

Technische Universität Berlin

Fakultät 6, Institut für Soziologie

Fachgebiet Soziologie technikwissenschaftlicher Richtung



Masterarbeit

Kontrolle ohne Korrektiv

KI-gestütztes stellvertretendes Management als geschützter Raum in der
Softwareentwicklung

Erstgutachter: Prof. Dr. Ingo Schulz-Schaeffer

Zweitgutachter: Prof. Dr. Simon Schaupp

Abgabedatum: 21.01.2026

Vorgelegt von: Marlene Kulla

Matrikelnummer: 494179

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne Hilfe Dritter und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe. Alle Stellen, die den benutzten Quellen und Hilfsmitteln wörtlich oder inhaltlich entnommen sind, habe ich als solche kenntlich gemacht.

Sofern KI-Tools verwendet wurden, habe ich Produktnamen, Hersteller, die jeweils verwendete Softwareversion und die Art der Nutzung (z.B. sprachliche Überprüfung und Verbesserung der Texte, systematische Recherche) benannt. Ich verantworte die Auswahl, die Übernahme und sämtliche Ergebnisse des von mir verwendeten KI-generierten Outputs vollumfänglich selbst.

Die Satzung zur Sicherung guter wissenschaftlicher Praxis an der TU Berlin vom 15. Februar 2023. https://www.static.tu.berlin/fileadmin/www/10002457/K3-AMBL/Amtsblatt_2023/Amtliches_Mitteilungsblatt_Nr._16_vom_30.05.2023.pdf habe ich zur Kenntnis genommen.

Ich erkläre weiterhin, dass ich die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt habe.

Berlin, den 21.01.2026,

.....

Einsatz von KI-Software (gemäß Eigenständigkeitserklärung)

Im Rahmen der Erstellung dieser Arbeit wurde folgende KI-Software verwendet:

- ChatGPT vom Hersteller Open AI in den Versionen GPT-4 bis GPT-5.2
- Claude vom Hersteller Anthropic in der Version Claude Sonnet 4.5

Art der Nutzung:

- Ergänzende Recherche möglicher theoretischer Anschlussstellen
- Entwicklung von Gliederungs- und Strukturierungsvorschlägen
- Sprachliche Verbesserung einzelner Formulierungen

Ich verantworte die Auswahl, die Übernahme und sämtliche Ergebnisse der von mir verwendeten KI-generierten Outputs vollumfänglich selbst.

Berlin, den 21.01.2026,

.....

Abstact (Deutsch)

KI-gestützte Software ist in der Arbeitspraxis der Softwareentwicklung inzwischen ein fester Bestandteil. Auf Grundlage einer netnographischen Analyse von Beiträgen auf der Plattform Reddit untersucht diese Arbeit, wie Entwicklerinnen und Entwickler die Implementierung von KI-Systemen in der Softwareentwicklung beschreiben, bewerten und aushandeln. Die Ergebnisse zeigen, dass KI in einigen Fällen die Rolle eines stellvertretenden Managements einnimmt, was teilweise als entlastend und selbstermächtigend beschrieben wird. Gleichzeitig offenbart sich eine ambivalente Dynamik: Arbeitsprozesse werden zunehmend durch KI-Vorschläge vorstrukturiert, Lernprozesse verändern sich und Abhängigkeiten von der Technologie nehmen zu.

Abstract (English)

AI-supported software has become an integral part of everyday work practices in software development. Based on a netnographic analysis of contributions on the platform Reddit, this study examines how software developers describe, evaluate, and negotiate the implementation of AI systems in software development. The findings indicate that, in some cases, AI is used as a form of self-regulatory management, which is partly described as relieving and empowering. At the same time, an ambivalent dynamic becomes apparent: work processes are increasingly pre-structured by AI-generated suggestions, learning processes are changing, and dependencies on the technology are increasing.

Inhaltsverzeichnis

1	Einleitung	1
2	Technologie und soziale Realität	4
2.1	Sozio-technische Transformationen	4
2.2	Digitale Resonanzräume	6
2.3	Digitale Steuerungsmechanismen in der Arbeitswelt	9
2.4	Automatisierung und Augmentation	13
3	Arbeitsfeld Softwareentwicklung	15
3.1	Professionstheorie	15
3.2	Open-Source-Kultur als zentrale Wissensproduktion	17
3.3	Leitbilder und Rollenverständnisse in der Softwareentwicklung	21
3.4	Einsatz von KI in der Softwareentwicklung	22
4	Methodischer Zugang	26
5	Perspektiven auf Softwareentwicklung mit KI	33
5.1	Dimension Arbeit	34
5.1.1	Produktivitätsgewinne	34
5.1.2	Anforderungszuwachs	35
5.1.3	Zwischenfazit: Dimension Arbeit	38
5.2	Dimension Identität	39
5.2.1	Kompetenzentwicklung	39
5.2.2	Deprofessionalisierung	42
5.2.3	Statusbewahrung	44
5.2.4	Anpassungsdruck	46
5.2.5	Zwischenfazit: Dimension Identität	47
5.3	Dimension Rollenzuschreibung	48
5.3.1	KI als Werkzeug	48
5.3.2	KI als Assistenz	50
5.3.3	KI als Strukturierungsinstanz	51
5.3.4	Zwischenfazit: Dimension Rolle	55
6	Jenseits der Effizienz: Strukturierende Effekte von KI	56
6.1	Subjektive Autonomieerfahrung	57
6.2	Erosion von Handlungs- und Lernprozessen	61
6.3	Grenzarbeit im Kontext KI-gestützter Softwareentwicklung	64
7	Fazit	66
8	Literaturverzeichnis	69

1 Einleitung

Systeme, die auf Künstlicher Intelligenz (KI)¹ beruhen, sind mittlerweile ein fester Bestandteil des Arbeitsalltags in der Softwareentwicklung. Sie werden dort unter anderem bei der Codegenerierung, der Fehlersuche oder der technischen Dokumentation eingebunden, und greifen damit unmittelbar in zentrale Arbeitsprozesse ein. Mit dieser zunehmenden Integration von KI-Systemen verändern sich nicht nur die Arbeitsabläufe, sondern auch die Lernprozesse und die professionelle Selbstverortung innerhalb der Programmierpraxis.

In der noch jungen Forschungslandschaft zu Künstlicher Intelligenz in der Arbeitswelt dominiert eine betriebs- bzw. organisationszentrierte Perspektive (Braganza et al., 2021; Brynjolfsson et al., 2025; Dell’Acqua et al., 2023; Frey & Osborne, 2017; Noy & Zhang, 2023). Im Fokus stehen vor allem die Substitution oder Ergänzung menschlicher Arbeit sowie die damit verbundenen Potenziale zur Effizienz- und Produktivitätssteigerung. Deutlich weniger Aufmerksamkeit erfahren bislang Studien, welche die subjektiven Wahrnehmungen und Beschreibungen von Arbeitnehmerinnen und Arbeitnehmern zu ihrer Arbeit mit KI und deren Einfluss auf ihr professionelles Selbstverständnis in den Blick nehmen. Ziel der vorliegenden Arbeit ist es deshalb nicht die technische Leistungsfähigkeit generativer KI-Systeme, sondern die subjektiven Deutungen und Aushandlungsprozesse zu untersuchen.

Diese Perspektive ist deshalb von besonderer Bedeutung, weil technologische Veränderungen ihre gesellschaftliche Wirksamkeit nicht allein über ihre technische Leistungsfähigkeit entfalten, sondern maßgeblich über die Art und Weise, wie sie von den betroffenen Akteurinnen und Akteuren wahrgenommen, interpretiert und in bestehende Arbeitspraktiken integriert werden. Die subjektiven Deutungen von Arbeitnehmerinnen und Arbeitnehmern strukturieren, wie KI-Technologien genutzt, akzeptiert, abgelehnt oder umgedeutet werden, und beeinflussen damit konkret die

¹ Der Begriff Künstliche Intelligenz (KI) wird in dieser Arbeit bewusst verwendet, obwohl er aus technischer Sicht unpräzise ist. Systeme wie Large Language Models (LLMs) basieren auf statistischer Mustererkennung und simulieren Intelligenz, ohne über Bewusstsein, Intentionalität oder Verstehen zu verfügen (Abramson et al., 2021; Cuskley et al., 2024; Hoffmann, 2022; Korteling et al., 2021). Dennoch wird der Begriff beibehalten, weil er im gesellschaftlichen und professionellen Diskurs jene selbstlernende Technologien bezeichnet, die als „intelligent“ oder „autonom“ wahrgenommen werden. Die Arbeit versteht KI daher nicht als technische, sondern als sozialwissenschaftliche Kategorie welche Zuschreibungen von Rationalität, Kreativität und Entscheidungsfähigkeit auf algorithmische Systeme bündelt (Abramson et al., 2021; D. P. Atkinson & Barker, 2023; Elmaresa & Irwansyah, 2025).

Ausgestaltung von Arbeit, Rollenverständnissen und professionellen Identitäten. Zudem sind Fragen der Verantwortung, Autonomie und beruflichen Expertise eng mit individuellen und kollektiven Aushandlungsprozessen verknüpft, die sich nicht aus organisations- oder technologiezentrierten Analysen erschließen lassen. Eine Fokussierung auf subjektive Wahrnehmungen und Deutungsmuster ermöglicht es daher, die sozialen und normativen Dimensionen des KI-Einsatzes sichtbar zu machen und die Transformation von Arbeit nicht nur als ökonomischen oder technischen, sondern als sinn- und identitätsbezogenen Prozess zu begreifen. Die leitende Frage lautet daher: Wie wird der Einsatz von KI in der Softwareentwicklung in öffentlichen Diskursräumen beschrieben und normativ ausgehandelt?

Da es sich um ein noch junges, dynamisches und bislang wenig untersuchtes Forschungsfeld handelt, eignet sich ein explorativ-induktiver Forschungsansatz. Zur Analyse subjektiver Wahrnehmungen und Deutungen ist darüber hinaus ein qualitatives Forschungsdesign geeignet. Um subjektive Erfahrungs- und Sinnwelten zu interpretieren, eignen sich insbesondere ethnographische Ansätze als Erhebungs- und Analysemethoden. Die Ethnographie zielt in ihrem Kern darauf ab, soziale Praktiken, Wissensbestände und Sinnzusammenhänge in ihrer Prozesshaftigkeit zu rekonstruieren und eröffnet damit die Möglichkeit, nicht nur Handlungen, sondern auch deren kulturelle und soziale Einbettungen zu erfassen (P. Atkinson et al., 2001; Knoblauch, 2014).

Da ein großer Teil des Austauschs von Softwareentwicklerinnen und -entwicklern im Umgang mit generativer KI außerhalb formaler Arbeitskontexte stattfindet, richtet die Arbeit den Blick dabei auf digitale Diskussionsräume, in denen Entwicklerinnen und Entwickler Erfahrungen, Routinen und Unsicherheiten offen thematisieren. Methodisch folgt die Untersuchung dem Ansatz der *Netnographie* nach Kozinets (2015, 2019), einer Weiterentwicklung ethnographischer Forschung für den digitalen Raum. Im Rahmen der Netnographie werden Interaktionen in sozialen Netzwerken beobachtet und systematisch ausgewertet. Da Nutzerinnen und Nutzer in anonymen Online-Umgebungen häufig offener und reflektierter über ihre Erfahrungen, Einschätzungen und Unsicherheiten sprechen (Lapidot-Lefler & Barak, 2012; Suler, 2004), eignet sich dieser Ansatz besonders für die Untersuchung der Selbstwahrnehmung im professionellen Kontext. Die Verfasserinnen und Verfasser der Reddit-Posts werden in dieser Arbeit als „User“ bezeichnet, da dies eine geschlechtsneutrale und zugleich pseudonyme Bezeichnung für die anonymen Diskussionsteilnehmenden darstellt.

Der theoretische Rahmen dieser Arbeit verbindet Ansätze der Science and Technology Studies (STS) mit der soziologischen Arbeits- und Professionsforschung. Grundlegend ist die Annahme, dass technologische Innovationen die Arbeitswelt nicht nur rationalisieren, sondern diese sozio-

technisch transformieren (Leonardi & Barley, 2011; Lühr et al., 2020; Orlikowski, 2007; Spicer, 2003). Technologien sind demnach keine neutralen Werkzeuge, sondern aktive Elemente der Arbeitsorganisation, die Handlungsspielräume, Wissensordnungen und Machtbeziehungen verändern. Gleichzeitig richtet die Arbeit den Blick auf die Handlungs- und Deutungsperspektiven der Akteurinnen und Akteure. Anknüpfend an professionssoziologische Ansätze werden Prozesse der Aushandlung professioneller Zuständigkeiten, Kompetenzen und Identitäten in den Fokus gerückt. Hierfür wird das Konzept der Grenzarbeit („Boundary Work“) herangezogen, das beschreibt, wie Akteure Grenzen zwischen Tätigkeitsfeldern, Wissensdomänen und Verantwortlichkeiten ziehen, verschieben oder neu konfigurieren (Faulconbridge et al., 2023; Gieryn, 1983; Langley et al., 2019). Ergänzend wird die historisch gewachsene Rolle der Softwareentwicklung berücksichtigt, um etablierte Leitbilder, Arbeitspraktiken und professionelle Sinnwelten zu erfassen, vor deren Hintergrund der Einsatz generativer KI gedeutet und bewertet wird.

Der Theorieteil dieser Arbeit setzt sich aus zwei Kapiteln zusammen: Kapitel 2 erläutert zunächst, wie Technologien als soziotechnische Systeme zur Hervorbringung und Strukturierung sozialer Realität beitragen. Diese soziotechnischen Transformationen werden anschließend anhand von zwei verschiedenen Bereichen konkretisiert. Zunächst wird darauf eingegangen, wie KI als digitaler Resonanzraum fungieren kann und dadurch eine attraktive Nutzungspraxis zur Bestätigung und Spiegelung von bestehenden Deutungsmustern bietet. Anschließend werden in zwei weiteren Abschnitten die Auswirkungen des Einsatzes von KI auf die Arbeitswelt thematisiert. Kapitel 3 widmet sich dann konkret dem Forschungsstand zu Arbeitspraktiken und professioneller Identität in der Softwareentwicklung und verortet die Arbeit damit im bestehenden wissenschaftlichen Diskurs. Darauf folgt im vierten Kapitel, die Darstellung des methodischen Zugangs der Untersuchung. Kapitel 5 präsentiert die empirischen Befunde der Studie. Diese lassen sich in drei zentrale Dimensionen gliedern: erstens die Dimension Arbeit, in der wahrgenommene Veränderungen der Arbeitspraxis thematisiert werden, zweitens die Dimension Identität, die sich mit beschriebenen Veränderungen des professionellen Selbstbildes befasst, sowie drittens die Dimension Rolle, in der insbesondere die subjektiven Deutungen der Mensch-Maschine-Interaktion im Mittelpunkt stehen. In Kapitel 6 werden die verschiedenen empirischen Befunde zusammengeführt und in den theoretischen Diskurs zur Nutzung von KI in der Softwareentwicklung eingebettet, um deren Einfluss auf Arbeit und professionelle Praxis zu analysieren. Abschließend werden in Kapitel 7, dem Fazit die zentralen Erkenntnisse der Arbeit zusammengefasst und deren Implikationen für zukünftige Forschung sowie für die Praxis diskutiert.

2 Technologie und soziale Realität

Dieses Kapitel legt die theoretischen Grundlagen, um die Frage nach den wahrgenommenen Auswirkungen der Nutzung Künstlicher Intelligenz auf die Arbeit zu beantworten. Kern dieses Kapitels ist es, das Zusammenspiel von Technologie und Arbeitswelt nicht als eindimensionalen Einflussprozess, sondern als wechselseitige Beziehung zu verstehen. Dazu werden zentrale Einflussparameter herausgearbeitet, die diese Beziehung strukturieren und für die Analyse der empirischen Befunde von Bedeutung sind.

