experimentellen Vorgehen permanent daran gearbeitet, Optimierungspotenziale zu finden und auszureizen. Ziel war es, einerseits Kriterien wie die Skalierbarkeit, die Auslastung der Server, die Verfügbarkeit der Anwendungen und die Latenzzeiten zu optimieren und andererseits die Kosten so gering wie möglich zu halten.

In diesem Zusammenhang wurden in den Start-ups neue Technologien und Konzepte für die Gestaltung verteilter Computer-Cluster entwickelt (z.B. „consistency models“, Vogels 2009). Teilweise in Zusammenarbeit mit Universitäten entstanden in dieser Phase auch neue Virtualisierungstechnologien wie Xen, die durch die konsequente Entkopplung von Hardware, Betriebssystemen und Applikationen eine höhere Auslastung vorhandener Ressourcen erzielten (Killalea 2008, 18). Zur Basisvoraussetzung wurde es, die Systeme von Grund auf fehlertolerant zu gestalten. Einzelne Serverausfälle wurden bei der Konzeption bewusst einkalkuliert. Sie sollten in Echtzeit durch redundante Server kompensiert werden und möglichst keine Auswirkungen auf die Performanz und Verfügbarkeit der Anwendungen haben (DeCandia et al.

¹⁴ Der für den Aufbau der technischen Infrastruktur bei Google verantwortliche Informatiker Urs Hölzle berichtete z.B., dass die durchschnittliche Latenzzeit, der Zeitraum zwischen der Eingabe einer Suchanfrage und dem Anzeigen eines Suchergebnisses, im Jahr 1999 noch 3,5 Sekunden betrug (Levy 2011, 37).

2007, 205). Jim Reese, der damalige Chef des IT-Betriebs von Google, beschreibt das Vorgehen folgendermaßen:

We built capabilities into the software, the hardware and the network – the way we hook them up, the load balancing, and so on – to build in redundancy, to make the system fault-tolerant (zit. n. Levy 2011, 184).¹⁵

Aus Kostengründen, aber auch aufgrund der Tatsache, dass viele Lösungen am Markt schlichtweg nicht existierten, verzichteten die Start-ups für das Verknüpfen und den Betrieb der Server-Cluster überwiegend auf proprietäre Softwarekomponenten. Stattdessen setzten sie auf der einen Seite bereits verfügbare kostenlose Open-Source-Technologien ein, z.B. Betriebssystemsoftware wie Linux, Web Server wie Apache, Datenbanksysteme wie MySQL und Programmiersprachen wie PHP/Python, und entwickelten sie für ihre Zwecke weiter (Vogl 2020, 94).¹⁶ Auf der anderen Seite entwickelten sie darüber hinaus eine Reihe software-basierter Technologien, die speziell auf die Anforderungen verteilter Systeme ausgelegt waren. Dazu zählen verteilte Datenbanksysteme (NoSQL) (Ghemawat et al. 2003; DeCandia et al. 2007), Lastenverteiler, die z.B. Nutzeranfragen verschiedenen (geographisch unterschiedlich lokalisierten) Serverclustern zuweisen, Softwarewerkzeuge für Konfigurationsmanagement und Systembeobachtung und viele weitere Komponenten einer „cluster-level infrastructure software“ (Barroso et al. 2013, 21). Diese Werkzeuge lassen die verteilten Systeme wie einen einzelnen Computer funktionieren (ebd., 24).¹⁷ Vieles, was anfangs noch händisch gemacht werden musste, konnte so automatisiert werden (Levy 2011, 199). Auch die Softwarewerkzeuge zur Massendatenverarbeitung, mit denen die Start-ups den „behavioral surplus“ (Zuboff 2018, 96) aus den Logs der Anwendungen extrahieren und nutzbar machen konnten, wurden für verteilte Systeme von Grund auf neu konzipiert. Zur Analyse der z.T. in unstrukturierten Daten vorliegenden Suchmuster und Webseitenzugriffe seiner Nutzer wurde z.B. bei Google der „MapReduce“-Algorithmus entwickelt, mit dem Berechnungen bei Datenmengen im Petabyte-Bereich parallelisiert und auf vielen Knoten verteilt durchgeführt werden können.

Das Spektrum der Maßnahmen weitete sich stetig. Bereits früh gingen Unternehmen wie Google, Amazon oder Facebook dazu über, die in ihren Datenzentren verwendete Ausrüstung, z.B. ihre Server und Router, selbst zu entwerfen und nach dem Vorbild der „fabriklosen“ (Lüthje 2001, 238) Hersteller aus der Halbleiterindustrie des Silicon Valley zu Billigpreisen von Kontraktfertigern in Asien produzieren zu lassen. Mittlerweile erzielen sie mit diesem Produktionsmodell gigantische Skaleneffekte und haben den Markt für Serverhardware, der zuvor

¹⁵ Auch bei Amazon wurde, wie Werner Vogels (2009, 40) berichtet, auf diese Weise vorgegangen: „When a system processes trillions and trillions of requests, events that normally have a low probability of occurrence are now guaranteed to happen and must be accounted for upfront in the design and architecture of the system.“

¹⁶ Vor allem in Deutschland wird Open-Source-Entwicklung in der öffentlichen Wahrnehmung häufig noch mit der freien Softwarebewegung aus den 1980er Jahren in Verbindung gebracht. Schrape (2015) hat gezeigt, dass diese Vorstellungen schon lange nicht mehr der Realität von Open-Source-Projekten entsprechen. Gerade der Übergang zu horizontal verteilten Systemen beförderte die Einbettung quelloffener Entwicklung in Unternehmensstrategien (O’Grady 2014). Um die Lasten der Technologieentwicklung auf die Industrie zu verteilen, begannen die Tech-Unternehmen, Open-Source-Projekte als Arenen konsortialer Zusammenarbeit zu nutzen. Die Hauptakteure in führenden Open-Source-Projekten sind daher mittlerweile auch „korporative Entwickler“ (Vogl 2020, 93ff).

¹⁷ Mit der Zeit wurde der Quellcode dieser Technologien zum Teil ebenfalls wieder als Open Source verfügbar gemacht, allerdings weniger von Unternehmen wie Google und Amazon als von Unternehmen wie Facebook, Twitter oder LinkedIn (O’Grady 2015, 13f).

von Herstellern wie Sun Microsystems, IBM oder HP dominiert wurde, vollständig verändert (McMillan/Metz 2012). Nachdem Google zu Beginn seine selbst zusammengestellten „Server-Racks“ noch in den Datenzentren von kommerziellen Betreibern wie Exodus in angemieteten Bereichen neben den Server-Systemen anderer Internetunternehmen platziert hatte, ging das Start-up in Erwartung exponentiell steigender Anforderungen, aber auch im Bewusstsein der strategischen Bedeutung seiner verteilten Server-Systeme rasch dazu über, eigene Datenzentren aufzubauen. Jim Reese zufolge erkannten sie, dass „our true advantage was actually the fact that we had this massive parallelized redundant computer network (...) and we realized that maybe it’s not in our best interests to let our competitors know“ (zit. n. Levy 2011, 198). Auch die Datenzentren wurden durch die Mitarbeiter von Google von Grund auf neu konzipiert, fortlaufend weiterentwickelt und auf möglichst geringe Kosten bei steigender Leistung ausgelegt – z.B. durch Senkung des Energieverbrauchs für die Kühlung (Barroso et al. 2013, 67ff). Neben dem Aufbau dieser „digital cathedrals“ (Mills 2020), deren Energieverbrauch allen Maßnahmen zu dessen Reduktion zum Trotz immer neue Rekorde bricht (Greenpeace 2017, 15), rückte auch die Infrastruktur für den Transport der Daten in den Fokus. Google und andere begannen Glasfasernetze aufzukaufen (Levy 2011, 186) und Unterseekabel zu legen (Satariano 2019). Durch immer höhere Investitionssummen werden diese in Privatbesitz befindlichen „globalen informatischen Infrastruktur(en)“ (Haug 2020, 24) in der Folge immer weiter ausgebaut.

Diese gigantischen verteilten Systeme aus Servermassenware, die durch die Entwicklung neuer Softwarewerkzeuge dynamisch skaliert und in der Bedienung wie ein einzelner Computer funktionieren, bilden den ersten Baustein des technologisch-organisatorischen Fundaments von Tech-Unternehmen.

3.2 „Microservices“: Modularisierung als Architekturprinzip verteilter Softwareapplikationen

Komplementär zum Aufbau großtechnischer Systeme rückte die Softwarearchitektur der Internetanwendungen in den Fokus der Innovationsbemühungen der Start-ups. Im Web 1.0 hatte ein Architektur-Modell dominiert, bei dem Applikationen als „single unit“ konzipiert werden (Lewis/Fowler 2014). Sie verfügen dabei zwar über unterschiedliche Schichten: z.B. eine clientseitige Benutzeroberfläche¹⁸, eine serverseitige Applikation und eine Datenbank, aber die Komponenten der Applikation sind sehr eng miteinander verknüpft. Die gesamte Darstellungs- und Geschäftslogik und sämtliche Funktionalitäten sind zumeist in der serverseitigen Applikation enthalten. Für Applikationen, die auf diese Weise gestaltet sind, hat sich in der Fachsprache die Bezeichnung „Monolith“ etabliert (ebd.).

Bei der Weiterentwicklung, Pflege und Skalierung ihrer durch Geschäftswachstum immer komplexer werdenden Software-Monolithen und immer riesigeren Datenbanken stießen die Entwickler:innen und Systemadministrator:innen in den Start-ups zunehmend an Grenzen. Zum einen mussten etwa bei der Erweiterung der Applikation um neue Funktionalitäten immer wieder große Teile der gesamten Applikation neu geschrieben werden. Dies ging mit

¹⁸ Ein wichtiger Aspekt war hier bereits die Entwicklung der dynamischen Programmiersprache JavaScript bei Netscape und des AJAX-Frameworks, die es ermöglichten, internetbasierte Anwendungen zu entwickeln, die sich wie desktopbasierte Anwendungen verhalten (eine Webseite muss bei Änderungen nicht vollständig neu geladen werden).

enormem Koordinationsaufwand zwischen den Teams einher und bremste die Geschwindigkeit bei der Weiterentwicklung der Lösung. Zum anderen musste zur Skalierung einzelner Prozesse der Applikation immer die gesamte Applikation skaliert werden, um die Last zu bewältigen (Vogels 2019). Die engen Interdependenzen in der Architektur der Applikation behinderten folglich Praktiken wie das Upgrading und Code Refactoring, die Auslieferung neuer Funktionalitäten sowie ihre Skalierung (Winters et al. 2020).

In dem Maße, wie die Anforderungen an die Applikationen wuchsen, potenzierten sich diese Probleme. Um mit dem Geschäftswachstum Schritt halten zu können, verfolgten daher z.B. die Entwickler bei Amazon für die E-Commerce-Anwendung etwa ab dem Jahr 2001 einen anderen Ansatz. Das Ziel war, die Skalierbarkeit als Konstruktionsprinzip in der Architektur der Anwendung zu verankern. Werner Vogels, CTO von Amazon, schilderte im Rückblick die Grundüberlegung in einem Blogpost folgendermaßen:

(...) scalability cannot be an after-thought. It requires applications and platforms to be designed with scaling in mind, such that adding resources actually results in improving the performance or that if redundancy is introduced the system performance is not adversely affected. Many algorithms that perform reasonably well under low load and small datasets can explode in cost if either requests rates increase, the dataset grows or the number of nodes in the distributed system increases (Vogels 2006).

Vor diesem Hintergrund wurde die monolithische Architektur der Applikation modularisiert und in eine Vielzahl lose gekoppelter Software-Services aufgespalten (Jamshidi et al. 2018, 25).¹⁹ Rob Brigham, zu dieser Zeit im Management der IT bei Amazon tätig und in diesen Prozess involviert, beschrieb in einem Vortrag das Vorgehen folgendermaßen:

We went through the code, and pulled out functional units that served a single purpose, and we wrapped those with a web service interface. For example, there was a single service that rendered the „Buy“-button on the retailer’s product detail pages. Another had the function of calculating the correct tax during checkout (zit. n. Fulton 2015).

Jeder dieser Services umfasste fortan ein Element der Geschäftslogik des Webshops mitsamt den dafür erforderlichen Datensätzen und wurde hinter einer Anwendungsschnittstelle (API) zusammengefasst. Die einzelnen Services können seither mit den anderen Services allein über diese Schnittstellen kommunizieren (Yegge 2011). Sobald die Seite von einem Nutzer besucht wird, rufen Applikationsserver über auf HTTP basierende Netzwerkprotokolle wie RESTful oder SOAP die Funktionalitäten und Informationen der einzelnen Dienste ab und aggregieren sie z.B. zur Startseite des Webshops.²⁰ Von Mechanismen für die automatische Erkennung von Diensten in einem Rechnernetz über neue ressourcenschonende Virtualisierungstechnologien auf der Applikationsebene wie „Container“ sowie Technologien zu ihrer Verwaltung wie „Kubernetes“ bis hin zu Programmbibliotheken für fehlertolerante Kommunikation zwischen den Diensten wurde und wird bei Amazon sowie in vielen anderen Start-ups und Open-Source-Projekten in mehreren Wellen eine Vielzahl von Softwarewerkzeugen für modularisierte

¹⁹ Lewis/Fowler (2014) weisen darauf hin, dass die Wurzeln modularer Gestaltungsprinzipien in der Softwareentwicklung bis zur Entwicklung von Unix zurückreichen und z.B. auch in der Objektorientierten Programmierung eine wichtige Rolle spielen.

²⁰ Bereits im Jahr 2006 erläuterte Vogels, dass „if you hit the Amazon.com gateway page, the application calls more than 100 services to collect data and construct the page for you“ (Vogels zit. n. Gray 2006, 16). Diese Zahl ist mittlerweile deutlich nach oben gegangen.

Applikationen entwickelt (Jamshidi et al. 2018, 26). Das neue Architekturmuster hat zum einen den Vorteil, dass einzelne Services der Applikation weitgehend unabhängig von den anderen Services weiterentwickelt²¹, betrieben und skaliert werden können.²² Steigt etwa die Nachfrage nach einer spezifischen Funktion einer Applikation, kann allein der Service skaliert werden, der diese Funktionalität umfasst (z.B. der Einkaufswagen im Falle einer Rabattaktion), ohne dass die gesamte Architektur der Applikation skaliert werden muss. Dadurch können enorme Ressourcen-, Zeit- und Kostenersparnisse erzielt werden. Insgesamt soll Amazon in sechs Jahren ca. \$ 1 Mrd. in das „Re-Architecting“ seiner E-Commerce-Anwendung investiert haben (Kim 2018, 2354). Darüber hinaus können die Anwendungsschnittstellen, über die definiert wird, wie auf die Funktionalität und die Daten der Dienste zugegriffen werden kann, prinzipiell auch externen Entwicklern über das Internet für die Entwicklung von Applikationen verfügbar gemacht werden (Yegge 2011).

Die Modularisierung der Applikation schuf bei Amazon die Grundlage für eine umfassende Transformation der Entwicklungsorganisation. Diese Transformation verlief keineswegs reibungslos, sondern gestaltete sich als ein komplexer, teilweise schmerzhafter Lernprozess:

The awkward and extended transition to this new code base (...) took over three years and caused all kinds of excruciating pain among its network engineers, who were forced to carry pagers so they could respond promptly to the numerous problems (Stone 2013, 202).

Im Zentrum stand die Neugestaltung der Entwicklungsprozesse und der Inbetriebnahme von Veränderungen an der Applikation. Während zuvor fast alle Entwicklerteams an derselben Codebasis der monolithischen Applikation gearbeitet hatten und selbst kleinere Veränderungen entsprechend hohen Koordinationsaufwand erforderten, wurden nun teilautonome Teams gebildet und einzelnen Services zugeordnet. Über die Entwicklung standardisierter Vorgehensmodelle und einer „internal culture of API Documentation“ (Lawson 2021, 38) auf der einen und den Aufbau automatisierter Testumgebungen und „Release-Pipelines“ (CI/CD) (Shahin et al. 2017) auf der anderen Seite wurden Praktiken etabliert und Werkzeuge entwickelt, die die Teams in die Lage versetzten, die einzelnen Dienste der Applikation unabhängig voneinander weiterzuentwickeln und in Betrieb zu nehmen. Während Amazon bei seiner Applikation im Jahr 2001 zwölfmal *jährlich* neue Softwarekomponenten in Betrieb nahm, schnellte diese Zahl bis zum Jahr 2013 auf 136.000 sog. Deployments *pro Tag* (Kim 2020). Mit der Fähigkeit, Software kontinuierlich in Betrieb nehmen zu können, wurde ferner die Grundlage für Praktiken wie das A/B Testing geschaffen, bei denen Entwickler z.B. neue Funktionalitäten mit einem bestimmten Kreis von Nutzern live ausprobieren und das Nutzerverhalten auswerten, ehe sie entweder verworfen, modifiziert und erneut getestet oder allen Nutzern der Anwendung verfügbar gemacht werden (Feitelson et al. 2013, 10; Boes et al. 2019, 128).

Die Entwicklung von Softwarewerkzeugen zur automatisierten Inbetriebnahme bspw. neuer Funktionalitäten trug bei Amazon zugleich dazu bei, die zuvor organisatorisch zementierte

²¹ Sie können auch mit einem eigenen Technologie-Stack, z.B. spezialisierten Datenbanksystemen, Programmiersprachen und Sicherheitsfunktionalitäten, versehen werden.