Im ersten Abschnitt wird ein theoretischer Rahmen vorgestellt, der Arbeit und Technologie als miteinander verflochtene soziotechnische Prozesse begreift. Im zweiten Unterkapitel wird erläutert, wie digitale Resonanzräume entstehen. Dabei wird gezeigt, dass die Architektur von KI-basierten Dialogsystemen, ähnlich wie Algorithmen sozialer Medien, so funktioniert, dass sie tendenziell die bestehenden Einstellungen und Meinungen der Gesprächspartner reproduzieren. Dieses Verständnis ist wichtig für die spätere Analyse der Mensch-Maschine-Interaktion in der Softwareentwicklung, da es erklärt, wie Personen mit sprachbasierten Assistenzsystemen interagieren. Das dritte und das vierte Unterkapitel greifen jeweils zentrale Erkenntnisse aus Industrie- und Arbeitssoziologie auf, die notwendig sind, um die empirischen Befunde zur Nutzung von KI in bestehende und vergangene Diskurse der Digitalisierung von Arbeit einzuordnen. Das dritte Unterkapitel befasst sich mit digitalen Steuerungssystemen und deren Auswirkungen auf Arbeitsprozesse. Der letzte Abschnitt widmet sich dann den Debatten um technologiegetriebene Automatisierung und Augmentierung von Arbeit und liefert damit einen zentralen Bezugsrahmen für die Analyse von Veränderungen der Arbeitswelt durch KI.

2.1 Sozio-technische Transformationen

Die Perspektive der Science and Technology Studies (STS) geht grundlegend davon aus, dass Technologien nicht isoliert entstehen oder wirken, sondern immer in soziale Prozesse eingebettet sind. Diese Forschungsrichtung bietet damit eine theoretische Basis, um Technologien als materielle Phänomene zu begreifen, die gesellschaftliche Strukturen prägen und zugleich von ihnen geprägt werden. Innerhalb der STS haben sich verschiedene theoretische Ansätze herausgebildet, die jeweils eigene Perspektiven auf die Handlungsträgerschaft technischer Artefakte eröffnen.

Ein zentraler Ansatz ist die von Pinch und Bijker (1984) entwickelte *Social Construction of Technology* (SCOT). Diese Theorie versteht technische Artefakte als Ergebnis sozialer

Aushandlungsprozesse, in denen unterschiedliche gesellschaftliche Gruppen jeweils ihre eigenen Bedeutungen, Interessen und Interpretationen einbringen. Als zentrales Beispiel nennen die beiden Autoren die Evolution der Funktion und des Designs von Fahrrädern, wie wir sie heute kennen. Während frühe Hochräder von sportlich orientierten Nutzern als Ausdruck von Geschwindigkeit und Abenteuer geschätzt wurden, galten sie für andere Gruppen als unsicher. Erst durch die Aushandlung dieser unterschiedlichen Bedeutungen setzte sich das heute verbreitete Fahrrad durch (Pinch & Bijker, 1984). Die Perspektive der SCOT rückt mit diesem Blickwinkel die soziale Bedingtheit technologischer Entwicklung in den Vordergrund.

Neben dieser Annahme vertreten Akteure innerhalb der STS auch die umgekehrte Perspektive, nach der Technologien selbst aktiv zur Konstruktion sozialer Wirklichkeit beitragen. Ein technologisches Artefakt wird hier nicht lediglich als neutrales Werkzeug verstanden, sondern als wirksamer Bestandteil sozialer Praxis, der (soziales) Handeln strukturiert und lenkt. Technologien eröffnen spezifische Handlungsmöglichkeiten, die Hutchby (2001) als *Affordanzen* bezeichnet, während sie zugleich andere Handlungsoptionen begrenzen oder gar vollständig ausschließen. Indem technische Artefakte also allein durch ihre Architektur suggerieren, welches Verhalten naheliegend und möglich ist, wirken sie aktiv an der Strukturierung menschlichen Handelns mit, ohne dieses vollständig zu determinieren. So lädt beispielsweise ein Stuhl aufgrund seiner Konstruktion dazu ein, sich in einer bestimmten Art und Weise auf ihn zu setzen, während er andere Sitzvarianten unwahrscheinlicher, unattraktiver oder sogar unmöglich macht. Eine die sozialen Handlungen noch mehr einschränkende Technologie wird in Latours berühmtem Essay zum Berliner Schlüssel (Latour, 2021) gezeigt. Dieser ist so konstruiert, dass er nur dann aus dem Schloss gezogen werden kann, wenn die Tür wieder verschlossen ist, wodurch eine die Tür gerade öffnende Person technisch dazu gezwungen wird, eine soziale Regel wie das Schließen der Haustür einzuhalten. Anstelle sozialer Appelle oder Kontrolle wird das gewünschte Verhalten direkt in das Artefakt eingeschrieben und alternative Handlungsoptionen ausgeschlossen. Vergleichbare Mechanismen finden sich auch in Temposchwellen, in obdachlosenfeindlicher Architektur oder in digitalen Benutzeroberflächen sozialer Netzwerke, die durch ihre Gestaltung bestimmte Handlungen nahelegen und andere verhindern. In all diesen Fällen wirkt Technik nicht neutral, sondern strukturiert soziale Praxis.

Während SCOT-Ansätze die Technik vor allem als Produkt sozialer Aushandlungsprozesse beschreiben, benennen andere Ansätze, dass Technologien nicht nur Ergebnis, sondern selbst Bedingung sozialen Handelns sind. Technische Artefakte sind damit auch als politisch zu verstehen, da sie nicht nur in soziale Praktiken eingebettet sind, sondern aktiv zur Stabilisierung oder

Umgestaltung gesellschaftlicher Machtverhältnisse beitragen (Winner, 2007). Gesellschaft prägt technische Entwicklung und technische Artefakte gestalten gesellschaftliche Realität.

2.2 Digitale Resonanzräume

Vor diesem Hintergrund richtet der nun kommende Abschnitt den Blick auf digitale Resonanzräume. Dabei wird gezeigt, dass die Architektur KI-basierter Dialogsysteme, vergleichbar mit der Architektur sozialer Medien, dazu beiträgt, bestehende Einstellungen zu stabilisieren und zu reproduzieren. Diese Perspektive bildet eine zentrale Grundlage für die abschließende Analyse, da sie Erklärungsansätze dafür liefert, weshalb die Nutzung von KI für bestimmte Personengruppen als strukturierende Instanz besonders anschlussfähig erscheinen kann.

Ein den digitalen Mechanismen vorgestellter kognitiver Mechanismus der Meinungsstabilisierung ist die Tendenz von Individuen, sich bevorzugt mit Informationen auseinanderzusetzen, die die eigenen Überzeugungen bestätigen. Menschen neigen dazu, Nachrichten zu konsumieren, die ihre bestehenden Überzeugungen stützen, um „kognitive Dissonanz“ zu vermeiden (Festinger, 1962). Daraus resultiert ein sogenannter Bestätigungsfehler (*confirmation bias*), welcher als Konzept durch den Kognitionspsychologen Wason (1960, 1968) in den 1960er-Jahren geprägt wurde. In seinen Experimenten zur Hypothesentestung zeigte er, dass Menschen dazu neigen, Hypothesen durch Beispiele zu bestätigen, anstatt nach widerlegenden Informationen zu suchen (Wason, 1960, 1968). Spätere Arbeiten wie etwa der umfassende Übersichtsartikel von Raymond S. Nickerson (1998) machten deutlich, dass der Bestätigungsfehler in unterschiedlichsten Kontexten auftritt: in der Urteilsbildung, bei der Argumentation, im wissenschaftlichen Denken und eben auch bei der Mediennutzung.

Im sozialen Austausch wird diese Verzerrung häufig durch homogene Kommunikationsräume verstärkt. Der Begriff Echokammer beschreibt ein soziales oder medial-kommunikatives Umfeld, in dem Informationen, Meinungen und Überzeugungen wiederholt innerhalb einer ideologisch homogenen Gruppe zirkulieren, ohne dass abweichende Perspektiven in angemessenem Maße berücksichtigt oder zugelassen werden. Charakteristisch für solche Echokammern ist nicht nur die inhaltliche Gleichförmigkeit (Sunstein, 2002; Conover et al., 2011), sondern auch eine aktive Abschottung gegenüber widersprechenden Informationen, die häufig als unglaubwürdig oder feindlich eingestuft werden (Jamieson & Cappella, 2008). Ein vielzitiertes Experiment von Garrett (2009) zeigte, dass bei der Auswahl aus einer Vielzahl von Nachrichten, die unterschiedliche politische Meinungen repräsentieren, Menschen dazu neigen, sich eher mit Informationen zu beschäftigen, die ihre Meinung bestätigen. Auch wenn die Aversion zu denjenigen Nachrichten, die

nicht der eigenen politischen Meinung entsprechen in dem Experiment nur moderat war, kann die Kombination aus selektiver Informationssuche und Vermeidung gegensätzlicher Artikel zu der Entstehung einer ideologischen Echokammer beitragen (Garrett, 2009).

Digitale Medioumgebungen verstärken viele dieser Prozesse durch neue technische Selektionslogiken. Der Begriff *Filterblase* wurde durch den Internetaktivisten Pariser (2011) in seinem gleichnamigen Buch geprägt. Eine zentrale Rolle spielen dabei algorithmische Selektionssysteme, welche Medieninhalte automatisiert filtern und personalisieren, um eine Person möglichst lange auf der Plattform zu binden und ihre Interaktionen zu maximieren. Solche Systeme analysieren das Verhalten der User, etwa Klicks, Verweildauer, Likes oder Suchanfragen und generieren daraus individuell zugeschnittene Inhalte. Dadurch entsteht eine "Blase" individualisierter Information, in der gegensätzliche Perspektiven zunehmend ausgeblendet werden (Pariser, 2011). Tatsächlich lassen sich einzelne wenn auch nur schwache Effekte solcher Selektionsmechanismen in Experimenten auch empirisch nachweisen (Bakshy et al., 2015; Hannak et al., 2013; Luzsa & Mayr, 2021).²

Die Diskussion um Echokammern und kognitive Verzerrungen verlagert sich zunehmend auch auf den alltäglichen Umgang mit Künstlicher Intelligenz, insbesondere interaktiven Chatbots. So konnte in einer Untersuchung gezeigt werden, dass Personen in Interaktionen mit dialogorientierter KI eher bestätigende Informationen erhalten; insbesondere dann, wenn die Nutzereingabe bereits eine implizite Meinung oder ein Vorurteil enthält. Die Modelle generieren also tendenziell Ergebnisse, welche die formulierte Ausgangsthese verstärkten, anstatt alternative Perspektiven darzustellen (Sharma et al., 2024). Auch bei autonom agierenden KI-Agenten wurde solch ein "Echokammer-Effekt" beobachtet: In Simulationen entwickelten diese Agenten ohne externe Eingriffe zunehmend einseitige Meinungen, die sich gegenseitig verstärkten und polarisierende Dynamiken aufwiesen (Ohagi, 2024). Wenn Nutzerinnen und Nutzer mit einer bestimmten Überzeugung oder Prämisse in das Gespräch mit einer KI treten, ist es wahrscheinlich, dass die KI diese Position durch plausible und wenig kritische Ausführungen stützt. Zentral lassen sich zwei

² Auch wenn Begriffe wie Echokammer oder Filterblase im öffentlichen Diskurs sehr präsent sind, ist ihr tatsächlicher Einfluss auf die Gesellschaft wissenschaftlich längst nicht so eindeutig belegt. Eine umfassende Literaturübersicht der der University of Oxford zeigt beispielsweise, dass das Vorkommen von *Echokammern* deutlich seltener ist als gemeinhin angenommen. Nur eine kleine Minderheit konsumiert Informationen ausschließlich aus ideologisch homogenen Quellen. In der Praxis werden viele Menschen sogar mehr als in klassischen Medienzeiten durch soziale Medien mit diversen Perspektiven konfrontiert. Die Autoren dieser Studie argumentieren daher, dass der Diskurs über Echokammern selbst zu einem gesellschaftlichen Narrativ geworden ist, das über deren tatsächliche Verbreitung hinausgeht (Ross Arguedas et al., 2022).

systematische Gründe identifizieren, die dieses Phänomen erklären: Erstens wird der Nutzereingabe (im Folgenden nur noch als „Prompt“ bezeichnet), ein besonders hoher Kontextwert beigegeben. Zweitens ist dieser Mechanismus auch auf eine bewusste Designentscheidung im sogenannten *Reinforcement Learning From Human Feedback (RLHF)*³ zurückzuführen (González Barman et al., 2025; Xiao et al., 2025).

Dass der Prompt selbst so einen signifikanten Einfluss auf das Ergebnis hat, wurde bereits in zahlreichen Studien belegt: So zeigt sich etwa, dass schon subtile und scheinbar unauffällige Änderungen im Prompt, wie etwa Synonymersetzungen, das Ausgabeergebnis eines Large Language Models (LLM) signifikant beeinflussen und zum Beispiel die Wahrscheinlichkeit erhöhen, dass das gewünschte Zielkonzept erwähnt wird (Lin et al., 2025; Meincke et al., 2025; Pan et al., 2025; Sahoo et al., 2025; Sun & Kok, 2025). Die Modelle orientieren sich dabei eng an der sprachlichen Struktur, dem Tonfall und dem impliziten Rahmen der Eingabe.

Das Bestärkende Lernsystem (RLHF) wird eingesetzt, um Sprachmodelle so auszurichten, dass sie möglichst hilfreich, höflich und nutzerzentriert antworten (Bai et al., 2022; Ouyang et al., 2022). Eine Studie von Bardol (2025) zeigt etwa, dass negativ-emotionale Eingaben von GPT-4 systematisch übersteuert werden: Anstatt die negative Stimmung aufzunehmen, erzeugt das Modell häufig neutral oder sogar positiv gerahmte Antworten. Dadurch erhält das Gespräch eine bewusst positivere, optimistischere Konnotation, was auf eine eingebaute Tendenz hindeutet, den Dialog zu deeskalieren und in eine konstruktive Richtung zu lenken (Bardol, 2025). Dies hat das Ziel, Modelle nicht nur faktisch korrekt, sondern auch kommunikativ angenehm und affirmativ zu gestalten (Sharma et al., 2025). Diese Nutzerzentrierung führt jedoch dazu, dass Widerspruch oder Konfrontation tendenziell vermieden werden, und zwar selbst dann, wenn sie sachlich geboten wären. Dadurch entsteht eine eklatante Verzerrung der Chatbot-Ergebnisse zugunsten einer emotional positiven Gesprächsführung.

Sowohl auf Seiten der KI als auch der Menschen wirken Mechanismen, die es wahrscheinlicher machen, dass eine anfängliche Annahme im Dialog mit einem Sprachmodell bestätigt und verstärkt wird. Des Weiteren sind sie darauf ausgelegt, eine positive und bestärkende Gesprächserfahrung zu schaffen, anstatt ausschließlich nach einer möglichst richtigen Antwort zu streben.

³ RLHF bezeichnet ein Trainingsverfahren, bei dem KI-Modelle mithilfe menschlicher Bewertungen ihrer Antworten über ein Belohnungsmodell nachoptimiert werden, um ihr Verhalten an menschliche Präferenzen auszurichten.

Dadurch entstehen digitale Resonanzräume, in denen bestehende Vorannahmen oder Überzeugungen eher stabilisiert werden.