²² Inwiefern dies gelingt, ist, wie Lewis/Fowler (2014) anführen, nicht zuletzt eine Frage der Umsetzung in der Praxis: „That’s not an absolute, some changes will change service interfaces resulting in some coordination, but the aim of a good microservice architecture is to minimize these through cohesive service boundaries and evolution mechanisms in the service contracts.“

Rollenverteilung zwischen der Entwicklung und dem Betrieb der Software aufzuheben (Newman 2015, 15). Teams wurden funktionsübergreifend besetzt und sollten sich um den gesamten Lebenszyklus der Services kümmern. Die Verankerung des Prinzips „you build it, you run it“ (Vogels zit.n. Gray 2006, 16) führte zudem dazu, dass sich auch die Entwickler mit dem Verhalten ihres Codes im täglichen Betrieb auseinandersetzen müssen und in direkten Kontakt mit den Nutzern der Dienste kommen.²³ Schritt für Schritt wurden so die technologischen und organisatorischen Voraussetzungen geschaffen, um jedem dieser Teams die vollständige Geschäftsverantwortung für einen einzelnen Service zu übertragen. Als Unternehmen im Unternehmen wurden die Teams mit weitreichenden Entscheidungshoheiten für die Gestaltung ihrer Services ausgestattet, während zugleich die Performance ihrer Services z.B. in einer intern ausgewiesenen Gewinn- und Verlustrechnung datenbasiert erfasst und visualisiert wurde (Stringfellow 2018; Lawson 2021, 39). Leistungsbasierte monetäre Anreize (z.B. über Aktienoptionen) komplementierten diese Arbeitsform, sodass die Teams, wie es ehemalige Mitarbeiter gegenüber Journalisten beschrieben haben, mit ihren Services in einer Art „purposeful darwinism“ (Kantor/Streitfeld 2015) untereinander wetteifern. Der Softwarearchitekt Michael T. Nygard schildert dies im Gespräch mit Gene Kim folgendermaßen:

So if you talk to the people [at Amazon] working on the services, if your service is not getting traffic it's not getting customers it's gonna get deleted and you'll get reassigned. If you're getting customers than you'll get to add on to your service. It's just the opposite of the way most companies work (zit. n. Kim 2020).

Durch den Aufbau großtechnischer verteilter Systeme, die Modularisierung der Applikationsarchitekturen und die Einführung neuer Arbeitsmethoden konnten Unternehmen wie Google und Amazon komplexe Internetanwendungen, wie die an einen echtzeitbasierten Online-Auktionsplatz für Werbung gekoppelte Suchmaschine, die das exponentiell wachsende World Wide Web vollständig indiziert, oder die E-Commerce-Plattform auch bei Hunderten Millionen Zugriffen gleichzeitig dynamisch skalieren, global hochverfügbar halten, permanent auf Basis datenbasierter Beobachtungen des Nutzerverhaltens weiterentwickeln und kosteneffizient betreiben. Es gelang ihnen dadurch, das Internet in neuartige Verwertungsstrategien einzubetten und in Rekordgeschwindigkeit den Werbemarkt oder Sektoren wie den Einzelhandel zu erschüttern. Sie konnten sich zudem gegenüber den zeitgleich wie Pilze aus dem Boden schießenden Start-up-Mitbewerbern durchsetzen und die erzielten Vorsprünge in immer höhere Markteintrittsbarrieren für potenzielle Konkurrenten bis hin zur Errichtung von „Quasi-Monopolen“ (Staab 2019, 105) auf ihren Feldern ummünzen.

Darüber hinaus transformierten die Tech-Arbeiter in den Start-ups mit diesen Basisinnovationen, die in der Praxis unaufhörlich weiterentwickelt werden, IT-Infrastrukturen von einer knappen zu einer tendenziell im Überfluss vorhandenen Ressource (Kushida et al. 2015). Sie schufen die Voraussetzungen, um Massendatenverarbeitung in immer weiteren Anwendungsfeldern einzusetzen und in neuartigen Formen (z.B. Machine Learning) zu betreiben.

²³ Wie Lewis/Fowler (2014) bemerken, können regelmäßige Pager-Alarme, die Entwickler mitten in der Nacht aus dem Bett holten, als „a powerful incentive to focus on quality when writing your code“ wirken. Dadurch hat sich ihre Arbeitssituation substanziell verändert. Die Integration von Entwicklung und Betrieb, für die später Bezeichnungen wie DevOps (Kim et al. 2016) und Site Reliability Engineering (Beyer et al. 2016) geprägt wurden, machte in den Tech-Unternehmen Schule und wird im Kontext des Internet der Dinge auch in Industrieunternehmen adaptiert (Ziegler 2020, 217).

3.3 „Cloud Computing“: Dynamisch skalierbare IT-Infrastrukturen als „Versorgungsgut“

Zu den ersten Unternehmen, die systematisch damit begannen, das Verwertungspotenzial zu erschließen, das in einer eigenständigen Kommerzialisierung dieser Basisinnovationen lag, zählte Amazon mit seiner Sparte für Webdienste. Zur Genese dieser epochalen Geschäftsmodellinnovation bei Amazon finden sich in der Literatur im Detail unterschiedliche Akzentuierungen. Huckman et al. (2012, 4f) zufolge begann die Entwicklungsorganisation von Amazon im Zuge der Modularisierung der Applikationsarchitektur Mitte des Jahres 2002 damit zu experimentieren, Informationen zu den Produkten des Webshops auch für Entwickler externer Partnerunternehmen über eine Anwendungsschnittstelle verfügbar zu machen. Partnerunternehmen konnten diese Informationen unmittelbar in ihre Webseiten einbetten und auf diese Weise bei Amazon erhältliche Waren bewerben sowie direkt mit dem Webshop verknüpfen. Für jeden Verkaufserlös, den sie Amazon vermittelten, erhielten sie eine Kommission von 5 bis 8 Prozent auf den Verkaufspreis von Amazon. Darüber hinaus wurde in dieser Phase in der Entwicklungsorganisation analysiert, für welche Tätigkeiten die Entwickler-Teams der E-Commerce-Anwendung von Amazon ihre Arbeitszeit aufwenden. Dabei wurde festgestellt, dass sämtliche Teams sehr viel Arbeitszeit für dieselben, vergleichsweise repetitiven Tätigkeiten, z.B. die Verwaltung der von ihren Services benötigten Rechen- oder Speicherkapazitäten, verwendeten. Um die Entwicklerteams von diesen Tätigkeiten freizumachen, sodass ihnen mehr Arbeitszeit für die Weiterentwicklung ihrer Funktionalitäten zur Verfügung steht, wurde innerhalb von Amazon ein Set aus Infrastruktur-Services aufgebaut (Foley 2009). In diesem Zusammenhang entstanden Services wie Dynamo für das Speichern von Objekten, deren APIs von den Entwicklerteams der E-Commerce-Anwendung z.B. für den Einkaufswagen oder die Betrugserkennung genutzt werden konnten (DeCandia et al. 2007, 214).

Angesichts des großen Erfolgs dieser Maßnahmen und der Beobachtung, dass andere Start-ups ähnliche Initiativen lancierten, wurde im Management die strategische Bedeutung dieser Experimente reflektiert.²⁴ Andy Jassy schildert die Überlegungen im Rückblick folgendermaßen:

It caused us to step back and wonder if something broader was going on. If developers would build applications from scratch using Web services, and if a broad array of Web services existed (which we believed would be the case), then the Internet would become the operating system. We asked ourselves, if the Internet became the operating system, what would the key elements be, which had already been built, and which would Amazon be best-equipped to provide for the community? At the time we were looking at it in 2003, none of the key elements of the Internet operating system had been built. When we thought about Amazon's strengths as a technology company that had simply applied its technology to the retail space first, and what Amazon had done well over the last decade, we realized we could provide a lot of the key building blocks (zit. n. Huckman et al. 2012, 4).

²⁴ Brad Stone (2013, 210) verweist auf die Bedeutung, die der einflussreiche Medienunternehmer Tim O'Reilly und Schöpfer des Terminus „Web 2.0“ hatte. Bei mehreren Treffen mit Jeff Bezos in dieser Zeit soll er ihn von der strategischen Bedeutung von APIs als Bausteinen für Plattformstrategien überzeugt haben. Als Bezos von seinem IT-Management erfuhr, dass einzelne Teams in seiner Entwicklungsorganisation bereits mit der Verfügbarmachung von APIs für externe Entwickler experimentierten, fasste er diese Bemühungen in einem strategischen Innovationsprojekt zusammen.

Das Management interpretierte die Ergebnisse dieser Experimente als Vorboten einer Entwicklung des Internets zum Betriebssystem immer weiterer Anwendungen und setzte darauf, die Kernkompetenzen, die Amazon im Kontext der Bereitstellung seiner E-Commerce-Anwendung aufgebaut hatte, in Verwertungsstrategien zur Erschließung dieser Potenziale einzusetzen.

In der Folge nahm Amazon die Arbeit daran auf, einzelne Elemente seiner Technologieplattform externen Entwicklern über das Internet verfügbar zu machen. Im Jahr 2006 brachte die Sparte Amazon Web Services (AWS) mit Simple Storage Service (S3) eine Anwendung für das Speichern von Objekten und mit Elastic Compute Cloud (EC2) eine Anwendung, welche die Nutzung von Rechenkapazitäten in den Datenzentren von AWS ermöglicht, auf den Markt, die ersten Produkte, die genuin für diesen Zweck entwickelt wurden (Stone 2013, 213ff). Knapp drei Jahre später wurden Funktionalitäten für das Monitoring, die elastische Lastenverteilung und die automatische Skalierung verfügbar gemacht, durch die das Management der dynamischen IT-Infrastrukturen (EC2) für Anwenderunternehmen entscheidend vereinfacht wurde (Barr 2009).²⁵ Dieses Geschäftsmodell wurde „Infrastructure-as-a-Service“ genannt. Wie Vogels im Rückblick erzählt, flossen die Erfahrungen mit der E-Commerce-Anwendung in die Konzeption dieser Infrastruktur-Services ein, die von Anfang an eine skalierbare evolutionäre Architektur erhielten:

So, I think one of the tenets up front was don't lock yourself into your architecture because two or three orders of magnitude of scale and you will have to rethink it (zit. n. Killalea 2020, 68).

Die Architektur des Service S3 etwa bestand anfangs aus acht Microservices und wuchs bis zum Jahr 2019 auf 262 Microservices an (ebd.).

In dem Maße, wie nun verteilte Rechen-, Speicher- und Netzwerkkapazitäten, Datenbanksysteme und in den Folgejahren viele weitere Hardware- und Softwarebausteine als Dienste über eine Internetverbindung bereitgestellt wurden, konnten die Erfahrungen mit der Gestaltung verteilter IT-Infrastrukturen in die Wirtschaft diffundieren. Anwenderunternehmen können dynamisch skalierbare IT-Infrastrukturen gegen Gebühr für die Entwicklung und den Betrieb ihrer Anwendungen nutzen. Aus der Perspektive der Anwenderunternehmen werden diese IT-Ressourcen so zu Versorgungsgütern, die analog zu elektrischer Energie oder Wasser nicht mehr im eigenen Unternehmen erzeugt, betrieben und gewartet werden müssen, sondern je nach Bedarf nahezu in Echtzeit über das Internet bezogen werden können (Armbrust et al. 2009, 1). Für dieses Modell zur Bereitstellung von IT-Infrastrukturen fand die Bezeichnung „Cloud Computing“ Verbreitung. Regalado (2011) zufolge war diese Bezeichnung zwar bereits im Jahr 1996 intern in der Innovationsabteilung des Serverherstellers Compaq geprägt worden, sie setzte sich dann aber, wie auch Boes et al. (2019, 116) herausstellen, vor allem infolge ihrer Verwendung durch Eric Schmidt auf einer Industriekonferenz von Google im August 2006 durch:

It starts with the premise that the data services and architecture should be on servers. We call it cloud computing – they should be in a ‚cloud‘ somewhere. And that if you have the right kind of browser or the right kind of access, it doesn't matter whether you have a PC or a Mac or a

²⁵ Mittlerweile bietet AWS über 175 Clouddienste an. Seit die Umsätze von AWS eigens ausgewiesen werden, wuchsen sie von \$ 3,1 Mrd. im Jahr 2013 auf \$ 35 Mrd. im Jahr 2019.

mobile phone or a BlackBerry or what have you – or new devices still to be developed – you can get access to the cloud (zit. n. Google Press Center 2006).

Seine Einführung veränderte die IT-Industrie fundamental. Ray Ozzie (2005), ehemaliger Chefarchitekt von Microsoft, brachte diese Entwicklung in einem internen Memo an seine Kollegen bei Microsoft bereits im Jahr 2005 auf den Begriff „Internet Services Disruption“. Erst einige Jahre später gelang es jedoch Unternehmen wie Microsoft, Google, IBM, Alibaba oder Tencent, nachzuziehen und ihrerseits ein Geschäft mit „Cloud Services“ aufzubauen.

Statt aufwendig eigene Datenzentren zu betreiben, können nun selbst kleine Start-ups innerhalb kürzester Zeit auf die elastisch skalierbaren IT-Infrastrukturen in einer Public Cloud für den verteilten Betrieb ihrer software- und datenbasierten Anwendungen zugreifen und so sehr schnell globale Reichweite erlangen. Der Aufbau ihrer Anwendungen kann durch die Elastizität der Cloud-Infrastrukturen zugleich mit dem Wachstum in den Nutzerzahlen und Umsätzen dynamisch Schritt halten. Dabei haben sie die Möglichkeit, nur für die IT-Ressourcen zu bezahlen, die sie tatsächlich benötigen. In dem Maße, wie so die Eintrittsbarrieren in den Aufbau daten- und softwarebasierter Internetanwendungen für neue Start-ups wie Uber, Spotify oder Airbnb signifikant reduziert wurden, wirkten Cloud-Konzepte als „a platform for innovation and entrepreneurship“ (Kushida et al. 2011, 214). Zugleich begeben sich die Anwenderunternehmen jedoch in Abhängigkeit von den Anbietern der Cloud Services (Staab 2019, 190).²⁶ Während die Produktion und Distribution von Versorgungsgütern wie Energie und Wasser aufgrund vergleichbarer Abhängigkeiten der Anwenderunternehmen und dadurch, dass sie weitgehend in öffentlicher Hand sind, stark reguliert ist, ist dies für „Cloud Computing“ bisher noch kaum der Fall (Kushida et al. 2011, 215).²⁷

Doch auch wenn der Zugriff auf IT-Ressourcen nunmehr einen Mausklick entfernt scheint, stellte die Entwicklung und der Betrieb komplexer Anwendungen auf der Basis von IT-Infrastrukturen in einer Public Cloud die Anwenderunternehmen vor enorme Herausforderungen. Auch hierfür existierten keine Blaupausen. Einer der Pioniere auf diesem Feld war das Start-up Netflix. Ursprünglich in den USA als Internetanwendung für den Verleih von DVDs gestartet, arbeitete das Unternehmen in Anbetracht der anhaltenden Verbesserungen in den Datenübertragungsraten des Internet ungefähr ab Mitte der 2000er Jahre daran, eine Streaminganwendung für Serien, Filme etc. über das Internet bereitzustellen. Die monolithische Applikation, die für die Internetanwendung für den DVD-Verleih aufgebaut worden war, konnte noch auf einer vergleichbar überschaubaren Zahl eigener Server betrieben werden. Aufgrund wachsender Anforderungen, z.B. steigender Nutzerzahlen, hatten die Beschäftigten von Netflix allerdings auch in dieser Phase immer wieder mit Datenverlust zu kämpfen. Es zeichnete sich daher bereits ab, dass die vorhandenen IT-Infrastrukturen für den angestrebten Aufbau einer Streaminganwendung bei weitem nicht ausreichen würden. Adrian Cockcroft, Cloud-

²⁶ Diese Abhängigkeit wird zum einen dann sichtbar, wenn die Cloud Services ausfallen und die darauf basierenden Anwendungen nicht mehr verfügbar sind (siehe dazu z.B. Schonfeld 2008). Zum anderen binden sich die Start-ups, indem sie Cloud Services in ihre Anwendungen integrieren, an die Technologien der Cloud-Anbieter. Um diesem „Lock-in-Effekt“ entgegenzuwirken, wurde in Open-Source-Projekten eine Reihe von Technologien entwickelt, die zumindest den Wechsel zwischen Anbietern erleichtern und die Entwicklung von „hybriden“ bzw. „Multi“-Cloud-Strategien ermöglichen sollen.

²⁷ Zu aktuellen Bestrebungen seitens der EU-Kommission im Rahmen des Digital Service Package siehe Piétron/Staab 2021.

Architekt des Start-ups, schildert die Entscheidungskonstellation, vor der das Unternehmen stand, im Rückblick folgendermaßen:

It was clear that we needed largescale redundant datacenters, but we couldn't easily predict how much datacenter capacity we would need for the rapidly growing streaming business and where it should be located. We had also had a fairly painful experience moving from a single datacenter to a pair of small ones (leased cages), and needed to decide whether to invest heavily in the staff and skills needed to run a large and high-growth-rate datacenter infrastructure, or outsource and leverage an external cloud supplier (zit. n. Farrow 2012, 44).