2.3 Digitale Steuerungsmechanismen in der Arbeitswelt

Der wechselwirksame Einfluss, den Technologie auf die soziale Wirklichkeit hat, zeigt sich auch in der Arbeitswelt, wo neue Technologien stets die Möglichkeitsräume der Arbeit beeinflussen und damit auch bestimmte Formen der Arbeitsorganisation ermöglichen. Das häufig herangezogene Beispiel der Erfindung der Dampfmaschine und der daraus hervorgehenden Fließbandarbeit ist vor allem ein plakatives Sinnbild für einen viel grundlegenderen Zusammenhang: Technologische Innovationen, ob sichtbar oder subtil, verändern die Möglichkeiten der Organisation und Kontrolle von Arbeit. Sie tragen beispielsweise dann zu einer Dequalifizierung von Arbeitskräften bei, wenn sie komplexe Tätigkeiten in einfachere, maschinell unterstützte Arbeitsschritte überführen, was massive Einschnitte in die Machtverteilung von Lohnarbeitern und Kapitalisten nach sich zieht (Braverman, 1998).

Heutzutage entstehen durch plattformbasierte Technologien neue Erwerbsformen wie *Crowdarbeit*⁴ und *Gigarbeit*⁵. Beide Erwerbsformen zeichnen sich dadurch aus, dass Arbeitsprozesse in einzelne, digital koordinierte Aufgaben zerlegt werden und oftmals mit unsicheren Einkommen, fehlender sozialer Absicherung und eingeschränkten Mitbestimmungsmöglichkeiten einhergehen (Kenney & Zysman, 2016). Gleichzeitig entstehen durch neue Technologien auch Requalifizierungsprozesse und neue Tätigkeitsfelder: Die Einführung von digital gesteuerten Produktionsanlagen erfordert etwa Fachkräfte, die diese Systeme programmieren, überwachen und warten. An vielen Stellen führt dies nicht zu einem einfachen „Abbau“ von Arbeit, sondern zu einer Verschiebung von Arbeitsinhalten von körperlich belastenden Tätigkeiten hin zu planenden, überwachenden und problemlösenden Aufgaben (Autor et al., 2003; Goos et al., 2014; Hirsch-Kreinsen, 2016).

Kellogg et al. (2020) greifen in ihrer Analyse „*Algorithms at Work*“ die einflussreiche Kontrolltypologie von Edwards (1979) auf, welche drei grundlegende Dimensionen organisatorischer

⁴ Unter Crowdarbeit werden Arbeitsformen verstanden, bei denen Aufgaben über digitale Plattformen an eine offene, meist anonyme Gruppe von Arbeitskräften vergeben werden. Die Tätigkeiten sind häufig kleinteilig, standardisiert und ortsunabhängig organisiert.

⁵ Gigarbeit bezeichnet demgegenüber kurzfristige, projekt- oder auftragsbasierte Tätigkeiten, die ebenfalls über digitale Plattformen vermittelt werden und außerhalb klassischer, langfristiger Beschäftigungsverhältnisse stattfinden.

Kontrolle unterscheidet: *Direction*, also die Vorgabe dessen, was zu tun ist; *Evaluation*, die Bewertung der Ausführung; und *Discipline*, die Sanktionierung oder Belohnung von Verhalten (Edwards, 1979). Darauf aufbauend identifizieren Kellogg et al. (2020) sechs Hauptmechanismen, die sich jeweils einer dieser drei Dimensionen zuordnen lassen. Diese werden in Tabelle 1 dargestellt.

Tabelle 1: Typologie algorithmischer Kontrolle

Edwards (1979)	Algorithmische Mechanismen (Kellogg et al., 2020)
Direction	Restricting (z. B. Vorgaben in Apps) Recommending (z. B. Vorschläge durch KI-Systeme)
Evaluation	Recording (z. B. lückenlose Datenerfassung) Rating (z. B. algorithmische Bewertungen)
Discipline	Replacing (z. B. algorithmische Entscheidung über Einsatz) Rewarding (z. B. Boni basierend auf Metriken)

Quelle: Eigene Darstellung nach Kellogg et al. (2020)

Die spezielle Wirkmächtigkeit algorithmischer Kontrolle ergibt sich aus spezifischen technologischen Eigenschaften. Die Autoren beschreiben diese Technologien entlang von vier zentralen Merkmalen algorithmischer Systeme: *comprehensive*, *instantaneous*, *interactive* und *opaque* (Kellogg et al., 2020). Konkret lassen sich diese wie folgt beschreiben: Algorithmische Systeme greifen auf eine breite und kontinuierlich wachsende Datenbasis zurück und sind dadurch besonders umfassend (*comprehensive*). Gleichzeitig werden diese Daten nahezu augenblicklich verarbeitet, sodass Anweisungen, Bewertungen oder Sanktionen unmittelbar erfolgen können (*instantaneous*). Zudem reagieren diese Systeme dynamisch auf das Verhalten der Beschäftigten und passen Aufgaben, Prioritäten oder Rückmeldungen laufend an (*interactive*). Ihre Funktionsweise sowie dessen Verantwortlichkeiten bleiben dabei für Nutzerinnen und Nutzer weitgehend

intransparent (*opaque*); diese Intransparenz führt zu einer Diffusion von Verantwortung und erschwert die Zuschreibung von Entscheidungen (Kellogg et al., 2020).⁶

Gerade diese Kombination aus technischer Steuerung und struktureller Intransparenz führt zu einer Verschiebung der Machtverhältnisse. Beschäftigte erhalten Aufgaben, Bewertungen oder Sanktionen über automatisierte Systeme, deren Funktionsweise ihnen nicht offenbart wird. Algorithmen wirken dabei nicht nur koordinierend, sondern auch disziplinierend und normativ: Sie filtern, bewerten, belohnen und bestrafen; und prägen damit unmittelbar das Verhalten der Arbeitenden (Kellogg et al., 2020). Aus Perspektive der Beschäftigten lässt sich das *algorithmische Management* kritisch bewerten, weil es in der Plattformökonomie systematisch die Autonomie der Beschäftigten untergräbt und ihnen zugleich die klassischen Instrumente des Arbeitskampfs und der Interessenvertretung entzieht (Heiland, 2020; Heiland & Schaupp, 2020; Pignot, 2023).

Diese, in den letzten Jahren im Bereich der Arbeits- und Industriesoziologie verstärkt formulierte Kritik an algorithmischer Kontrolle beruht im Wesentlichen auf der Offenlegung ihrer subtilen und intransparenten Lenkungsmechanismen, welche im Widerspruch zu den in diesem Bereich oft propagierten Ideal von Autonomie, Selbstbestimmung und Partizipation stehen⁷. Unter dem Versprechen von Eigenverantwortung und Effizienz ersetzen Betriebe traditionelle hierarchische Kontrolle durch datenbasierte, scheinbar objektive Mechanismen der Koordination und Bewertung, die Beschäftigte dazu anregen, ihre eigene Leistung permanent zu überwachen und zu optimieren (Manokha, 2018; Moore & Robinson, 2016; Srnicek, 2017; Yeung, 2018; Zuboff, 2018).

Werden anstelle klassischer algorithmischer Systeme selbstlernende Technologien in Arbeitsprozesse eingebunden, können sie unabhängig von intendierten Steuerungsabsichten einen strukturierenden Einfluss auf Arbeit entfalten. Dies ist unter anderem darauf zurückzuführen, dass solche Systeme in der Lage sind, große Datenmengen in Echtzeit zu analysieren und daraus

⁶ Besonders verdichtet lassen sich diese Steuerungsmechanismen in der Plattformökonomie finden. Arbeitsabläufe, Leistungserwartungen, Bewertungen und Sanktionen werden hier in vielen Bereichen vollständig durch algorithmische Systeme organisiert, deren Funktionsweise für die Beschäftigten weitgehend undurchsichtig bleibt. Die Plattform selbst fungiert dabei als intermediäre Kontrollinstanz, die hinter der technischen Architektur und der neutralisierenden Rhetorik lediglich als digitaler Vermittler tätig zu sein weitgehend unsichtbar wird (Ametowobla & Kirchner, 2023; Gillespie, 2010; Langley & Leyshon, 2017)..

⁷ Beispielsweise akquiriert die Fahrdienstleistungsplattform Uber neue Fahrerinnen und Fahrer mit den Worten: „Unlock flexible and independent opportunities to make money when you choose to drive with Uber. The app gives you the flexibility to set your own hours and make money on your time“ (*Driver Jobs in Berlin - Choose Your Hours*, o. J.). Algorithmische Steuerungssysteme knüpfen dabei paradox an jene seit den 1970er-Jahren zentralen Leitbilder von Flexibilität und Selbstverwirklichung an, die den Wandel von fordistischer Fremdsteuerung zu postfordistischer Selbstführung begleitet haben (Boltanski & Chiapello, 2006; Sennett, 1998).

detaillierte Leistungs-, Verhaltens- und Effizienzprofile abzuleiten. Während frühere algorithmische Systeme vor allem regelbasiert arbeiteten, können KI-Modelle Muster erkennen, Vorhersagen treffen und Entscheidungen automatisiert anpassen, ohne dass menschliche Intervention notwendig ist (Jordan & Mitchell, 2015; Russell & Norvig, 1995). Dadurch entstehen engmaschigere und dynamischere Formen der Überwachung, die nicht nur Arbeitsprozesse optimieren, sondern durch personalisierte Interaktion, adaptive Zielvorgaben und kontinuierliche Bewertung dieser auch subtile Verhaltenssteuerung ermöglichen.

Besonders die Architektur selbstlernender Systeme im Bereich des *unüberwachten Lernen* (unsupervised learning)⁸ trägt maßgeblich dazu bei, dass algorithmische Entscheidungslogiken bei der betrieblichen Nutzung an Einfluss auf den Arbeitsprozess gewinnen. Solche Systeme entwickeln eine eigene, maschinell gelernte Form der Mustererkennung, die sich von der lebensweltlichen Sinnggebung menschlicher Akteure lösen kann. Damit sind sie nicht mehr wie klassische Technologien intentional als aufgabenspezifische Werkzeuge entworfen, deren Funktionsweise sich eindeutig nachvollziehen oder gar kontrollieren lassen. Während klassische Technologien anhand von vorab definierten Anweisungen funktionieren, tritt an deren Stelle bei der Nutzung von KI zunehmend eine interaktive, strategische Einflussnahme, bei der Ergebnisse eher ausgehandelt als eindeutig vorgegeben werden (Schulz-Schaeffer, 2025). Wird KI im betrieblichen Kontext eingesetzt, nimmt ihre zugrunde liegende Architektur und die damit ermöglichten und beschränkten Handlungsoptionen potenziell Einfluss auf Arbeitspraktiken.

Insgesamt zeigt sich, dass technologische Innovationen die Arbeitswelt nicht linear „verbessern“ oder „verschlechtern“, sondern sie auf ambivalente Weise transformieren und die strukturellen Machtverhältnisse beeinflussen. Sie eröffnen neue Handlungs- und Gestaltungsmöglichkeiten, schaffen jedoch auch neue Konfliktlinien und Machtasymmetrien. Aus einer soziotechnischen Perspektive ist Arbeit damit als dynamisches Ko-Produkt von sozialen Praktiken und technischen Artefakten zu verstehen: Technologien sind weder neutrale Werkzeuge noch autonom wirkende Kräfte, sondern eingebettet in soziale, ökonomische und politische Aushandlungsprozesse, welche beeinflussen, wie Arbeit organisiert, verteilt und erlebt wird.

⁸ Überwachtes Lernen (*supervised learning*) bezeichnet Verfahren des maschinellen Lernens, bei denen ein Modell aus annotierten Trainingsdaten lernt, d. h. zu jedem Eingabebeispiel liegt eine bekannte Zielvariable (z. B. Klasse oder numerischer Wert) vor. Ziel ist es, eine Abbildung von Eingaben auf Ausgaben zu approximieren, etwa für Klassifikations- oder Regressionsaufgaben. Unüberwachtes Lernen (*unsupervised learning*) hingegen nutzt ausschließlich unannotierte Daten und zielt darauf ab, verborgene Strukturen, Muster oder Verteilungen in den Daten zu identifizieren, z. B. durch Clustering oder Dimensionsreduktion (Bishop, 2006; Jordan & Mitchell, 2015)

2.4 Automatisierung und Augmentation

Dieser Abschnitt widmet sich zuletzt noch der Unterscheidung zwischen Automatisierung und Augmentation als zentralem Deutungsrahmen aktueller Diskurse über den Einsatz von KI in der Arbeitswelt. Entlang dieser beiden Konzepte zeigt sich, wie die Rolle von KI antizipiert, bewertet und normativ gerahmt wird. Während Automatisierung die Substitution menschlicher Arbeit und den Verlust von Handlungsträgerschaft im Arbeitsprozess oder an Arbeitsplätzen meint, verweist Augmentation auf Vorstellungen der Unterstützung, Erweiterung und Entlastung menschlicher Tätigkeit. Beide Perspektiven prägen gegenwärtige Debatten über KI-Nutzung in der Arbeitswelt und beeinflussen, welche Erwartungen und Aversionen an technologische Entwicklungen herangetragen werden können. Ziel dieses Abschnitts ist es, die Begriffe im aktuellen Diskurs einzuordnen und empirisch zu prüfen.

Die Diskussion um die von Volkswagen konzipierte hochautomatisierte Produktionsstätte *Halle 54* Anfang der 1980er-Jahre wurde schnell zum Symbol einer dystopischen Zukunft menschenleerer sogenannter „lights-out“-Fabriken⁹. Medienberichte und Unternehmensdarstellungen erweckten den Eindruck, Roboter könnten bald ganze Fabriken übernehmen. Ein Szenario, das bei Gewerkschaften und Beschäftigten die Angst vor massiver technikbedingter Arbeitslosigkeit auslöste (Heßler, 2014). Dagegen zeigten damalige Studien wie etwa von Barley (1986) bereits, dass die Einführung neuer Technologien zwar tiefgreifende Auswirkungen auf Handlungsspielräume, Qualifikationsanforderungen und Machtverhältnisse innerhalb von Organisationen haben können, jedoch nicht zwangsläufig zu massenhafter Arbeitslosigkeit führen.

Heute, mit den Entwicklungen in den Bereichen künstlicher Intelligenz, maschinellem Lernen und automatisierter Entscheidungsfindung, erleben diese Ängste wieder eine neue Konjunktur: Die Sorge vor weitreichender Arbeitslosigkeit infolge der flächendeckenden Implementierung von KI-Systemen ist insbesondere im Dienstleistungs- und Wissensarbeitssektor stark präsent. Prominente Artikel wie der von Frey und Osborne (2017), welcher prognostiziert, dass rund 47 % der US-amerikanischen Arbeitsplätze ein hohes Automatisierungsrisiko aufweisen, haben diese Diskussion weiter angeheizt. Diese viel diskutierte Prognose wurde dabei unter anderem auch von einer OECD-Studie (Arntz et al., 2016) aufgegriffen und in einer Folgestudie betont, dass nicht ganze Berufe automatisiert werden, sondern nur bestimmte Tätigkeiten innerhalb eines Berufs. Dies

⁹ Lights-out-Fabriken (auch „Dark Factories“) bezeichnen Produktionsstätten, die weitgehend oder vollständig automatisiert arbeiten und deshalb kein dauerhaftes menschliches Personal benötigen, weshalb das Licht ausgeschaltet bleiben kann.

miteinbezogen kommen die Autoren nur auf einen deutlich geringeren Anteil potenziell gefährdeter Jobs, insgesamt 9 % im Mittel aller OECD-Länder (Arntz et al., 2016). Bei der Frage nach der Substitution von Arbeit ist also ein ganzheitlicher Blick wichtig, insbesondere weil Teilsubstitutionen mit dynamischen Anpassungsprozessen innerhalb von Branchen sowie der Möglichkeit struktureller Transformationen einhergehen. So prognostiziert eine die Studie von Eloundou et al. (2023) beispielsweise, dass zwar etwa 80 % der US-Arbeitskräfte mindestens 10 % ihrer Arbeitsaufgaben von der Einführung großer Sprachmodelle betroffen sehen könnten, während für nur rund 19 % der Arbeitnehmer ein deutlich signifikanter Teil ihrer Arbeit von mindestens 50 % ihrer Aufgaben betroffen sein könnte. Dazu ist zu betonen: Auch wenn einzelne Tätigkeiten an KI-Systeme delegiert werden können, wird praktisch auch oft nicht weniger gearbeitet. Berufe mit hoher KI-Exposition zeigen beispielsweise tendenziell längere Arbeitszeiten und eine Reduktion der Freizeit (Jiang et al., 2025). Der Grund dafür, dass es trotz Automatisierungsbestrebungen immer noch so viele Jobs gibt, liegt also darin, dass diese stets komplementäre, menschliche Arbeit erzeugt (Autor, 2015). Dabei wird zunehmend deutlich, dass es weniger um die vollständige Ersetzung von Arbeitskräften geht, sondern vielmehr um eine tiefgreifende Transformation von Arbeitsinhalten, Qualifikationsanforderungen und Beschäftigungsstrukturen. Dort, wo einzelne Tätigkeiten automatisiert werden, werden weiterhin Menschen zur Kontextualisierung, Überprüfung und Einbettung der Ergebnisse benötigt (Pfeiffer, 2020). Butollo (2024) weist der Automatisierung insgesamt drei verschiedene Bumerang-Effekte (*Rebound-Effekte*) zu, welche zusätzliche Arbeitslast erzeugen. Erstens die steigende Komplexität von Produkten und Wertschöpfungsketten, zweitens der erhebliche Einführungs- und Anpassungsaufwand digitaler Technologien sowie drittens der wachsende Bedarf an sozialer und infrastruktureller Reproduktionsarbeit. Diese sorgen dafür, dass menschliche Arbeit weiterhin zentral bleibt. Dabei werden zwar einzelne Tätigkeiten automatisiert, die Gesamtarbeitsmenge nimmt aber häufig eher zu als ab (Butollo, 2024).

Zusammenfassend lässt sich festhalten, dass der Einsatz von KI zwar in vielen Bereichen zu einer Automatisierung von Tätigkeiten führt und damit bestimmte Arbeitsaufgaben ersetzt oder vereinfacht, gleichzeitig jedoch auch eine weitreichende Augmentation von Arbeit bewirkt. Da KI-Systeme enorme Mengen an Daten, Inhalten und Prozessen generieren, entsteht ein wachsender Bedarf an Tätigkeiten, welche diese Ergebnisse kontextualisieren, interpretieren und kuratieren. Somit kann angenommen werden, dass insbesondere in Bereichen, die sich mit der Steuerung, Bewertung und Integration der von KI erzeugten Ergebnisse befassen, die Nutzung von KI nicht zwangsläufig zu einem Rückgang, sondern vielmehr zu einer Transformation und Zunahme von Arbeit führt.

3 Arbeitsfeld Softwareentwicklung

Anschließend an den im letzten Kapitel dargestellten allgemeinen Forschungsstand zur Arbeit mit Künstlicher Intelligenz (KI) widmet sich das zweite Kapitel nun den organisationalen und kulturellen Rahmenbedingungen der Softwareentwicklung als Arbeitsfeld. Da Reaktionen auf KI-Technologien ebenso wie ihre konkrete Implementierung in Arbeitsprozesse wesentlich von historisch gewachsenen Strukturen und normativen Leitbildern innerhalb dieser Berufsgruppe abhängt, soll die Profession selbst in ihren zentralen Merkmalen in den Blick genommen werden. Das Kapitel ist in vier Abschnitte gegliedert. Zunächst werden professionssoziologische Grundlagen vorgestellt, die als theoretischer Rahmen für die weiteren Abschnitte dienen. Anschließend wird mit der Open-Source-Kultur eine historisch gewachsene Form der Wissensproduktion betrachtet, die das Selbstverständnis und die Arbeitspraktiken der Softwareentwicklung nachhaltig geprägt hat. Darauf aufbauend widmet sich das dritte Unterkapitel den beruflichen Leitbildern und Rollenverständnissen innerhalb des Feldes. Den Abschluss bildet ein Überblick über die aktuellen Einsatzfelder und Rahmenbedingungen generativer KI in der Softwareentwicklung.

3.1 Professionstheorie

Berufliche Hegemonie basiert auf Exklusivität des Wissens, spezialisierten Ausbildungswegen sowie der Fähigkeit, den Zugang zu beruflicher Praxis zu regulieren (Freidson, 1984, 2017).¹⁰ Professionen sichern ihre Autorität demnach nicht allein durch ihre Kompetenzen, sondern auch durch erfolgreiche Grenzziehungsprozesse gegenüber konkurrierenden Berufsgruppen. Andrew Abbott (1988) erweitert diese Perspektive um die dynamische Dimension professioneller Machtverhältnisse. In seiner Theorie versteht er Professionsbildung als Ergebnis fortlaufender, interdependenter Aushandlungsprozesse. Professionen stehen demnach in dauerhafter Konkurrenz um sogenannte *Jurisdiktionen*, also um klar abgegrenzte Zuständigkeits- und Aufgabenbereiche. Diese werden kontinuierlich durch gesellschaftliche Veränderungen, staatliche Regulierung und technologischen Wandel beeinflusst (Abbott, 1988).

¹⁰ Die Professionssoziologie ab den 50er Jahren thematisierte zunächst die Frage, anhand welcher Kriterien eine Tätigkeit als Profession klassifiziert wird, und knüpft dabei an Parsons' (1991) funktionalistische Perspektive an, die Professionen als zentrale Mechanismen zur Stabilisierung gesellschaftlicher Ordnung versteht. Einen stärker machttheoretischen Zugang derselben Disziplin entwickelte später Eliot Freidson (2017), welcher Professionen als soziale Gruppen beschreibt, die versuchen, Kontrolle über ihr spezifisches Wissensgebiet zu erlangen und Monopolstellungen zu sichern.

In neueren professionssoziologischen Ansätzen wird zunehmend betont, dass Professionen nicht allein durch institutionelle Strukturen oder machstrategische Abgrenzungen konstituiert werden, sondern wesentlich auch durch gemeinsame Werte, normative Orientierungen und professionelle Identitäten (Evetts, 2014; Noordegraaf, 2015; Saks, 2012). Professionen werden als soziale Gemeinschaften verstanden, deren Mitglieder wechselseitig auf Anerkennung ihrer Expertise und ihres gesellschaftlichen Beitrags angewiesen sind. Gesellschaftliche Aushandlungen lassen sich aus dieser Perspektive generell als Kämpfe um Anerkennung verstehen (Honneth, 2018). Die Sicherung beruflicher Autonomie und Zuständigkeit ist somit stets auch eine Frage der sozialen Wertschätzung und moralischen Legitimation durch gesellschaftliche Relevanz einer Profession.

Ein ergänzender theoretischer Zugang ergibt sich aus Gieryns (1983) Konzept der Grenzarbeit („*Boundary work*“). Der Begriff beschreibt die symbolische und diskursive Abgrenzungsarbeit, mit der soziale Gruppen ihre Zuständigkeiten, Autorität und Zustimmungsansprüche gegenüber anderen Bereichen behaupten oder neu definieren (Gieryn, 1983). Grenzarbeit ist dabei nicht ausschließlich defensiv ausgerichtet: Die Autoren Langley et al. (2019) unterscheiden in ihrer ausführlichen Kartographie verschiedene Formen von Grenzarbeit: Kompetitive Grenzarbeit bedeutet, dass konkurrierende Gruppen etwa bewusst Grenzen betonen, um sich abzugrenzen und Vorteile zu sichern. Kollaborative Grenzarbeit meint genau im Gegenteil, dass Grenzen flexibel gehandhabt oder gar überbrückt werden, damit Zusammenarbeit gelingt. Konfigurationale Grenzarbeit beschreibt schließlich das bewusste Anordnen oder Neugestalten von Grenzen durch externe Personen, um ganz neue Bereiche zu konstituieren (Langley et al., 2019).

Aufbauend auf dieser Kartographie führen die Autoren Foulconbridge et al. (2023) das Konzept der ineinandergreifenden Grenzarbeit („*Intertwined Boundary Work*“) ein, indem sie zeigen, dass Beschäftigte in der Wissensarbeit als Reaktion auf die Einführung Künstlicher Intelligenz verschiedene Formen der Grenzarbeit gleichzeitig und miteinander verflochten einsetzen. Sie verteidigen einerseits ihre angestammten beruflichen Zuständigkeiten gegen die Automatisierung durch KI, suchen aber zugleich aktiv die Kooperation, um die neuen Technologien sinnvoll in ihre Arbeit zu integrieren. Darüber hinaus nutzen sie die Veränderungen, um ihre eigenen beruflichen Rollen neu zu definieren und weiterzuentwickeln. Die Autoren machen damit deutlich, dass es sich um einen komplexen, dynamischen Aushandlungsprozess handelt, in dem berufliche Identitäten, Zuständigkeiten und Praktiken in Reaktion auf technologische Umbrüche neugestaltet werden (Faulconbridge et al., 2023). Grenzziehungsarbeit ist immer auch ein Akt der relationalen Selbstverortung: Es geht darum, die eigene berufliche Identität, Aufgabenfelder und Zuständigkeiten zu definieren, indem man sie symbolisch von anderen Bereichen abgrenzt oder bewusst in

Beziehung zu ihnen setzt. Damit ist Grenzziehungsarbeit nicht nur ein Prozess der Abwehr oder Kooperation zwischen sozialen Gruppen, sondern auch ein Mittel, technologische Artefakte wie KI in das berufliche Feld einzuordnen.

So lässt sich festhalten, dass Berufe aus soziologischer Sicht aktiv hergestellt und verteidigt werden. Die Konstitution von Professionen ist demnach ein gesellschaftlicher, historisch variabler Prozess, in dem sich fachliche Expertise, berufliche Autonomie und soziale Legitimation wechselseitig bedingen.

3.2 Open-Source-Kultur als zentrale Wissensproduktion

Ein zentraler Bestandteil der Softwareentwicklung ist die *Open-Source-Kultur*. Darunter wird die Praxis verstanden, Software so zu entwickeln und zu veröffentlichen, dass ihr Quellcode frei zugänglich ist und von anderen eingesehen, genutzt, verändert und weiterverbreitet werden kann. Diese Offenheit bildet eine wichtige Grundlage für den Wissensaustausch und die Zusammenarbeit innerhalb des Berufsfeldes. Der hier folgende Abschnitt beleuchtet daher die Bedeutung von Open-Source für die Softwareentwicklung, um jene grundlegenden Strukturen sichtbar zu machen, die auch für die spätere Diskussion über den Einfluss von KI-Technologien in diesem Arbeitsfeld maßgeblich sind.

Nach den historischen Beschreibungen von Bretthauer (2001) entwickelte sich die Open-Source-Kultur als Praxis aus den spezifischen sozialen und technischen Gegebenheiten der Zeit heraus. In den 1970er- und frühen 1980er-Jahren war es in vielen wissenschaftlichen und technischen Kontexten üblich, Quellcode frei zu teilen, anzupassen und weiterzugeben was eine Fortsetzung jener akademischen Normen des offenen Austauschs bedeutete, die schon zuvor die Informatik geprägt hatten. Erst mit der zunehmenden Kommerzialisierung der Softwareindustrie, die durch rechtliche Veränderungen im Urheberrecht begünstigt wurde, entstand ein Spannungsverhältnis zwischen proprietären und offenen Entwicklungsformen. Wiederum als Reaktion auf die zunehmende Privatisierung von Software kristallisierten sich in den 1980er- und 1990er-Jahren konkrete Projekte heraus, die das Ziel hatten, freie Betriebssysteme zu schaffen und die Grundlagen der heutigen Open-Source-Bewegung legten. Besonders prägend waren hierbei die Entstehung offener Betriebssysteme aus dem GNU-Projekt, die Weiterentwicklung der BSD-Systeme sowie die weltweite Verbreitung des Linux-Kernels. Diese Projekte wurden nicht nur durch technologische Innovationen getragen, sondern vor allem durch die Herausbildung eigenständiger Entwicklungsgemeinschaften, die auf Prinzipien wie Kooperation, Transparenz und gemeinschaftliche Kontrolle setzten. Bretthauer zeigt damit, dass sich die Open-Source-Kultur in einem mehrstufigen

Prozess etablierte. Beginnend als eine implizite Praxis kollektiven Teilens, über erste bewusste Abgrenzungen gegenüber proprietären Modellen hin zu einer konsolidierten Bewegung, die mit GNU, BSD und Linux ihre zentralen Bezugspunkte fand und die Wissensproduktion in der Softwareentwicklung dauerhaft prägte (Bretthauer, 2001; DiBona & Ockman, 1999; Tozzi, 2017).

Die von Steven Levy in seinem Werk *Hackers: Heroes of the Computer Revolution* (1984) erstmals systematisch beschriebene *Hackerethik* stellte eine normative Grundlage der frühen US-amerikanischen Hacker-Kultur, wie sie zunächst am MIT praktiziert wurde, dar. Ihre Kernprinzipien, freier Zugang zu Computern, Offenheit von Informationen, Misstrauen gegenüber Autoritäten, Meritokratie, schöpferischer Umgang mit Technik und der Glaube an die gesellschaftlich positive Kraft von Computern, wurden zu einem kulturellen Referenzrahmen, der weit über den akademischen Kontext des MIT hinaus Wirkung entfaltete. In Europa fand diese Ethik eine Institutionalisierung durch den im Jahr 1981 gegründeten Chaos Computer Club (CCC), der die Prinzipien in kollektive Strukturen überführte und sie um spezifische Leitlinien wie den Schutz privater Daten und den freien Zugang zu öffentlichen Informationen erweiterte. Damit transformierte der CCC die von Levy dokumentierte, eher lose Subkultur in eine dauerhafte Organisation, die bis heute als Sprachrohr der Hacker-Gemeinschaft wirkt und die Ethik durch praxisorientierte Aktionen von Sicherheitsanalysen bis zu politischen Stellungnahmen in gesellschaftliche Debatten einbringt (CCC | *Hacker Ethics*, 2025).

Die Grundsätze der Hackerethik und die damit verbundene Open-Source-Kultur bringen in der Praxis konkrete Vorteile mit sich. Anschaulich beschrieben werden diese von Raymond in seinem Werk *The Cathedral and the Bazaar* (1998). In seinem Buch kontrastiert Raymond zwei Entwicklungsmodelle: das „Kathedralen“-Modell, welches für geschlossene, hierarchische und zentral geplante Softwareentwicklung steht, und das „Basar“-Modell, welches für offene, dezentrale und kollaborative Entwicklungsweisen steht. Während in traditionellen, hierarchisch organisierten „Kathedralen“-Projekten nur wenige Auserwählte am Quellcode arbeiten und Neuerungen erst nach langen Entwicklungszyklen veröffentlicht werden, entfaltet sich im „Basar“-Modell eine Kultur der kontinuierlichen Beteiligung. Der Quellcode ist öffentlich zugänglich, Fehler und Verbesserungsvorschläge können von einer großen, vielfältigen Gemeinschaft eingebracht werden, und neue Versionen erscheinen in kurzen Abständen (Raymond, 1998).¹¹

¹¹ Die Stärke dieser Vorgehensweise liegt nach Raymond darin, dass Wissen und Erfahrung aus vielen unterschiedlichen Perspektiven zusammenfließen. Mit seinem berühmten Linus's Law („Given enough eyeballs, all bugs are shallow“) beschreibt er den Mechanismus, dass Fehler schneller erkannt und behoben

In der Praxis zeigt sich die Open-Source-Kultur am häufigsten auf Plattformen wie GitHub, Stack Overflow oder Reddit, die heute zentrale Orte des digitalen Austauschs sind. Auf GitHub können Entwicklerinnen und Entwickler gemeinsam an Softwareprojekten arbeiten, deren Quellcode öffentlich zugänglich ist. Es ist dann in Versionskontrollsystemen möglich, Programme nicht nur herunterladen und zu nutzen, sondern auch Fehler zu melden, Verbesserungsvorschläge zu machen oder sogar selbst Änderungen im Quellcode vorzuschlagen. Letzteres läuft über sogenannte „Pull Requests“: Jemand entwickelt eine Lösung für ein Problem, stellt sie der Gemeinschaft vor, und andere prüfen, diskutieren und verbessern den Vorschlag, bevor dieser gegebenenfalls in das Projekt übernommen wird.¹² Im Jahr 2025 zählt *GitHub* über 150 Millionen registrierte Nutzende der Plattform (*About GitHub*, 2025). Diese enorme Nutzerbasis macht GitHub zur mit Abstand größten Plattform für kollaborative Softwareentwicklung weltweit.