Das Start-up führte vor diesem Hintergrund parallel Experimente durch. Während es auf der einen Seite eine deutliche Aufstockung der eigenen Datenzentrumskapazitäten testete, prüfte es auf der anderen Seite die Möglichkeit, IT-Infrastrukturen von AWS zu nutzen. Letztere Option lieferte deutlich bessere Ergebnisse, sodass das Management die Entscheidung traf, das Geschäftsmodell auf dieser Basis aufzubauen. Aus ersten Untersuchungen und Tests entspann sich in der Folge ein umfassender Lernprozess. Über mehrere Jahre wurden, wie Cockroft berichtet, nach und nach die eigenen Server abgeschaltet und Anwendungsfälle in die Cloud migriert:

Through 2009, we explored the cloud platform with several pathfinder projects and non-customer-facing workloads such as encoding and Hadoop-based log analysis.²⁸ In early 2010 we brought up the first customer-facing workloads, starting with the simplest ones with fewest dependencies, and gradually filling in the data sources until almost everything is running in the cloud, but with the data resident in both cloud and datacenter. In 2011 we gradually moved the ‚source of truth‘ systems into the cloud, with copies in the datacenter as needed (ebd.).

Um das Potenzial skalierbarer IT-Infrastrukturen nutzen zu können, wurde in diesem Prozess auch bei Netflix die monolithische Architektur der Anwendung aufgelöst. Die Anwendung wurde in eine Vielzahl lose gekoppelter Microservices aufgeteilt, denen einzelne Teams zugeordnet wurden, die auch hier die Verantwortung für deren Weiterentwicklung, Betrieb und Wirtschaftlichkeit übernehmen sollten (Bukowski et al. 2016).²⁹ In der Entwicklungsorganisation bei Netflix wurden mehrere Softwarewerkzeuge für die Entwicklung und den Betrieb modularisierter Anwendungen mit Cloud-Infrastrukturen entwickelt und eingesetzt. Dazu zählen z.B. „Chaos-Engineering“-Technologien, die nach dem Zufallsprinzip automatisiert Fehler in der IT-Infrastruktur auslösen, um so permanent die Zuverlässigkeit der Anwendung im laufenden Betrieb in der AWS-Cloud zu testen (Izrailevsky/Tseitlin 2011). Mit der Zeit wurde ihr Quellcode von Netflix in Open-Source-Projekten verfügbar gemacht, um die Weiterentwicklung gemeinsam mit anderen Unternehmen voranzutreiben und die Kosten dafür zu teilen.

Der Aufstieg von Netflix zu einem der weltgrößten Streamingdienste mit seinem gewaltigen Datendurchsatz demonstrierte die Leistungsfähigkeit, Zuverlässigkeit und Kosteneffizienz von IT-Infrastrukturen in einer Public Cloud (Vogl 2017). In der Folge begannen zunehmend auch

²⁸ Bei Apache Hadoop handelt es sich um ein quelloffenes Rahmenwerk, das verschiedene Softwarewerkzeuge (u. a. den bereits erwähnten zuerst bei Google proprietär entwickelten MapReduce-Algorithmus) für Großdatenanalysen auf Computerclustern umfasst.

²⁹ Darüber hinaus hat Netflix im Jahr 2012 in Partnerschaft mit Internet-Service-Providern das eigene Content-Delivery-Netzwerk „Open Connect“ aufzubauen begonnen, um Inhalte lokal vorhalten zu können und weniger Internetverkehr zu erzeugen. Im Jahr 2016 waren bereits über 1.000 von Netflix entwickelte Hardwareeinheiten in den weltweit verteilten Datenzentren von ISPs lokalisiert (Florence 2016).

etablierte Unternehmen damit, Teile ihrer IT-Infrastrukturen auf die rasant wachsenden Clouds der Tech-Unternehmen zu migrieren und eigene Rechenzentren abzuschalten. Dynamisch skalierbare IT-Infrastrukturen setzen sich dadurch in immer weiteren Teilen der Wirtschaft durch.

4. Fazit und Ausblick: Die neuen IT-basierten Maschinensysteme als Fundament von Tech-Unternehmen

Der Blick ins Innere der Tech-Unternehmen hat gezeigt, dass die kosteneffiziente Bereitstellung und kontinuierliche Weiterentwicklung von Anwendungen über das Internet bei global mehreren Milliarden Zugriffen täglich keineswegs so trivial ist, wie sie den Nutzer:innen der PCs, Smartphones oder Tablets erscheint. Sie gründet vielmehr auf einer neuen „Technobasis“ (Haug 2003, 10), deren Entwicklung und Beherrschung den Aufbau umfassender technologischer und organisatorischer Kompetenzen erforderte und nicht zuletzt mit zahlreichen computerwissenschaftlichen Durchbrüchen einherging (Helland 2016). Die Menschen in den Start-ups der Internetwirtschaft konnten diese Fähigkeiten nicht einkaufen; sie haben sie aufgebaut. Andy Jassy formuliert dies aus der Perspektive des Managements im Rückblick folgendermaßen:

(...) most people (excluding CIOs and CTOs) didn't realize the extensive and complex technology infrastructure required to operate Amazon. You could not buy the software necessary to operate at Amazon's scale. You may have been able to buy pieces, but they would have needed to be highly customized and carefully strung together. Amazon built virtually every piece of software necessary to run a Web business that could scale, on demand, to virtually any level imaginable. Only a handful of companies around the world could claim that level of software competency (zit. n. Huckman et al. 2012, 4).

Inmitten des Zusammenbruchs der New Economy wurden an der US-amerikanischen Westküste neue Technologien, Konzepte und Praktiken für die kosteneffiziente Skalierung von IT-Infrastrukturen entwickelt und umgesetzt. Der Aufbau der damit verbundenen Kernkompetenzen bildete das Fundament für die Entwicklung, den Betrieb und das dynamische Wachstum der komplexen Internetanwendungen. Als Verwertungsstrategie ist das Tech-Unternehmen vollständig von ihrem Funktionieren abhängig. Ausfälle gefährden die Zielerreichung.

In diesem neuartigen Spektrum an Fähigkeiten können analytisch zwei Ebenen unterschieden werden. Die IT-Infrastrukturen basieren auf der einen Seite auf gigantischen verteilten Systemen aus Servermassenware, die durch die Entwicklung neuer Softwarewerkzeuge dynamisch skaliert und in der Bedienung wie ein einzelner Computer funktionieren. Auf der anderen Seite gründen sie auf einer modularisierten Applikationsarchitektur, die komplexe Anwendungen in eine Vielzahl lose gekoppelter Microservices aufspaltet. Diese Microservices können weitgehend unabhängig voneinander weiterentwickelt, betrieben und gepflegt werden, sodass die Internetanwendungen nicht nur kosteneffizient, sondern auch schneller an sich verändernde Marktbedingungen angepasst werden können. Im Zusammenspiel bilden die daraus entstehenden dynamisch skalierbaren IT-Infrastrukturen die *neuen IT-basierten Maschinensysteme* von Tech-Unternehmen. Für das Tech-Unternehmen als Strategie der (Einzel-)Kapitalverwertung kommt ihnen vergleichbare Bedeutung zu wie den klassischen Maschinensystemen für den Industriebetrieb.

Die Beherrschung der neuen IT-basierten Maschinensysteme ist ausschlaggebend dafür, dass aus einem Webshop für Bücher ein „Everything Store“ oder aus einem College-Netzwerk ein globales soziales Netzwerk mit über zwei Milliarden Nutzern werden konnte.³⁰ Durch den Aufbau riesiger Clouds können sie von Anwenderunternehmen als Versorgungsgut bezogen werden und sich in immer weiteren Teilen der Wirtschaft als Quasistandard durchsetzen. Indes hält die Arbeit an ihrer Weiterentwicklung bei rasanter Geschwindigkeit an. Ähnlich wie Marx bei der Einführung der Maschinerie in den Industrieunternehmen Großbritanniens feststellte, „folgen Schlag auf Schlag neue Methoden zu ihrer wohlfeilern Reproduktion und Verbesserungen, die nicht nur einzelne Teile oder Apparate, sondern ihre ganze Konstruktion ergreifen“ (MEW 23, 427). Die Kompetenzen, die im Umgang mit diesen neuen IT-basierten Maschinensystemen in den Tech-Unternehmen aufgebaut wurden, bildeten zugleich den Nährboden für eine Reihe weiterer Basisinnovationen (z.B. im Bereich von Machine Learning).