Das Forum Stack Overflow zielt ebenfalls auf Wissensaustausch ab, ist aber grundlegend anders strukturiert. Hier können Entwicklerinnen und Entwickler Fragen zu konkreten Problemen von einfachen Fehlern im Code bis hin zu komplexen Architekturfragen stellen. Andere Mitglieder antworten direkt, geben Tipps oder verweisen auf passende Ressourcen. Fragen und Antworten werden dabei stets von der Gemeinschaft bewertet: Die hilfreichsten steigen nach oben und sind so für alle späteren Lesenden sofort sichtbar. Dadurch entsteht über die Zeit ein riesiges, frei zugängliches Wissensarchiv, das weltweit genutzt wird, um Entwicklungsprobleme zu lösen. Diese zentrale Stellung erklärt sich aus den strukturellen Eigenschaften softwaretechnischer Arbeit: Softwareentwicklung ist durch hohe Dynamik, schnelle technologische Zyklen sowie eine Vielzahl kontextspezifischer und kleinteiliger Problemlagen geprägt. Unter diesen Bedingungen erweisen sich institutionalisierte Wissensformen wie etwa Dokumentationen, Standards oder formale Supportstrukturen häufig als unzureichend, da sie mit der situativen und kontingenten Praxis der Entwicklung nicht Schritt halten können. Wissen entsteht stattdessen in konkreten Handlungssituationen und bleibt oftmals fragmentiert und personengebunden. Plattformen wie Stack Overflow ermöglichen es, diese situierte Expertise in eine öffentlich zugängliche, wenn auch

werden, wenn genügend Entwicklerinnen und Entwickler den Code prüfen. Open-Source-Projekte profitieren somit von einer Art kollektiver Intelligenz: Sie sind anpassungsfähiger, weil Rückmeldungen und Korrekturen unmittelbar einfließen, und zugleich stabiler, weil Transparenz und Peer-Review zu einer hohen Codequalität beitragen (Raymond, 1998).

¹² Auch in Entwicklerteams, die an proprietären oder organisationsinternen Projekten arbeiten, finden vergleichbare Austauschprozesse statt, allerdings in einem deutlich kleineren und abgeschlossenen Rahmen. Dieser Austausch ist häufig stärker durch formale Vorgaben, Projektziele und organisationale Hierarchien geprägt, wobei alle Beteiligten in der Regel auf denselben Input zurückgreifen. Innovation entsteht zwar auch hier kollaborativ, bleibt jedoch nicht öffentlich sichtbar.

informell organisierte Form zu überführen (Abdalkareem et al., 2017; Barua et al., 2014; Treude et al., 2011; Wu et al., 2019). Zwar ist die Plattform in ihrer Funktion zentral darauf ausgerichtet, technische Probleme gemeinschaftlich zu lösen und Wissen frei zugänglich zu machen, jedoch stößt das System manchmal auf Grenzen: Empirische Untersuchungen zeigen, dass auf Stack Overflow Phänomene wie das selektive Zurückhalten von Wissen und teilweise abwertende Kommunikationsformen verbreitet sind. Insbesondere neu auf die Plattform kommende Entwicklerinnen und Entwickler berichten von Problemen, sich anfänglich zielführend und sicher auf der Plattform zu bewegen. Sie sehen sich manchmal sogar mit Kritik oder Herabsetzungen konfrontiert, wenn ihre Fragen nicht den etablierten Kommunikationsnormen der „Community“ entsprechen (Cheriyian et al., 2021; Krishtul et al., 2022). Solche Exklusionsmechanismen führen nicht nur zu negativen sozialen Erfahrungen und eingeschränkter Teilhabe, sondern unterlaufen auch die zentrale Funktion des Forums: die offene Weitergabe von Wissen.

Die Plattform Reddit stellt einen allgemeinen Diskussionsraum dar, in dem vielfältige Inhalte pseudonym erstellt, bewertet und diskutiert werden können. Dadurch ermöglicht Reddit einen offenen Austausch zwischen den Nutzenden. Gleichzeitig spiegeln die auf der Plattform etablierten Moderationspraktiken sowie spezifische Community-Regeln informelle Diskussionspraktiken wider, in denen sich gesellschaftliche Konflikte, Normen und Machtverhältnisse zeigen. Reddit kann somit sowohl als Ort diskursiver Aushandlung als auch als Raum für Polarisierung und problematische Diskurse verstanden werden. Auf der Plattform haben sich zahlreiche programmspezifische Gruppen (*Subreddits*) etabliert, in denen neben fachlichen Fragen auch subjektive Empfindungen zum Lernprozess selbst Gegenstand des Austauschs sind (Liu & Anwar, 2022).

Insgesamt zeigt sich, dass die Open-Source-Kultur ihren Ursprung in akademischen Wissensgemeinschaften hat, in denen gemeinschaftliches und frei zugängliches Arbeiten zunächst selbstverständlich war und keiner besonderen Rechtfertigung bedurfte. Erst im Zuge der zunehmenden Kommerzialisierung von Software entwickelten sich proprietäre Produktions- und Verwertungsmodelle, denen Open-Source nach und nach als bewusstes Gegenkonzept gegenübergestellt wurde. In der heutigen Programmierpraxis ermöglichen kollaborative Plattformen wie GitHub oder Stack Overflow eine gemeinsame, iterative Weiterentwicklung von Code. Gleichzeitig sind auch diese digitalen Räume nicht frei von sozialen Barrieren, insbesondere für Personen, die mit den informellen Kommunikations- und Beteiligungsregeln der jeweiligen Plattformen nicht vertraut sind.

3.3 Leitbilder und Rollenverständnisse in der Softwareentwicklung

Aufbauend auf der professionssoziologischen Einordnung lässt sich die berufliche Identität von Softwareentwicklerinnen und -entwicklern insbesondere durch die Leitbilder und Narrative verstehen, die das Feld von innen heraus prägen. Solche kulturellen Orientierungsmuster strukturieren das professionelle Selbstverständnis, indem sie beschreiben, was „gute“ Softwareentwicklung ausmacht, welche Werte und Praktiken als erstrebenswert gelten und wie sich die einzelnen Berufsgruppen selbst in ihrem Arbeitskontext positionieren und abgrenzen.

Verschiedene Untersuchungen verdeutlichen konkrete Leitbilder, nach denen das Selbstverständnis in der Softwareentwicklung konstruiert wird: Auf der einen Seite existiert ein ingenieurwissenschaftlich geprägtes Rollenbild, welches die Effizienz und Standardisierung der Softwareentwicklung betont. Diese Bezugspunkte gaben vor allem der noch jungen Disziplin zu Beginn ein vergleichsweise hohes Maß an Professionalität und Legitimation. Mit der zunehmenden Komplexität von Softwareprojekten und der wachsenden Bedeutung von Kreativität, Anpassungsfähigkeit und Nutzerorientierung verschiebt sich dieses Selbstverständnis allmählich. Spätestens seit der Jahrtausendwende hat sich verstärkt ein weiterer Wertehorizont herausgebildet, der unter dem Begriff „*Software Craftsmanship*“ diskutiert wird. In diesem Paradigma gilt Softwareentwicklung stärker als Handwerk, in dem Erfahrung, ästhetisches Empfinden für Codequalität und kontinuierliches Lernen zentrale Rollen einnehmen. Entwicklerinnen und Entwickler verstehen sich hier nicht allein als Produzenten funktionierender Systeme, sondern als kreative Gestalter von langlebigen Lösungen (Jemielniak, 2008; Mancuso, 2015; McBreen, 2002; Sundelin et al., 2021). Während klassische Ingenieure, insbesondere in Bereichen wie Maschinenbau oder Elektrotechnik, lange Zeit die zentrale Elite in Unternehmen bildeten, gewinnen heute Softwareentwickler, Data Scientists und UX/UI-Designer, immer mehr Einfluss. Diese „neuen“ Tech-Angestellten bringen nicht nur andere technische Fähigkeiten mit, sondern auch andere Arbeitskulturen: agile Methoden, flache Hierarchien und eine stärkere Betonung von Kreativität und Autonomie. Der gegenwärtige Zustand wird dabei als eine Phase der „unsicheren Elite“ beschreiben: Beide Gruppen sind für die Unternehmen unverzichtbar, doch ihre Rollen, Statuspositionen und Zukunftsperspektiven sind in Konkurrenz zueinander (Krzywdzinski et al., 2025).

Parallel dazu hat die Verbreitung agiler Methoden eine Verschiebung hin zu kollektiv orientierten Identitätsansätzen eingeleitet. Studien zeigen, dass Entwicklerinnen und Entwickler in agilen Umfeldern ihre Rolle stärker über Zusammenarbeit, Kommunikation und geteilte Verantwortung definieren (Hoda et al., 2013; Sharp & Robinson, 2008). Dies bedeutet, dass die Identität nicht ausschließlich aus der Beherrschung von Technologien hervorgeht, sondern ebenso aus der

Fähigkeit, sich in interdisziplinären Kontexten zu positionieren und Brücken zwischen technischen und fachlichen Anforderungen zu schlagen und ein gemeinsames „Wir-Gefühl“ zu erzeugen (Santos et al., 2023).

Zusammenfassend wird deutlich, dass die berufliche Identität in der Softwareentwicklung von verschiedenen historisch gewachsenen und zugleich im Wandel befindlichen Leitbildern geprägt ist. Neben dem ursprünglich dominanten ingenieurwissenschaftlichen Selbstverständnis haben sich mit den agilen Rollenmodellen neue Narrative etabliert, die Kreativität, kontinuierliches Lernen, Teamorientierung und Verantwortung für Codequalität betonen. Die *Open-Source-Kultur* als Form der Wissensproduktion ist dabei ein zentraler Einfluss, welcher Werte wie Autonomie, gemeinschaftliche Wissensproduktion und meritokratische Anerkennung tief in die Identitätsbildung der Berufsgruppe eingeschrieben hat.

3.4 Einsatz von KI in der Softwareentwicklung

Während die vorherigen Abschnitte die historischen Wurzeln und das berufliche Selbstverständnis der Softwareentwicklung beschrieben haben, richtet sich der Blick nun auf die konkrete Integration von Künstlicher Intelligenz in den Arbeitsalltag. Da für die Softwareentwicklung grundlegende Tätigkeiten wie das Quellcodeschreiben bereits durch KI-Systeme automatisiert werden können, ist ein Blick auf die Funktionsweisen zentral, um zu verstehen, welchen Einfluss die KI auf die alltägliche Arbeitspraxis hat. Dieser Abschnitt gibt einen Überblick über aktuelle Anwendungen, ihre Leistungsfähigkeit und die damit verbundenen Herausforderungen und schafft damit die Grundlage für die anschließende Betrachtung der Auswirkungen auf berufliche Rollen und Zuständigkeiten.

KI-gestützte Anwendungen finden sich inzwischen entlang der gesamten Wertschöpfungskette der Softwareproduktion, von der Planung über den Schreibprozess bis hin zu Test, Betrieb und Wartung von Systemen. Sprachbasierte Transformermodelle wie GitHub Copilot oder ChatGPT sind in der Lage, im laufenden Schreibprozess auf Basis von Kommentaren oder Fragmenten bestehenden Codes vollständige Funktionen vorzuschlagen, wodurch sich sowohl die Entwicklungszeit als auch die Fehleranfälligkeit reduzieren lassen kann (Husein et al., 2025; Imai, 2022; Peng et al., 2023). Auch die automatisierte Code-Dokumentation stellt einen zentralen Anwendungsbereich dar: Large Language Modelle generieren erklärende Kommentare oder technische Beschreibungen, die langfristig zur Verbesserung der Wartbarkeit und der interpersonellen Verständlichkeit beitragen (Macke & Doyle, 2024; Shi et al., 2022). Darüber hinaus werden auf maschinellem Lernen basierende Modelle zunehmend auch für das automatisierte Testen von

Software eingesetzt. Die automatisierte Generierung von solchen sogenannten *Unit-Tests* kann die Testabdeckung signifikant erhöhen und somit die Qualitätssicherung optimieren (Chang & Shokrolah Shirazi, 2025; Schäfer et al., 2024; Yuan et al., 2024). Eine Studie des Capgemini Research Institute prognostiziert, dass bis 2026 etwa 85 % der Softwareentwickler generative KI nutzen werden, gegenüber fast 50 % im Jahr 2024 („Gen AI in Software“, 2024).

Eine von den Autoren Zhang et al. (2023) durchgeführte qualitative Analyse von Diskussionsbeiträgen in Programmierforen zeigt, welche Vorteile und Einschränkungen Nutzerinnen und Nutzer von GitHub Copilot wahrnehmen. Zu den häufig genannten positiven Aspekten zählen die nützliche Code-Generierung, eine signifikante Zeitersparnis sowie eine insgesamt positive Nutzererfahrung. Darüber hinaus werden eine verbesserte Codequalität und die Fähigkeit zur Anpassung an das individuelle Programmierverhalten sowie eine hohe Kompatibilität mit verschiedenen Editoren hervorgehoben. Auf der anderen Seite werden als zentrale Herausforderungen und Limitationen unter anderem die Generierung fehlerhaften Codes, Bedenken hinsichtlich Datenschutz und Sicherheit, mangelnde Nachvollziehbarkeit der Vorschläge sowie Schwierigkeiten bei der Integration und beim Zugang zum Tool thematisiert (Zhang et al., 2023). Einer Studie, die quantitative und qualitative Erhebungen miteinander verbindet (Mixed Methods), zufolge werden KI-Programmierassistenten vor allem dann gerne genutzt, wenn sie konkrete, effizienzsteigernde Unterstützung bieten, also Tastaturarbeit reduzieren, Aufgaben schneller erledigt werden oder diese das Syntaxwissen auffrischen. Weniger hilfreich werden die Tools beim kreativen Brainstorming von Lösungsansätzen empfunden (Liang et al., 2024). Eine weitere Studie, welche sich mit Produktivitätssteigerungen durch KI in der Softwareentwicklung beschäftigt, zeigte, dass die Entwicklerinnen und Entwickler bei der Nutzung von KI sogar rund 19 % mehr Zeit für ihre Aufgaben benötigten. Besonders paradox ist dabei, dass diese gleichzeitig überzeugt waren, dank der KI schneller geworden zu sein: Sie hatten vorab eine Beschleunigung erwartet und schätzten auch nach Abschluss der Aufgaben, die KI habe ihnen Zeit gespart (Becker et al., 2025).

Eine Studie von Schmitt et al. (2024) zeigt, dass Entwicklerinnen und Entwickler sowohl im *Junior* als auch im *Senior*¹³ Bereich generative KI entlang der drei Bedürfnisse der

¹³ Die Bezeichnungen *Junior* und *Senior* werden im Bereich der Softwareentwicklung informell verwendet, um unterschiedliche Erfahrungsniveaus und Rollenprofile zu kennzeichnen. Junior Entwicklerinnen und Entwickler verfügen in der Regel über begrenzte Berufserfahrung und arbeiten häufiger unter Anleitung, insbesondere bei Architekturentscheidungen, komplexen Problemlösungen oder der Priorisierung von Aufgaben. Senior Entwicklerinnen und Entwickler zeichnen sich hingegen durch umfangreiche Berufserfahrung aus und übernehmen typischerweise Verantwortung für konzeptionelle Entscheidungen, Code Reviews, die Strukturierung von Entwicklungsprozessen sowie die fachliche Unterstützung weniger erfahrener

Selbstbestimmungstheorie (*Kompetenz, Autonomie und Verbundenheit*) sehr unterschiedlich wahrnehmen und bewerten. „Junior Engineers“ erleben Generative KI zwar als effizienzsteigernd, fürchten jedoch, dass eine zu starke Nutzung ihre Kompetenzentwicklung und damit ihren beruflichen Aufstieg beeinträchtigt. Sie begrenzen den Einsatz daher bewusst, um ein „echtes“ fachliches Fundament aufzubauen. „Senior Engineers“ hingegen sehen ihre Kompetenz weniger bedroht und interpretieren generative KI überwiegend als Werkzeug mit begrenztem Nutzen für komplexe Aufgaben. Ihre zentrale Sorge betrifft vielmehr die Verantwortung für Codequalität und die Notwendigkeit, den KI-Einsatz der Juniors zu überwachen, was zugleich ihre Autonomie und Führungsrolle bestätigt. Juniors befürchten, durch unterschiedliche Arbeitsweisen den Anschluss an die Seniors zu verlieren, während Seniors einen möglicherweise unkritischen KI-Gebrauch der Juniors als Gefahr für gemeinsame Standards wahrnehmen (Schmitt et al., 2024).

Vor dem Hintergrund dieser Unterschiede wird ein Blick auf die funktionalen Fähigkeiten generativer KI besonders aufschlussreich. Solche Systeme sind inzwischen in der Lage, anwendungsfähigen Code allein auf Basis natürlicher Sprache zu erzeugen. Systeme wie GitHub Copilot, Amazon CodeWhisperer oder DeepMind AlphaCode übernehmen bereits eigenständig komplexe Synthese- und Transformationsprozesse. Dadurch wird die Schwelle zur Erstellung funktionaler Software deutlich gesenkt: Personen ohne jegliche formale Programmierkenntnisse können mithilfe solcher Software bereits funktionsfähige Anwendungen entwickeln, ohne die dahinterliegenden Programmiersprachen oder Softwarearchitekturen zu verstehen. Die Transformation natürlicher Sprache in Quellcode bringt jedoch auch inhärente Herausforderungen mit sich. Eine empirische Untersuchung von Mastropaolo et al. (2023) zeigt, dass etwa 46 % der generierten Code-Vorschläge von GitHub Copilot variieren, wenn die sprachliche Beschreibung der Aufgabe verändert wird, selbst bei gleichbleibender inhaltlicher Intention. Darüber hinaus hat die Formulierung der Eingabe einen messbaren Einfluss auf die Korrektheit des resultierenden Codes. Diese Befunde werfen Zweifel an der Robustheit solcher Systeme auf und verdeutlichen die zentrale Rolle, die die Qualität und Präzision der sprachlichen Eingaben für die Verlässlichkeit und Konsistenz der generierten Ergebnisse spielt (Mastropaolo et al., 2023). Zudem hat auch die Generierungshäufigkeit einen wesentlichen Einfluss auf die Erfolgsquote. In vielen Fällen sind mehrere Iterationen erforderlich, bis ein funktionierender und qualitativ hochwertiger Code erzeugt wird (Chen et al., 2021).

Teammitglieder. Die konkreten Anforderungen und Aufgabenbereiche sind jedoch nicht standardisiert und variieren je nach Organisation, Projekt und technologischem Kontext.

Aus diesen Befunden lässt sich schließen, dass die Nutzung generativer KI-Systeme zwar grundsätzlich ermöglicht, Quellcode allein durch natürliche Sprache zu erstellen, die erfolgreiche Implementierung jedoch nach wie vor erhebliche Schwierigkeiten birgt. Die starken inhaltlichen und qualitativen Unterschiede zwischen den generierten Codevarianten machen deutlich, dass ohne ein grundlegendes Verständnis von Programmierlogik und Softwarestrukturen die Qualität, Funktionalität und Fehleranfälligkeit des Ergebnisses kaum zuverlässig beurteilt werden können. Somit eröffnet die Technologie zwar neue Zugänge zur Softwareentwicklung, ersetzt aber nicht die Notwendigkeit technischer Kompetenzen, um die erzeugten Ergebnisse kritisch zu bewerten und gezielt zu verbessern.

4 Methodischer Zugang

Ausgangspunkt dieser Arbeit ist das Interesse daran zu verstehen, wie der Einsatz generativer KI in der Softwareentwicklung aus der Perspektive von Entwicklerinnen und Entwicklern beschrieben, bewertet und ausgehandelt wird. Wie bereits in Kapitel 2.2 (digitale Resonanzräume) erwähnt, ist es in der Softwareentwicklung gängige Praxis, sich in Online-Foren über inhaltliche Lösungen für konkrete technische Probleme auszutauschen. Oft ist dabei der Austausch fachlicher Informationen zwar zentral, dieser lässt sich allerdings nie ausschließlich auf der sachlichen Ebene interpretieren. Mit jeder Frage und Antwort werden auch implizit soziale Aushandlungsprozesse vollzogen: Es wird innerhalb der Interaktion verhandelt, wer sich zu welchem Thema äußern darf, wessen Expertise anerkannt wird und wer in welchen Bereichen Deutungshoheit beanspruchen kann. In diesen digitalen Räumen werden also auch nicht direkt fachliche Themen weitgehend implizit mitdiskutiert. Beispielsweise werden arbeitspraktische Fragen zur Integration von generativer KI thematisiert und sich darüber ausgetauscht, wie sich Arbeitsaufwand, Komplexität oder Effizienz einzelner Aufgaben durch den Einsatz von KI verschieben und in welchen Bereichen sich diese sinnvoll einsetzen lässt. Gleichzeitig treten in den Fragen implizit auch grundlegende, berufsethische Fragen in den Vordergrund, die den professionellen Umgang mit dieser Technologie betreffen. Diskutiert wird beispielsweise, in welcher Weise die Nutzung von generativer KI die Verantwortung und Zuständigkeiten im Entwicklungsprozess verändert oder welche Rolle und welche Handlungsträgerschaft der KI im Arbeitskontext zugeschrieben werden sollten. Damit sind die Online-Diskurse nicht nur eine Quelle für technisches Erfahrungswissen, sondern zugleich ein Ort, an dem die Berufspraxis, professionelle Identität und Ethik der Softwareentwicklung im Lichte neuer technologischer Möglichkeiten aktiv konstruiert und ausgehandelt werden. Genau diese Verflechtung von inhaltlichem Wissensaustausch und impliziter Identitäts- und Rollenverhandlung macht Programmierforen zu einer zentralen Quelle für Aussagen zu der hier formulierten Frage.

Um diese Aushandlungsprozesse systematisch zu interpretieren, ist es notwendig, methodisch direkt in das Feld einzutreten, in dem sie stattfinden. Die sogenannte *Netnographie* nach Kozinets (2015) überträgt zentrale ethnographische Grundprinzipien wie teilnehmende Beobachtung, Kontextsensitivität oder Reflexivität der Forschenden (P. Atkinson et al., 2001; Knoblauch, 2014) auf digital vernetzte Kommunikationsräume. Als methodischer Zugang erlaubt diese noch junge Forschungsmethode die systematische Analyse von Interaktionen, Diskursen und kollektiven Sinnbildungsprozessen innerhalb virtueller Gemeinschaften und Online-Plattformen.

Ein weiterer zentraler Vorteil der Methodik besteht darin, dass die Datenerhebung während der *Immersionsphase* nicht in die bestehenden Kommunikationsprozesse eingreift.¹⁴ Dies bedeutet jedoch nicht, dass die Vorgehensweise keinen Einfluss auf den Erkenntnisprozess hat. Die *Immersion* in das Feld vollzieht sich vielmehr auf einer reflexiven Ebene: Die Netnographie sieht vor, dass der Forschende das eigene Verhältnis zum Untersuchungsfeld, die eigenen Vorannahmen sowie die Wirkung der digitalen Umgebung auf die eigene Wahrnehmung kritisch reflektiert und dokumentiert (Kozinets, 2019)¹⁵.

Der Datenzugang begann im Februar 2025 mit der nach Kozinets (2019) vorgesehenen *Investigationsphase*, bei der zunächst verschiedene der bekanntesten Foren und Plattformen in den Blick genommen wurden, welche im Bereich der Softwareentwicklung für den berufsspezifischen Austausch genutzt werden (*11 Best Programming Forums*, 2025). Dabei zeigte sich, dass sich die Plattformen hinsichtlich ihrer Kommunikationskultur und der dort produzierten Inhalte deutlich unterscheiden. So sind die meisten exklusiv für Programmierfragen vorgesehenen Foren in ihrer Plattformarchitektur stark darauf ausgerichtet, lediglich relevante technische Fragen zu stellen und abschweifenden Themen weniger Sichtbarkeit einzuräumen. Diese Form der Interaktion ist stark auf formale Wissensvermittlung ausgerichtet, während subjektive Erfahrungsdimensionen oder narrative Beschreibungen des beruflichen Alltags vergleichsweise kaum Raum erhalten. Das Forum Reddit ist dagegen nicht ausschließlich für IT-Themen konzipiert. Stattdessen gliedert sich die Plattform in eine Vielzahl von themenspezifischen Unterforen, sogenannten *Subreddits*, die jeweils eigene Communitys und Kommunikationskulturen ausbilden.

Ein Teil dieser Subreddits ist thematisch breit angelegt und dient dem allgemeinen Diskurs über Programmierung, technologische Entwicklungen und aktuelle Trends. In solchen Bereichen wie *r/programming* oder *r/coding* werden vor allem Artikel, Meinungsbeiträge und Erfahrungsberichte geteilt, die einen übergreifenden Blick auf Softwareentwicklung als Praxis und Berufsfeld ermöglichen. Daneben gibt es Subreddits mit einem starken Lern- und Unterstützungsfokus, welche

¹⁴ Kozinets sieht im Rahmen der Netnographie neben der *Immersionsphase* auch eine *Interaktionsphase* vor, in der Forschende entweder offen oder verdeckt aktiv an den Diskussionen teilnehmen, eigene Beiträge verfassen oder gezielte Fragen stellen können. Diese Phase ist jedoch ausdrücklich optional und dient primär der zusätzlichen Verdichtung der zuvor immersiv erhobenen Daten (Kozinets, 2019). Ich habe mich gegen eine aktive Interaktion mit dem Feld entschieden, da diese potenziell zu einer Verzerrung der Daten führen, die Interaktionsdynamiken innerhalb der Foren beeinflussen und den Fokus der Analyse von beobachteten Praktiken hin zu forschungsinduzierten Interaktionen verschieben könnte.

¹⁵ Die in der Netnographie vorgesehene *Immersion* entspricht damit in ihren Grundzügen dem ethnographischen Ideal der „reflexiven Subjektivität“, wonach Forschende ihre eigene Position, Wahrnehmung und Beteiligung am Erkenntnisprozess systematisch reflektieren (Clifford & Marcus, 2023).

sich an Personen richten, die sich neue Technologien aneignen möchten. In Subreddits wie *r/learnprogramming* stehen konkrete Fragen, Problemlösungen und didaktische Erklärungen im Vordergrund. Ergänzt werden diese Bereiche durch technologie- oder domänenspezifische Subreddits, die sich auf einzelne Programmiersprachen, *Frameworks* oder Arbeitsfelder konzentrieren und jeweils nach diesen benannt sind. In anderen Subreddits wie *r/cscareerquestions* oder *r/ExperiencedDevs* werden Fragen zu Karriereverläufen, Arbeitsbedingungen und Organisationsstrukturen diskutiert. Hier findet man teilweise auch Aushandlungen bezüglich der professionellen Identität sowie performative Abgrenzungen innerhalb des Feldes der Softwareentwicklung. Schließlich existieren Subreddits, die sich explizit mit Künstlicher Intelligenz und Automatisierung befassen und damit eine Schnittstelle zwischen Softwareentwicklung und generativer KI darstellen. In Subreddits wie *r/MachineLearning*, *r/artificial* oder *r/ChatGPT* werden sowohl technische Fragen als auch reflexive Diskussionen über die Auswirkungen von KI auf die Arbeit, Kompetenzanforderungen und berufliche Rollenverständnisse geführt.

In diesen Subreddits konnte eine besonders hohe Dichte an subjektiv-narrativen Beschreibungen des Programmieralltags gefunden werden. Nach einer ersten Sichtung verschiedener Online-Foren und der anschließenden Auseinandersetzung mit programmierspezifischen Subreddits habe ich mich dafür entschieden, ausschließlich Reddit als Untersuchungsplattform für die Erhebung zu wählen. Ausschlaggebend war, dass sich hier im Vergleich die meisten narrativen und subjektiven Beschreibungen des Programmieralltags finden. Beitragende berichten nicht nur über technische Probleme, sondern erzählen ausführlich von ihren Arbeitsweisen, ihren Erfahrungen mit neuen Technologien und ihren beruflichen Selbstverständnissen. Da diese Formen der Darstellung in unterschiedlichen Subreddits in ähnlicher Weise vorkommen, wurde die Datensuche nicht auf einzelne Subreddits beschränkt, sondern auf Reddit insgesamt durchgeführt.

Aufgrund der Tatsache, dass die überwiegende Mehrheit der relevanten Online-Foren ohnehin in englischer Sprache geführt wird und die Kommunikation in der Entwickler-Community generell stark englischsprachig geprägt ist, wurde die Datenerhebung ebenfalls auf Englisch durchgeführt. Englisch fungiert in der Softwareentwicklung als zentrale Arbeitssprache, wodurch Diskussionen und Erfahrungsberichte international zugänglich sind und sich nicht auf lokale oder territoriale Kontexte beschränken. Eine englischsprachige Erhebung ermöglicht daher einen breiteren Bezugsrahmen und trägt dem Umstand Rechnung, dass Programmiererinnen und Programmierer in global vernetzten Arbeits- und Diskursräumen agieren. Mittels englischer Suchbegriffe wie „programming with ChatGPT“ oder „programming with AI“ wurden einschlägige Diskussionsstränge plattformübergreifend auf Reddit, also unabhängig vom Subreddit, gesucht. Ein Diskussionsstrang auf Reddit besteht aus einem ursprünglichen Beitrag eines Nutzers sowie allen

darauffolgenden Kommentaren und Antworten, die in einer hierarchischen, baumartig verzweigten Struktur organisiert sind. Nutzer können sowohl auf den ursprünglichen Beitrag als auch auf einzelne Kommentare antworten, wodurch verschachtelte Diskussionsverläufe entstehen. Diese Diskussionstränge werden im weiteren Verlauf der Arbeit als „Thread“ bezeichnet.