Die vorgelegte Analyse zum Fundament von Tech-Unternehmen demonstriert ferner die Ertragskraft einer soziologischen Forschungsperspektive, welche Tech-Unternehmen in Anlehnung an den Münchner Betriebsansatz „als Strategie“ begreift und ihre dynamische Dimension zu ergründen sucht. Während soziologische Analysen bisher primär die *Strategiemuster* von Tech-Unternehmen erfassen konnten, rückt dadurch ihre *Strategiebildung* und die Arbeit der Menschen in den Tech-Unternehmen ins Bild. Sichtbar werden die komplexen Lernprozesse, in denen Tech-Arbeiter:innen die technologisch-organisatorischen Kernkompetenzen und das Erfahrungswissen aufgebaut haben, welche die Entwicklung und Umsetzung der Strategiemuster und so z.B. die Monopolisierung von Netzwerkeffekten überhaupt erst ermöglichten. Für eine ganze Reihe dieser Kernkompetenzen scheint zu gelten, was die Managementforscher Teece et al. (1997, 518) über die aus ihrer Sicht für den Erfolg von Unternehmen ausschlaggebenden „dynamic capabilities“ schrieben: „they typically must be built, because they cannot be bought.“ In dem Maße, wie „Tech“ mit dem Aufstieg des Internet der Dinge zu einem „Layer“ (Mims 2018) in den Verwertungsprozessen jeder Branche wird, machen gerade etablierte Konzerne und Unternehmen immer wieder diese Erfahrung. Entsprechend prominent rückt aktuell die Frage „how to build a tech company“ (Hendel 2020ff) auf die strategische Agenda dieser Unternehmen, ihrer Führungskräfte und Beschäftigten. In immer mehr Bereichen der Wirtschaft werden Tech-Unternehmen zu den zentralen Orten, „an denen sich die Organisierung des gesellschaftlichen Produktionsprozesses vollzieht“ (Altmann/Bechtle 1971, 21). Indem soziologische Forschung den Blick auf die komplexe soziale und kulturelle Praxis von Tech-Unternehmen und die besondere Bedeutung menschlicher Fähigkeiten in ihrer Entwicklung schärft, zeigt sie zugleich viele Eingreifpunkte auf, um Tech-Unternehmen nachhaltig im Sinne der Beschäftigten zu gestalten.

In künftigen Forschungsarbeiten gilt es sowohl die Analyse der neuen IT-basierten Maschinensysteme zu vertiefen und in ihrer weiteren Entwicklung zu verfolgen als auch aufbauend auf die vorgelegte Basisdefinition Schritt für Schritt weitere Dimensionen der Fähigkeiten in den Fokus zu rücken, welche für die Umsetzung der Strategie „Tech-Unternehmen“ entwickelt

³⁰ Wie Penenberg (2010, 143ff) schildert, brach z.B. bei Friendster in der entscheidenden Wachstumsphase die IT-Infrastruktur regelmäßig über längere Zeiträume zusammen. McCullough (2018, 261) resümiert: „The engineering challenges of delivering what was quickly becoming a deluge of content were at a whole new scale, and Friendster simply wasn't up to the challenge.“

wurden. In diesem Aufsatz lediglich angedeutet werden konnte zum Beispiel, dass die Realisation der Potenziale dynamisch skalierbarer Internetanwendungen komplementär die Entwicklung neuer Konzepte wie DevOps für die Organisation der Arbeit, die Gestaltung der Arbeitskultur und den Aufbau von Qualifikationen insbesondere in den Entwicklungsorganisationen bedingte. Doch was genau sich hinter der Kategorie DevOps verbirgt, wie DevOps-Praktiken umgesetzt werden, worauf es dabei ankommt, wie sich DevOps auf die Arbeitssituation von Tech-Arbeiter:innen auswirkt und von ihnen erlebt wird, ist künftig eingehender zu untersuchen. Aber auch für Felder wie die Managementpraktiken, die Gestaltung der unternehmensinternen Organisations- und Governance-Strukturen oder die Entwicklung der unternehmensübergreifenden Zusammenarbeit bis hin zur Strategiebildung selbst stellt sich die Frage, inwiefern in den Tech-Unternehmen in diesen Dimensionen spezifische Fähigkeiten aufgebaut wurden und wie diese in der Praxis umgesetzt werden.

Gleichzeitig ist keineswegs von der Hand zu weisen, dass nicht alle Tech-Unternehmen gleich sind. Von den Tech-Konzernen, welche mit Plattformanwendungen in bestimmten Märkten dominante Stellungen eingenommen haben, über fortlaufend neu entstehende Start-ups oder Tech-Unternehmen im industriellen Umfeld wie Tesla bis hin zu etablierten Unternehmen, welche sich in Tech-Unternehmen transformieren, sind mittlerweile unterschiedliche Ausprägungen von Tech-Unternehmen entstanden. Während sie als Verwertungsstrategien allesamt um Internetanwendungen zentrieren, folgen sie in ihrem „Autonomiebestreben“ (Altmann/Bechtle 1971, 26) heterogenen Entwicklungspfaden und verfügen in der Strategiebildung über unterschiedliche Handlungsspielräume. Wie lassen sich diese Ausprägungen gruppieren? Welche Unterschiede gibt es? Welche Gemeinsamkeiten bestehen und in welche Richtung entwickeln sie sich in Zukunft? Wenngleich mit diesem Aufsatz ein erster Schritt auf dem Weg zur Ausarbeitung eines soziologischen Begriffs des Tech-Unternehmens als neuen Unternehmertypus gegangen werden konnte, bleiben somit weiter viele Fragen offen und harren der Bearbeitung.

5. Literatur

- Altmann, Norbert/Bechtle, Günter (1971): Betriebliche Herrschaftsstruktur und industrielle Gesellschaft. Ein Ansatz zur Analyse. München: Hanser
- Armbrust, Michael/Fox, Armando/Griffith, Rean/Joseph, Anthony D./Katz, Randy H./Konwinski, Andrew/Lee, Gunho/Patterson, David A./Rabkin, Ariel/Stoica, Ion/Zaharia, Matei (2009): Above the Clouds. A Berkeley View of Cloud Computing. <https://www2.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf>. Zugriff am 12. Mai 2021
- Barr, Jeff (2009): New Features for Amazon EC2. Elastic Load Balancing, Auto Scaling, and Amazon Cloud Watch. <https://aws.amazon.com/de/blogs/aws/new-aws-load-balancing-automatic-scaling-and-cloud-monitoring-services/>. Zugriff am 12. Mai 2021
- Barroso, Luiz A./Dean, Jeffrey/Hölzle, Urs (2003): Web Search for a Planet. The Google Cluster Architecture. In: IEEE Micro, 23, 2, 2-28
- Barroso, Luiz A./Clidaras, Jimmy/Hölzle, Urs (2013): The Datacenter as a Computer. An Introduction to the Design of Warehouse-Scale Machines. 2. Aufl. San Rafael [CA]: Morgan & Claypool
- Beyer, Betsy/Jones, Chris/Petoff, Jennifer/Murphy, Niall R. (2016): Site Reliability Engineering. How Google Runs Production Systems. Sebastopol [CA]: O'Reilly
- Blank, Marco/Held, Maximilian/Nicklich, Manuel/Pfeiffer, Sabine/Sauer, Stefan/Tihlarik, Amelie (2020): Mindset – Begriffskarriere zwischen Management-Talk, Wissenschaft und Ideologie. Eine explorative Annäherung. <https://laboratory.de/downloads/WP-01-2020-Mindset.pdf>. Zugriff am 29. März 2021
- Boes, Andreas/Kämpf, Tobias/Langes, Barbara/Lühr, Thomas (2015): Landnahme im Informationsraum. Neukonstituierung gesellschaftlicher Arbeit in der digitalen Gesellschaft. In: WSI-Mitteilungen, 67, 2, 77-85
- Boes, Andreas/Gül, Katrin/Kämpf, Tobias/Langes, Barbara/Lühr, Thomas/Marrs, Kira/Vogl, Elisabeth/Ziegler, Alexander (2018): Silicon Valley. Vorreiter im digitalen Umbruch. Folgen für Deutschland und Europa. Forschungsreport. München: ISF München
- Boes, Andreas/Langes, Barbara/Vogl, Elisabeth (2019): Die Cloud als Wegbereiter des Paradigmenwechsels zur Informationsökonomie. In: Boes, Andreas/Langes, Barbara (Hrsg.): Die Cloud und der digitale Umbruch in Wirtschaft und Arbeit. Freiburg u.a.: Haufe, 115-144
- Böhle, Fritz/Milkau, Brigitte (1988): Sinnliche Erfahrung und Erfahrungswissen im industriellen Arbeitsprozess. München.
- Böhle, Fritz (2018): Arbeit als Handeln. In: Böhle, Fritz/Voß, G. Günter/Wachtler, Günther (Hrsg.): Handbuch Arbeitssoziologie. Band 1: Arbeit, Strukturen und Prozesse, 2. Auflage. Wiesbaden: Springer, 171-200
- Borras, Michael/Zysman, John (1997): Globalization with borders. The Rise of Wintelism as the Future of Global Competition. In: Industry and Innovation, 4, 2, 141-66
- Bukowski, Ed/Moyles, Brian/McGarr, Mike (2016): How we build Code at Netflix. <https://netflixtechblog.com/how-we-build-code-at-netflix-c5d9bd727f15>. Zugriff am 14. Mai 2021
- Burns, Brendan (2018): Designing Distributed Systems. Patterns and Paradigms for Scalable, Reliable Services. Sebastopol [CA]: O'Reilly
- Busse, Caspar/Freiberger, Harald (2020): Absturz eines Börsenlieblings. <https://www.sueddeutsche.de/wirtschaft/sap-aktie-klein-1.5094517>. Zugriff am 29. März 2021
- Butollo, Florian/de Paiva Lareiro, Patricia (2020): Digitale Revolution? Widersprüche der Produktivkraftentwicklung im Postwachstumskapitalismus. In: Das Argument 335, 82-102
- Dash, Anil (2016): There is no „technology industry“. <https://medium.com/humane-tech/there-is-no-technology-industry-44774dfb3ed7>. Zugriff am 29. März 2021
- Dean, Jeffrey/Barroso, Luiz A. (2013): The Tail at Scale. In: Communications of the ACM, 56, 2, 74-80
- DeCandia, Giuseppe/Hastorun, Deniz/Jampani, Madan/Kakulapati, Gunavardhan/Lakshman, Avinash/Pilchin, Alex/Sivasubramanian, Swaminathan/Vosshall, Peter/Vogels, Werner (2007): Dynamo. Amazon's Highly Available Key-value Store. In: ACM SIGOPS Operating Systems Review, 41, 6, 205-220
- Dolata, Ulrich (2015): Volatile Monopole. Konzentration, Konkurrenz und Innovationsstrategien der Internetkonzerne. In: Berliner Journal für Soziologie, 24, 4, 505-529