Aus den Suchergebnissen habe ich anschließend erste für die Forschungsfrage relevante Threads identifiziert und diese dann inklusive aller zugehörigen Antworten rezipiert, um den Bedeutungszusammenhang der Diskussion zu erfassen. Alle für die Forschungsfrage relevanten Beiträge wurden anschließend in die qualitative Analysesoftware MAXQDA 24 überführt, wobei die originale Diskussionsstruktur des Forums erhalten blieb. Dadurch bleibt sichtbar, wie einzelne Aussagen aufeinander Bezug nehmen und in welchem argumentativen Verlauf sie stehen. In der Analysesoftware wurden die einzelnen Aussagen der jeweiligen Beiträge dann im Kontext ihrer jeweiligen Stellung innerhalb des Threads als Segment kodiert, sodass die Analyse nicht auf isolierten Aussagen beruht, sondern innerhalb eines Forumseintrags stets die Einbettung in den fortlaufenden Diskurs berücksichtigt. Im Sinne einer explorativ-offenen, induktiven Vorgehensweise wurden zunächst Codes an allen gefundenen subjektiven Aussagen zur Programmierarbeit mit generativer KI vergeben, ohne dass ein vorab festgelegtes Kategoriensystem zugrunde lag. Dabei habe ich mich an meiner Forschungsfrage orientiert und all jene Aussagen kodiert die zur Beantwortung beitragen, etwa indem sie Deutungen, Beschreibungen und normative Aushandlungen beinhalten. Ziel war es, die Sichtweisen der Beitragenden möglichst nah am Material herauszuarbeiten und als einzelne Kodierungen festzuhalten.

Im Verlauf dieses Prozesses kristallisierten sich sukzessive drei zentrale Dimensionen von kodierten Aussagen heraus:

1. Veränderung der Arbeit selbst: Hierunter fallen Beiträge, die thematisieren, ob und inwiefern sich Umfang, Schwierigkeit oder Anstrengungsgrad der Arbeit durch den Einsatz von KI verändern. Dazu gehören z. B. Aussagen dazu, ob die Arbeit quantitativ mehr oder weniger wird, ob sie als komplexer oder einfacher empfunden wird oder wie sich der Arbeitsalltag und die alltäglichen Routinen insgesamt wandeln.
2. Einfluss auf die eigene Identität: Diese Kategorie umfasst Aussagen, in denen Entwicklerinnen und Entwickler über eine Veränderung ihres beruflichen Selbstverständnisses reflektieren. Dies betrifft etwa die Wahrnehmung neuer Tätigkeitsprofile, Verschiebungen der eigenen Expertise oder die Frage, welche Rolle man im Entwicklungsprozess innehat.
3. Rollenzuschreibung an die KI: In dieser Dimension werden Zuschreibungen thematisiert, die Nutzende der KI gegenüber vornehmen. So wird die KI teils als „Assistenz“ oder „Werkzeug“

beschrieben, teils aber auch als „Kollege“ oder als quasi eigenständiger Akteur im Entwicklungsprozess.

Wichtig ist hervorzuheben, dass diese Dimensionen nicht a priori festgelegt, sondern aus dem Material selbst induktiv entwickelt wurden. Sie stellten sich im Zuge der Kodierung zunehmend als wiederkehrende Bezugsgruppen der jeweiligen Aussagen heraus und wurden daraufhin zu analytischen Hauptkategorien verdichtet. Jede kodierte Aussage lässt sich in mindestens eine der drei Dimensionen eindeutig zuordnen. Da die drei Dimensionen lediglich als übergeordnete Deutungsebene dienen, können die einzelnen kodierten Aussagen durchaus in mehreren Dimensionen zugleich auftreten. So lässt sich beispielsweise die Aussage, dass mit der Implementierung von KI vermehrt müßige Arbeit wie das Fehlerkorrigieren anfällt, als Hinweis auf zusätzliche Belastung in der Dimension Arbeit interpretieren. Gleichzeitig kann dies jedoch auch als eine Verschiebung beruflicher Tätigkeiten und damit als eine Veränderung des beruflichen Selbstverständnisses gedeutet werden.

Nachdem die drei Dimensionen gefasst waren, wurden diese verdichtet. Dazu wurden gezielt weitere Beiträge identifiziert, die spezifisch zu den bereits entwickelten Dimensionen weitere Einsichten liefern konnten. Hierfür kamen verfeinerte Schlagwortsuchen zum Einsatz, etwa die Suche nach „role AI“ im Kontext der Dimension Rollenzuschreibung. Auf diese Weise konnte das Datenmaterial entlang der emergenten Kategorien systematisch erweitert und verdichtet werden. Im Untersuchungszeitraum vom 01.03.2025 bis zum 30.04.2025 wurden insgesamt acht für die Forschungsfrage relevante Forenbeiträge ausgewählt, welche jeweils zwischen 46 und 973 Antworten umfassten. Sämtliche Antworten, insgesamt 2.601 Beiträge, wurden vollständig rezipiert und in die Analyse einbezogen. Auf dieser Grundlage wurden am Ende der Erhebungsphase 275 für die Forschungsfrage relevante Textsegmente identifiziert und kodiert.

Aufbauend auf den nun verdichteten Daten folgte in einem zweiten Schritt eine zentrale, strukturierende Auswertung des Datenmaterials. Ziel dieses Auswertungsschrittes war es, die Vielzahl einzelner, zunächst isoliert stehender Codes in übergeordnete Bedeutungszusammenhänge zu überführen und auf diese Weise eine systematische Übersicht der innerhalb der jeweiligen Dimensionen vertretenen Positionen zu gewinnen. Zu diesem Zweck wurden thematisch verwandte Kodierungen zu übergeordneten Typen gebündelt. Diese Typen fassen wiederkehrende Argumentationslinien, Deutungsmuster und Bewertungslogiken zusammen und repräsentieren jeweils eigenständige, in sich konsistente Perspektiven auf den Einsatz Künstlicher Intelligenz im Programmierkontext. Jeder Typ ist dabei einer der drei analytischen Dimensionen Arbeit, Identität oder Rolle zugeordnet und bildet innerhalb dieser Dimension eine spezifische Ausprägung des

Umgangs mit und der Deutung von Künstlicher Intelligenz ab. Die Typenbildung verfolgt keinen erklärenden oder normativen Anspruch, sondern dient einer primär deskriptiven Kartierung der im Datenmaterial artikulierten Positionen. Die Typen werden in Kapitel 5 anhand konkreter Zitate ausführlich beschrieben.

Die gewählte Methode bringt mehrere Einschränkungen mit sich, welche bei der Interpretation der Ergebnisse zu berücksichtigen sind. Zunächst ist die Gruppe der Personen, die auf Reddit Beiträge verfassen, stark selektiv. Es handelt sich nicht um eine zufällige oder repräsentative Stichprobe von professionellen Softwareentwicklerinnen und -entwicklern, sondern um eine selbstgewählte Gruppe von Personen, die bereit sind, ihre Erfahrungen öffentlich zu teilen. Darüber hinaus kann die Identität der Verfassenden nicht überprüft werden. Es lässt sich daher nicht mit Sicherheit feststellen, ob es sich tatsächlich um Personen handelt, die beruflich im Bereich der Softwareentwicklung tätig sind, oder ob sie sich beispielsweise lediglich in ihrer Freizeit mit diesem Thema beschäftigen. Auch der demografische Hintergrund der Beitragenden bleibt unbekannt. Faktoren wie Berufserfahrung, Ausbildung, Alter, Geschlecht oder geografischer Standort können nicht erfasst werden und lassen sich daher nicht in die Interpretation der Aussagen einbeziehen. Darüber hinaus lässt die Erhebungsmethode nicht zu, dass inhaltliche Bezüge zwischen den einzelnen kodierten Segmenten gezogen werden. So kann lediglich eine oberflächliche Kartierung verschiedener Aussagen vorgenommen werden, ohne prüfen zu können, ob bestimmte Positionen typischerweise gemeinsam auftreten oder ob einzelne Diskutanten konsistent bestimmte Sichtweisen vertreten. Die fehlende Zuordenbarkeit verhindert somit die Identifikation möglicher Zusammenhänge zwischen Aussagen und begrenzt die Tiefe der Analyse erheblich.

Auch die händisch ausgewählten Threads sind nicht repräsentativ für die Fülle der Themen, die auf Reddit diskutiert werden. Die Auswahl erfolgte nicht systematisch, sondern orientierte sich an den von Reddit aufgrund ihrer Beliebtheit oder der Interaktionshäufigkeit vorgeschlagenen Inhalten. Dadurch werden vor allem stark diskutierte oder besonders sichtbare Beiträge berücksichtigt, während weniger beachtete Threads unberücksichtigt bleiben. Es zeigten sich in den erhobenen Daten deutliche Unterschiede in der Häufigkeit, mit der kodierte Aussagen genannt wurden. Diese quantitativen Differenzen wurden jedoch bewusst nicht interpretiert, um keine irreführenden Aussagen über die Verbreitung bestimmter Erfahrungen zu treffen. Angesichts der selektiven Stichprobe und der Selbstselektionsmechanismen, die mit freiwilligen Online-Äußerungen einhergehen, wäre eine solche Bewertung methodisch nicht gerechtfertigt. Ebenso kann nicht ausgeschlossen werden, dass einzelne Beiträge bewusst verzerrt oder provokativ formuliert sind, um Reaktionen hervorzurufen oder Diskussionen anzuheizen. Solche sogenannten Troll-

Inhalte¹⁶ sind auf Reddit durchaus verbreitet (Brubaker et al., 2021; Cheng et al., 2017; Datta & Adar, 2019; Saeed et al., 2022). Sie können den Diskussionsverlauf maßgeblich beeinflussen und stellen eine weitere Verzerrungsquelle dar.

Trotz der genannten Einschränkungen erweist sich die Methode der Netnographie als geeignet, um eine erste, oberflächliche Kartierung unterschiedlicher in der Softwareentwicklung vertretener Positionen vorzunehmen. Ziel der Untersuchung ist nicht eine repräsentative Abbildung der Entwickler-Community, sondern das Sichtbarmachen verschiedener Perspektiven, Deutungen und Erfahrungsweisen, die in öffentlichen Online-Diskussionen artikuliert werden. Netnographische Ansätze ermöglichen es, solche Diskurse in ihrem natürlichen Entstehungskontext zu erfassen und typische Argumentationsmuster, Spannungsfelder und wiederkehrende Themen herauszuarbeiten. Diverse Studien zeigen übereinstimmend, dass Menschen in digitalen Kommunikationsumgebungen offener und weniger gehemmt kommunizieren, da Anonymität, reduzierte soziale Hinweisreize und räumliche Distanz soziale Kontrollmechanismen abschwächen (Lapidot-Lefler & Barak, 2012; Suler, 2004). Gerade durch die Analyse digitaler Diskussionen lassen sich unterschiedliche Positionen nebeneinanderstellen und vergleichend betrachten, ohne den Anspruch zu erheben, deren Häufigkeit oder Verbreitung quantitativ zu bewerten. In diesem Sinne eignet sich die Netnographie besonders, um die Vielfalt der in der Softwareentwicklung vertretenen Sichtweisen aufzuzeigen und zentrale Diskurslinien sichtbar zu machen. Die Methode ermöglicht somit eine explorative Annäherung an das Forschungsfeld und bildet eine sinnvolle Grundlage für weiterführende, vertiefende Untersuchungen.

¹⁶ Als Trollen wird das absichtliche Verfassen provokativer, irreführender oder stark zugespitzter Beiträge bezeichnet, mit dem Ziel, emotionale Reaktionen auszulösen oder Diskussionen gezielt zu beeinflussen. Diese Praxis ist auf Online-Plattformen wie Reddit weit verbreitet und Teil der dortigen Diskussionskultur.

5 Perspektiven auf Softwareentwicklung mit KI

Der folgende Empirieteil präsentiert die Ergebnisse der qualitativen Analyse und der darauf aufbauenden Typenbildung. Einen ersten Überblick bietet Tabelle 2. Die Tabelle gliedert sich in drei Spalten: Die mittlere Spalte enthält die vergebenen Kodierungen, die aus Gründen der semantischen Nähe zu den englischsprachigen Originaldaten in englischer Sprache geführt werden. Die linke Spalte ordnet diese den analytischen Dimensionen Arbeit, Identität und Rolle zu, während die rechte Spalte die daraus hervorgehende Typenbildung abbildet.

Tabelle 2: Zuordnung der Kodierungen zu Dimensionen und Typen

Dimension	Kodierungen	Typus
ARBEIT	ai automates (simple) tasks ai enables higher quality of results ai use requires extra skill ai generates more bugs ai spread raises skill requirements ai spread raises work demand	Produktivitätsgewinne
		Anforderungszuwachs
IDENTITÄT	enhances skills shift to other activities only seniors should use ai ai helps beginners to learn	Kompetenzentwicklung
	loss of skills, dependancy knowledge monopoly shame of use maintain job without skills	Deprofessionalisierung
	no use/need ai can't replace a good programmer	Statusbewahrung
	fear of loosing job we have to adapt to ai / use is necessary	Anpassungsdruck
ROLLENZUSCHREIBUNG	ai can only do simple code ai takes boring, repetitive tasks user controls the output of ai	KI als Werkzeug
	use as an assistant ai completes my thoughts ai needs to get a strong direction	KI als Assistenz
	ai as a mentor ai sets the flow ai doesn't judge me ai is wrong confidently	KI als Strukturierungsinstanz

Quelle: Eigene Darstellung

5.1 Dimension Arbeit

Diese Dimension erfasst Beschreibungen dazu, ob und wie sich die konkreten Tätigkeiten im Bereich der Softwareentwicklung durch den Einsatz von KI verändern, etwa hinsichtlich ihrer Menge, Komplexität oder Effizienz. Während der explorativen Erhebung wurden insgesamt sechs verschiedene Aussagen kodiert, welche sich nachträglich in zwei gegensätzliche Typen unterscheiden lassen: Auf der einen Seite stehen Produktivitätsgewinne und auf der anderen Seite ein Anforderungswandel.

5.1.1 Produktivitätsgewinne

Zentrale Aussagen:

- „ai automates (simple) tasks“
- „ai enables higher quality of results“

Beschreibung:

„They're just using AI to reduce the time“ (Thread 8: 48), beschreibt ein User seine Kollegen, und bringt dabei einen zentralen Nutzungsgrund auf dem Punkt. Die wahrgenommene Zeitersparnis wird insbesondere darauf zurückgeführt, dass KI-Tools einfache, repetitive oder administrative Aufgaben, welche ansonsten manuell und zeitaufwendig auszuführen wären, vollständig übernehmen können. Es entsteht also eine Automatisierung einfacher Tätigkeiten, was aus exemplarischen Kommentaren wie den Folgenden hervorgeht: „Personally it saves me time by doing some of the mundane stuff (Thread 7: 49)“, oder „Fuck yes. It saves me so much time on bullshit work“ (Thread 6: 65–66).

Die dadurch von den Entwicklerinnen und Entwicklern hier als gewonnen wahrgenommene Zeit kann sich direkt in weiteren Produktivitätsgewinnen bei ihren Tätigkeiten bemerkbar machen. Gleichzeitig steigt dabei auch der Bedarf, was sich relativierend auf den Effekt auswirken kann, am Ende liegt der Fokus hier aber auf dem Produktivitätsgewinn.

„If an LLM can make me twice as productive as I am now, bring it on. It's not like there's a shortage of work to be done, and if I can do twice as much work, I'm quite sure that my customer can think up twice as many things that they'd like to build“ (Thread 4: 63).

Infolge dieser durch KI erzielten Zeitersparnis kommt es, wie andere User berichten, auch zu einer qualitativen Verschiebung der Tätigkeiten. Indem routinisierte und vergleichsweise einfache Aufgaben an KI-Systeme delegiert werden, entsteht für die Entwicklerinnen und Entwickler Freiraum, sich stärker auf komplexere, kognitiv anspruchsvollere Aspekte ihrer Arbeit zu konzentrieren, wie folgende Aussage verdeutlicht: „I find it’s useful for completing basic tasks and that alone saves enough time for me wracking my brain to get the idea out of my head and into the editor. I also don’t have to search high and low for answers” (Thread 6: 29). Aus demselben Grund zählt die KI, wie ein anderer User beschreibt mittlerweile sogar schon in die Kategorie „Fixkosten“. Die Aussage macht deutlich, dass KI für ihn zu einer selbstverständlichen Voraussetzung professioneller Arbeit geworden ist: Auf den Einsatz der Technologie könne er nicht mehr verzichten:

“I would say I am now more productive, coding faster. That extra time I use for evaluating the code more often, which in effect results in better code. I would not want to code without it anymore at this point, so it's now one of my fixed costs” (Thread 6: 51).