- Evans, Benedict (2019): Netflix is not a Tech Company. <https://www.ben-evans.com/benedictevans/2019/7/31/Netflix>. Zugriff am 29. März 2021
- Farrow, Rik (2012): Netflix Heads into the Clouds. Interview with Adrian Cockroft. In: ;login:, 37, 1, 44-46
- Feitelson, Dror G./Frachtenberg, Eitan/Beck, Kent L. (2013): Development and Deployment at Facebook. In: IEEE Computing, 17, 4, 8-17
- Florence, Ken (2016): How Netflix Works with ISPs around the Globe to Deliver a Great Viewing Experience. <https://about.netflix.com/en/news/how-netflix-works-with-isps-around-the-globe-to-deliver-a-great-viewing-experience>. Zugriff am 29. März 2021
- Foley, John (2009): Chief of the Year. Amazon's Werner Vogels. https://dsimg.ubm-us.net/envelope/259152/465273/InformationWeek_Analytics_Alerts_chiefoftheyear.pdf. Zugriff am 29.03.2021
- Fulton, Scott (2015): What Let Amazon to its Own Microservices Architecture, <https://thenewstack.io/led-amazon-microservices-architecture/>. Zugriff am 12. Mai 2021
- Galloway, Scott (2018): The Four. The Hidden DNA of Amazon, Apple, Facebook, and Google. New York: Portfolio Penguin
- Ghemawat, Sanjay/Gobioff, Howard/Leung, Shun-Tak (2003): The Google File System. In: ACM SIGOPS Operating Systems Review, 37, 5, 29-43
- Google Press Center (2006): Search Engine Strategies Conference. Conversation with Eric Schmidt hosted by Danny Sullivan. <https://www.google.com/press/podium/ses2006.html>. Zugriff am 29.03.2021
- Gray, Jim (2006): A Conversation with Werner Vogels, in: ACM Queue, 4. Jg., H. 4, 14-22
- Greenpeace (2017): Clicking Clean. Who is Winning The Race to Build a Green Internet?. https://www.greenpeace.de/sites/www.greenpeace.de/files/publications/20170110_greenpeace_clicking_clean.pdf. Zugriff am 29. März 2021
- Guzzetta, Marli (2016): Why even a salad chain wants to call itself a tech company. The case for why you should or shouldn't call yourself a tech company. <https://www.inc.com/magazine/201605/marli-guzzetta/tech-company-definition.html>. Zugriff am 29. März 2021
- Haug, Wolfgang Fritz (2003): High-Tech-Kapitalismus. Analysen zu Produktionsweise, Arbeit, Sexualität, Krieg und Hegemonie. Hamburg: Argument
- Haug, Wolfgang Fritz (2020): Staabs ‚Digitaler Kapitalismus‘. Eine forschende Auseinandersetzung. In: Das Argument 335, 19-56
- Heath, Catherine (2017): What is a ‚tech company‘ anyway?. <https://technation.io/news/tech-company-definition/>. Zugriff am 29. März 2021
- Heim, Tino (2013): Metamorphosen des Kapitals. Kapitalistische Vergesellschaftung und Perspektiven einer kritischen Sozialwissenschaft nach Marx, Foucault und Bourdieu. Bielefeld: Transcript
- Helland, Pat (2016): Standing on Distributed Shoulders of Giants. In: Communications of the ACM, 59. Jg., H. 6, 58-61
- Hendel, Anja (2020ff): How to Build a Tech Company. <https://medium.com/how-to-build-a-tech-company>. Zugriff am 29. März 2021
- Highfield, Tim (2016): Social Media and Everyday Politics. Cambridge et al.: Polity
- Huckman, Robert S./Pisano, Gary P./Kind, Liz (2012): Amazon Web Services. Boston: HBS Case Study
- Isaac, Mike/Frenkel, Sheera (2020): Facebook Is ‚Just Trying to Keep the Lights On‘ as Traffic Soars in Pandemic. <https://www.nytimes.com/2020/03/24/technology/virus-facebook-usage-traffic.html>. Zugriff am 12. Mai 2021
- Izrailevsky, Yuri/Tseitlin, Ariel (2011): The Netflix Simian Army. <https://netflixtechblog.com/the-netflix-simian-army-16e57fbab116>. Zugriff am 12. Mai 2021
- Jamshidi, Pooyan/Pohl, Claus/Mendonça, Nabor C./Lewis, James/Tilkov, Stefan (2018): Microservices. The Journey So Far and Challenges Ahead. In: IEEE Software, 35, 3, 24-35
- Kantor, Jodi/Streitfeld, David (2015): Inside Amazon: Wrestling Big Ideas in a Bruising Workplace, <https://www.nytimes.com/2015/08/16/technology/inside-amazon-wrestling-big-ideas-in-a-bruising-workplace.html>. Zugriff am 29. März 2021

- Kenney, Martin (Hrsg.) (2000): Understanding Silicon Valley. The Anatomy of an Entrepreneurial Region. Stanford [CA]: Stanford Business Books
- Kersten, Mik (2018): Project to product. How to survive and thrive in the age of digital disruption with the flow framework. Portland: IT Revolution
- Killalea, Tom (2008): Meet the virts. Virtualization technology isn't new, but it has matured a lot over the past 30 years. In: ACM Queue, 6, 1, 14-18
- Killalea, Tom (2020): A Second Conversation with Werner Vogels. In: ACM Queue, 18, 5, 67-92
- Kim, Gene/Humble, Jez/Debois, Patrick/Willis, John (2016): The DevOps Handbook. How to Create World-Class Agility, Reliability & Security in Technology Organizations. Portland: IT Revolution
- Kim, Gene (2018): The Unicorn Project. A Novel About Developers, Digital Disruption and Thriving in the Age of Data. Portland: IT Revolution (zit. n. Kindle Ausgabe)
- Kim, Gene (2020): The Idealcast. Episode 8: Architecture as the Organizing Logic for Components, and the Means for their Construction. Guest Michael Nygard, <https://itrevolution.podbean.com/e/architecture-as-the-organizing-logic-for-components-and-the-means-for-their-construction/>. Zugriff am 29.03.2021
- Kirchner, Stefan/Beyer, Jürgen (2016): Die Plattformlogik als digitale Marktordnung. In: Zeitschrift für Soziologie, 45, 5, 324–339
- Kushida, Kenji/Murray, Jonathan/Zysman, John (2011): Diffusing the Cloud. Cloud Computing and Implications for Public Policy. In: Journal of Industry, Competition and Trade, 11, 3, 209-237
- Kushida, Kenji/Murray, Jonathan/Zysman, John (2015): Cloud Computing. From Scarcity to Abundance. In: Journal of Industry, Competition and Trade, 15, 1, 5-19
- Langes Barbara/Vogl, Elisabeth (2019): Arbeit in der Informationsökonomie. In: Boes, Andreas/Langes, Barbara (Hrsg.): Die Cloud und der digitale Umbruch in Wirtschaft und Arbeit. Freiburg u.a.: Haufe, 147-172
- Lawson, Jeff (2021): Ask your developer. How to Harness the Power of Software Developers and Win in the 21st Century. New York: Harper Business
- Levy, Steven (2011): In the Plex. How Google Thinks, Works and Shapes Our Lives. New York et al.: Simon & Schuster
- Lewis, James/Fowler, Martin (2014): Microservices. <https://martinfowler.com/articles/microservices.html>. Zugriff am 12. Mai 2021
- Lüthje, Boy (2001): Standort Silicon Valley. Ökonomie und Politik der vernetzten Massenproduktion. Frankfurt/M, New York: Campus
- McCullough, Brian (2018): How the Internet Happened. From Netscape to the iPhone, New York et al.: W. W. Norton
- McMillan, Robert/Metz, Cade (2012): How Amazon Followed Google into the World of Secret Servers, 11.03.2012, <https://www.wired.com/2012/11/amazon-google-secret-servers/>. Zugriff am 12. Mai. 2021
- MEW 3 = Marx, Karl (1978): Thesen über Feuerbach. In: MEW, Band 3, Berlin: Dietz
- MEW 23 = Marx, Karl (1962): Das Kapital. Kritik der politischen Ökonomie. In: MEW, Band 23, Berlin: Dietz
- MEW 25 = Marx, Karl (1964): Das Kapital. Kritik der politischen Ökonomie. In: MEW, Band 25, Berlin: Dietz
- Mills, Mark P. (2020): Digital Cathedrals, New York: Encounter Books
- Mims, Christopher (2018): Every company is now a tech company. <https://www.wsj.com/articles/every-company-is-now-a-tech-company-1543901207>. Zugriff am 29. März 2021
- Nachtwey, Oliver/Staab, Philipp (2015): Die Avantgarde des digitalen Kapitalismus. In: Mittelweg36. Zeitschrift des Hamburger Instituts für Sozialforschung, 14, 6, 59-84
- Nachtwey, Oliver/Staab, Philipp (2016): Die Digitalisierung der Dienstleistungsarbeit. In: APuZ, 18-19, 24-31
- Nachtwey, Oliver/Staab, Philipp (2020): Das Produktionsmodell des digitalen Kapitalismus. In: Maasen, Sabine/Passoth, Jan-Hendrik (Hrsg.): Soziologie des Digitalen – Digitale Soziologie? Soziale Welt, Sonderband23. Baden-Baden, 285-304
- Newman, Sam (2015): Building Microservices. Designing Fine Grained Systems. Sebastopol [CA]: O'Reilly
- Nies, Sarah/Sauer, Dieter (2010): Theoriegeleitete Fallstudienforschung. Forschungsstrategien am ISF München. In: Pongratz, Hans J./Trinczek, Rainer (Hrsg.): Industriesoziologische Fallstudien. Entwicklungspotenziale einer Forschungsstrategie. Berlin, 119–162

- O'Grady, Stephen (2014): The Scale Imperative. <https://redmonk.com/sogrady/2014/12/01/the-scale-imperative/>. Zugriff am 12. Mai 2021
- O'Grady, Stephen (2015): The Software Paradox. Sebastopol [CA]: O'Reilly
- O'Mara, Margaret (2019): The Code. Silicon Valley and the Remaking of America. New York: Penguin Press
- Ozzie, Ray (2005): The Internet Services Disruption, <https://www.cnet.com/news/ozzie-memo-internet-services-disruption/>. Zugriff am 12. Mai 2021
- Payne, Alex (2012): What is and is not a technology company. <https://www.al3x.net/blog/2012/05/08/what-is-and-is-not-a-technology-company>. Zugriff am 29. März 2021
- Penenberg, Adam L. (2010): Viral Loop. From Facebook to Twitter, how Today's Smartest Businesses Grow Themselves. New York: Tantor Media
- Pfeiffer, Sabine (2019): Digitale Transformation: Great, greater, tilt ...? Von der Produktivkraft- zur Distributivkraftentwicklung. In: Dörre, Klaus/Rosa, Hartmut/Becker, Karina/Bose, Sophie/Seyd, Benjamin (Hrsg.): Große Transformation? Zur Zukunft moderner Gesellschaften. Wiesbaden: Springer, 383-399
- Piétron, Dominik/Staab, Philipp (2021): EU gegen Big Tech: Das Ende der Gesetzlosigkeit?. In: Blätter für deutsche und internationale Politik, 2, 95-101
- Popper, Nathaniel/Siegel Bernhard, Tara (2020): High-Flying Trading App Robinhood Goes Down at the Wrong Time. <https://www.nytimes.com/2020/03/03/technology/trading-app-robinhood-outage.html>. Zugriff am 29. März 2021
- Regalado, Antonio (2011): Who Coined Cloud Computing? <https://www.technologyreview.com/2011/10/31/257406/who-coined-cloud-computing/>. Zugriff am 14. Mai 2021
- Ries, Eric (2017): The Start-up Way. How Entrepreneurial Management Transforms Culture and Drives Growth. New York: Portfolio Penguin
- Satariano, Adam (2019): How the Internet Travels across Oceans. <https://www.sfgate.com/business/article/How-the-internet-travels-across-oceans-13704126.php>. Zugriff am 12. Mai 2021
- Schrage, Jan-Felix (2015): Open Source Softwareprojekte zwischen Passion und Kalkül. SOI Diskussionspapier, Nr. 02, Stuttgart
- Schonfeld, Erick (2008): Amazon Web Services Goes Down, Takes Many Startup Sites with it. <https://techcrunch.com/2008/02/15/amazon-web-services-goes-down-takes-many-startup-sites-with-it/>. Zugriff am 12. Mai 2021
- Schwan, Ben (2019): iCloud: Apple versucht, auf eigene Rechenzentren umzuschwenken. <https://www.heise.de/mac-and-i/meldung/iCloud-Apple-versucht-auf-eigene-Rechenzentren-umzuschwenken-4408993.html>. Zugriff am 29. März 2021
- Shahin, Mojtaba/Babar, Muhammad Ali/Zhu, Liming (2017): Continuous Integration, Delivery and Deployment. A Systematic Review on Approaches, Tools, Challenges and Practices. In: IEEE Access, 5. Jg., 3909-3943
- Srnicek, Nick (2018): Plattform-Kapitalismus, Hamburg: Hamburger Edition
- Staab, Philipp (2016): Falsche Versprechen. Wachstum im digitalen Kapitalismus. Hamburg: Hamburger Edition
- Staab, Philipp (2019): Digitaler Kapitalismus. Markt und Herrschaft in der Ökonomie der Unknappheit, Frankfurt/M: Suhrkamp
- Stone, Brad (2013): The Everything Store. Jeff Bezos and the Age of Amazon. New York: Back Bay Books
- Stringfellow, West (2018): Building Product Teams. Examples from Amazon, Google, Apple, Basecamp and Fogcreek, <https://laptrinhx.com/building-product-teams-examples-from-amazon-google-apple-basecamp-and-fog-creek-951573196/>. Zugriff am 14.05.21
- Tarnoff, Ben/Weigel, Moira (2020): Voices from the Valley. Tech Workers Talk about What They Do and how They Do It. New York: Farrar Strauss & Giroux
- Teece, David, J./Pisano, Gary P./Shuen, Amy (1997): Dynamic Capabilities and Strategic Management. In: Strategic Management Journal, 18, 7, 509-533
- Varley, Ian (2016): Salesforce Is All about Data, <https://engineering.salesforce.com/salesforce-is-all-about-data-d6fdf57cc8eb>. Zugriff am 12. Mai 2021
- Vise, David A./Malseed, Mark (2005): The Google Story. New York: Bantam Dell

- Vogels, Werner (2006): A Word on Scalability, <https://engineering.salesforce.com/salesforce-is-all-about-data-d6fdf57cc8eb>. Zugriff am 12. Mai 2021
- Vogels, Werner (2009): Eventually consistent. In: Communications of the ACM, 52, 1, 40-44
- Vogels, Werner (2019): Modern Applications at AWS, <https://www.allthingsdistributed.com/2019/08/modern-applications-at-aws.html>. Zugriff am 12. Mai 2021
- Vogl, Elisabeth (2017): Netflix. Vom online-Filmverleiher zum Streaminggiganten in der Cloud, 07.03.2017, https://idguzda.de/wp-content/uploads/2020/05/2017-03_final.pdf. Zugriff am 12. Mai 2021
- Vogl, Elisabeth (2020): Open Source-Entwicklung als Quelle von Empowerment?, in: Boes, Andreas/Gül, Katrin/Kämpf, Tobias/Lühr, Thomas (Hg.): Empowerment in der agilen Arbeitswelt. Analysen, Handlungsorientierungen und Erfolgsfaktoren für eine neue Humanisierung. Freiburg u.a.: Haufe, 93-109
- Winters, Titus/Manshreck, Tom/Wright, Hyrum (2020): Software Engineering at Google. Lessons Learned from Programming over Time. Sebastopol [CA]: O'Reilly
- Yegge, Steve (2011): Stevey 's Google Platform Rant, <https://gist.github.com/chitchcock/1281611>. Zugriff am 12. Mai 2021
- Ziegler, Alexander (2020): Der Aufstieg des Internet der Dinge. Wie sich Industrieunternehmen zu Tech-Unternehmen entwickeln. Frankfurt/M, New York: Campus
- Zuboff, Soshana (2018): Das Zeitalter des Überwachungskapitalismus. Frankfurt/M, New York: Campus