Außerdem verweisen einige User auch auf eine technische Erweiterung der Handlungsmöglichkeiten im Sinne einer Augmentation. Durch die Unterstützung der KI sind Nutzer in der Lage, über ihre bisherigen Fähigkeiten hinauszugehen, individuell angepasste Lösungen zu entwickeln und ihre Arbeitsergebnisse inhaltlich insgesamt zu verbessern. Diese Form von Effizienz ist also nicht nur mengen- oder zeitorientiert, sondern beinhaltet auch eine mögliche qualitative Verbesserung der Ergebnisse:

„Productivity will be up, mistakes down. The more you can remove error from your code through the use of tools like AI/copilot, the less time you’ll spend later debugging some problem you could have entirely avoided in the first place. This will result in more value/better outcomes“ (Thread 6: 31).

Zusammenfassend weisen die User darauf hin, dass die Nutzung von KI direkt durch Automatisierung oder indirekt mittels Augmentierung zu Produktivitätsgewinnen führt.

5.1.2 Anforderungszuwachs

Zentrale Aussagen:

- „ai use requires extra skill“

- „ai generates more bugs“
- „ai spread raises skill requirements“
- „ai spread raises work demand“

Beschreibung:

Im genauen Gegensatz zu den Aussagen, die KI als Produktivitätsgewinn ansehen, wird hier die Integration von KI als eine Quelle neuer Komplexitäten und Belastungen beschrieben. Die Arbeit wird nicht (nur) einfacher, produktiver und schneller, sondern anspruchsvoller. Zunächst benötigt man für den sinnvollen Einsatz von KI-Systemen zusätzliche Fähigkeiten, etwa die Kompetenz, gute und präzise Ergebnisvorstellungen zu formulieren, um mit möglichst wenig Nachbearbeitung den gewünschten Inhalt zielgerichtet zu generieren. “It saves time typing but you need to get a good understanding about the concepts/codes it provides otherwise you will just be stuck debugging for hours” (Thread 5: 51). Nachdem die Inhalte generiert wurden, ist es dann zusätzlich noch notwendig, fundierte und programmierbezogene Fachkenntnisse einzubringen, um die generierten Inhalte zu selektieren: “You still need a skilled programmer to know what is good, what will work, what is a pitfall, what is a bug, and what is an AI hallucination” (Thread 2: 16).

Besonders häufig wurde auf Reddit ein weiteres Symptom der Implementierung von KI thematisiert, welches die Arbeit komplizierter macht: die Fehleranfälligkeit der KI-Systeme. Jeder einzelne Fehler muss von einer verantwortlichen Person zunächst manuell gefunden und anschließend auch korrigiert werden, was verschiedene Probleme mit sich bringt:

„Let's imagine those models will get a lot better. They will be able to write bigger parts of the software by themselves. What now? Someone has to proof read it, unless they are 100% correct and the company providing the model will take responsibility for bugs. I think reading thousands of lines of code still takes a huge amount of time and of course you need someone who understands the code. And understanding code you haven't written yourself is harder cause you might have written it in a completely other way with a different vocabulary etc. Beside that, when you write code you got a total different insight. You kinda have a stack of functions and values inside your head, an understanding of the data flow. It's hard to get that good of an understanding by just reading someone else's code. Surely if 90% is correct all the time a lot of people will just fly-over the code and scan it but don't "debug" it in there head, to catch nasty bugs” (Thread 4: 75-76).

Es ist also nötig, jedes Mal neu zuerst den Syntax des KI-generierten Codes zu verstehen, um Fehler zu finden, was deutlich schwieriger und müßiger ist, als seinen selbstgeschriebenen Code nachzuvollziehen. Deshalb zieht beispielsweise ein Reddit-User die Konsequenz, dass die Fehler der KI die Produktivität am Ende des Tages sogar behindern: „On my side, it is not increasing the productivity and actually it is dropping because the suggestions are leading me in a wrong direction and i find myself continuous deletion [sic] and retyping the code” (Thread 5: 39). Einmal wird die KI aufgrund der Fehler als „distraction“ (Thread 5: 40) benannt, ein anderer Kommentar beschreibt die Interaktion mit der KI, als „fighting with it to find some gold“ (Thread 8: 77-78). Weitere Kommentare suggerieren, dass die Fehleranfälligkeit von KI Systemen auch zusätzliche Kontroll- und Reparaturaufgaben in der Sicherung der kritischen Infrastruktur erforderlich machen: “On the other hand, I'm looking forward to being a big-bucks contractor that gets called in when one of these things introduces a new vulnerability to a critical system” (Thread 2: 12).

Neben den zusätzlichen Kompetenzen, welche für die Bedienung von KI und das Fehlerbeheben erforderlich sind, erhöhen und diversifizieren sich auch die Verantwortlichkeiten. So wird von Usern auf Reddit prognostiziert, dass die Erwartungen, insbesondere auch an weniger ausgebildete Personen, mit dem Einsatz der KI höher werden: „I think the biggest sea change will be that the bar / expectations for your average non-specialist / non-godlike-lead / regular generalist will go way up” (Thread 4: 66). Einige User berichten auch, dass Aufgaben, die zuvor auf mehrere Personen verteilt waren, nun häufig von einer einzelnen Person übernommen werden müssen. Neben der Erwartung, beispielsweise gleich mehrere Programmiersprachen zu beherrschen oder in unterschiedlichen Bereichen des Softwareproduktionsprozesses kompetent zu sein, erweitern sich die geforderten Fähigkeiten teilweise auch auf fachfremde Disziplinen, etwa in Richtung Produktmanagement. So wird zum Beispiel auf Basis der bisherigen Entwicklung folgendes Zukunftsszenario antizipiert:

„The software developer of the future is a mix between programmer and product manager. This person understands both business and technology well enough to understand customer needs and use AI to create solutions. Creation of smarter programming tools has, historically, resulted in merging of roles. It used to be that in the early 60s, the person who wrote the code wasn't the person who entered the code into the computer. This is because computers were complicated enough that there was a specialized operator whose primary job was to operate the computer. After the invention of teletype terminal, and automated scheduling, the programmer became the operator. From the 70s-90s, software developer, QA and operations engineer were distinct roles. This is because testing is a complicated

task that requires specialized knowledge. Similarly, managing production software is a complicated task that requires specialized [sic] knowledge. However, as the processes and tools matured, we got to a point where the 3 roles merged into one” (Thread 4: 49-50).

Gleichzeitig wird auch davon ausgegangen, dass digitale Systeme mit steigender Komplexität auch in weitere Anwendungsfelder expandieren. Mit dieser Ausweitung steigt zugleich der Bedarf an Fachkräften, die solche Systeme entwickeln, betreiben und verantworten. Die User betrachten die wachsenden Anforderungen und Bedarfe daher nicht zwangsläufig als Belastung, sondern teilweise auch als Form von beruflicher Stabilität: Die steigende Komplexität und Ausweitung digitaler Systeme sichern demnach eine anhaltende Nachfrage nach noch mehr Arbeitskraft.

„I don't think it will replace us or be a big threat. [...] It will lead to bigger and more complex software. It will lead to more domains pushing forward digitalization and automation” (Thread 4: 73).

Neben dem Bedarf an zusätzlichen Qualifikationen, die für den sachgerechten Einsatz und die effektive Bedienung von KI-Systemen notwendig werden, verweisen die Beiträge auch auf eine erhöhte Fehleranfälligkeit KI-generierter Inhalte. Diese zieht einen zusätzlichen Arbeitsaufwand nach sich, insbesondere in Form von Prüf-, Korrektur- und Nachbearbeitungstätigkeiten. Darüber hinaus wird KI als Faktor beschrieben, der zu einer Ausweitung interdisziplinärer Kompetenzanforderungen führt. Schließlich wird die Verbreitung von KI auch mit einer quantitativen Expansion des IT-Sektors in Verbindung gebracht, da Effizienzgewinne mit einer steigenden Nachfrage nach technischer Arbeit einhergehen.

5.1.3 Zwischenfazit: Dimension Arbeit

Im Zwischenfazit zur Dimension *Arbeit* zeigt sich ein ambivalentes Wirkungsbild des KI-Einsatzes in der Softwareentwicklung. Einerseits werden deutliche Produktivitätssteigerungen beschrieben, die direkt entstehen, weil KI-Systeme bestimmte Aufgaben automatisieren und dabei teilweise sogar schneller, zuverlässiger oder technisch effizienter sind. Dazu ergeben sich indirekte Produktivitätsgewinne, wenn die Nutzung von KI zeitliche oder kognitive Ressourcen freisetzt, die für zusätzliche oder anspruchsvollere Tätigkeiten genutzt werden können. Andererseits verweisen einige User darauf, dass der Einsatz von KI mit steigenden Anforderungen an Arbeitsprozesse und individuelle Kompetenzen einhergeht. Die Notwendigkeit zur Kontrolle, Korrektur und

Einordnung KI-generierter Ergebnisse sowie die Ausweitung der Aufgabenbereiche einzelner Entwicklerinnen und Entwickler können dazu führen, dass Arbeitsprozesse insgesamt komplexer werden. Dies muss allerdings nicht immer negativ gesehen werden, sondern kann auch als Zeichen einer wachsenden Branche und damit einer sicheren Arbeit gelesen werden.

Beim Vergleich ist besonders interessant, dass die beiden Perspektiven auf Produktivitätsgewinne und steigende Anforderungen nicht getrennt voneinander stehen, sondern ineinandergreifen und sich gegenseitig bedingen. Produktivitätssteigerungen wurden zwar deutlich und positiv benannt, teilweise aber im selben Atemzug aber auch wieder relativiert, da mit der breiten Verfügbarkeit von KI zugleich die Erwartungen an den Output steigen. Der anfängliche Effizienzvorteil einzelner Entwicklerinnen und Entwickler verliert damit an Wirkung, sodass sich die Arbeitsmenge auf einem höheren Niveau einpendelt und Effizienzgewinne unmittelbar in eine Ausweitung der Arbeit überführt werden.

Gleichzeitig wird von Usern, welche einen Anforderungszuwachs ansprechen, die flächendeckende Implementierung von KI auch als Faktor einer quantitativen Expansion der Branche verstanden, die zur Stabilisierung des Arbeitsmarktes beitragen und Ängste vor Arbeitsplatzverlust mindern kann. KI erscheint in dieser Perspektive weniger als Rationalisierungsinstrument denn als Treiber wachsender Arbeitsnachfrage.

5.2 Dimension Identität

Diese Dimension reflektiert die subjektive Bedeutung von generativer KI für das berufliche Selbstbild. Insgesamt sind dieser Dimension 12 verschiedene Aussagen zugeordnet, welche sich in insgesamt vier Kategorien verdichten lassen: Die erste Kategorie umfasst Aussagen, die eine erlebte Kompetenzentwicklung durch den Einsatz von KI beschreiben. Die zweite Kategorie beinhaltet dagegen Beiträge, in denen ganz im Gegenteil eine Deprofessionalisierung bzw. Gefährdung der professionellen Integrität thematisiert wird. Damit verbunden, aber analytisch als eigene Kategorie zu fassen, ist die in den Reddit-Foren performativ ausgeübte Verteidigung des Berufsethos. Die vierte Kategorie schließlich umfasst Aussagen, in denen ein erlebter Anpassungsdruck artikuliert wird. Im Folgenden werden diese vier Typen erläutert.

5.2.1 Kompetenzentwicklung

Zentrale Aussagen:

- „enhances skills“
- „shift to other activities“

- „only seniors should use ai“
- „ai helps beginners to learn“

Beschreibung:

Aussagen, die dem Typus Kompetenzentwicklung zugeordnet werden, beschreiben KI in erster Linie als Chance zur persönlichen und fachlichen Weiterentwicklung. Dabei wird KI als eine Technologie verstanden, die durch ihre interaktive Nutzung kontinuierliches Lernen unterstützt und zur fortlaufenden Aktualisierung von Fachwissen beiträgt: “That also keeps you educated and fresh, as a nice side effect” (Thread 6: 50). Im Zentrum steht oft die Idee der Augmentation: KI erweitert nicht nur bestehende Fähigkeiten, sondern kann deren Wirkung sogar vervielfachen, wie ein User schreibt: „AI is a multiplier of someone’s potential” (Thread 8: 77).

Andere Aussagen suggerieren, dass es nicht nur zu einer normativen Verbesserung der Fähigkeiten führt, sondern einfach eine Veränderung der Tätigkeitsbereiche bedeutet. Beispielsweise berichten einige User, dass sie seit der Einführung von generativer KI deutlich weniger Zeit damit verbringen, selbst ihren Code zu schreiben, und stattdessen vielmehr Korrekturlesen. „Tools like copilot mean that instead of writing code, we’re reviewing code” (Thread 5: 12). Diese Verschiebung von Kompetenzen wird auf anderer Seite in die genau gegenteilige Richtung beschrieben. Ein anderer User sieht beispielsweise den eigenen Wert ebenfalls explizit von der operativen Ausführung einzelner Tätigkeiten wie dem Schreiben von Code gelöst, verortet ihn stattdessen aber in übergeordneten, integrativen Kompetenzen. Entscheidend scheint hier die Fähigkeit zu sein, geschäftliche Anforderungen zu verstehen, zu übersetzen und in ein konsistentes Produkt zu überführen.

“My value isn’t the code I write or diagrams I draw. My value is the transformation and alignment of business needs to a finalized product. The code is merely one aspect of the final product. I’m OK with an LLM writing that if it’s capable. Currently LLMs only threaten devs that only know how to write (poor) code and nothing else - most likely juniors. Replacing juniors with LLMs will save companies money in the short term, but it’ll cripple their code and products long term as more experienced devs will eventually leave or retire from the company. If anything, I’ll likely have more work in the future to help maintain these future ‘legacy’ applications” (Thread 4: 85-89).

Besonders interessant an den Aussagen ist, dass das Erfahrungslevel der KI-anwendenden Person eine große Rolle in der Diskussion über die Auswirkungen von generativer KI zu spielen scheint. Die Bewertungen dazu fallen dabei sogar sehr gegensätzlich aus: Auf der einen Seite finden sich Aussagen, dass KI nur bei erfahrenen Programmierern und Programmierern bzw. Seniors sinnvolle Anwendung findet. Den Kommentaren zufolge sollten nur solche Tätigkeiten an die KI delegiert werden, die man selbst sicher beherrscht, um Fehler zu vermeiden und wirklich produktiv zu sein. Das Credo lautet hier entsprechend: „You can’t really use AI to get work done unless you know what you are doing or you are working on something super basic” (Thread 8: 39). Der Lerneffekt mit KI ist also an eine gewisse Kenntnisgrundlage geknüpft:

„It's fine as long as you understand the concepts behind the library you're using. If that's the case and Copilot is just helping you learn the syntax then that's great. If you don't understand the concepts, and you're using the Copilot as a crutch to avoid doing so then that's a problem” (Thread 7: 50).

Andere Kommentare legen im Gegenteil nahe, dass insbesondere Personen mit geringer fachlicher Erfahrung von generativer KI profitieren können, da diese bei Tätigkeiten unterstützt, die noch nicht beherrscht werden:

“For me when I'm using my usual languages and tools that I know, I don't use AI at all. But when I'm using a language or something I won't really know [...], the [AI is] really helpful to help me figure out what the heck is going on!” (Thread 8: 46-47).

Insbesondere bei neuen Tätigkeiten wie etwa dem Programmieren in unvertrauten Programmiersprachen werden hier generative KI-Systeme eingesetzt. Dabei geht es nicht nur darum ein vorzeigbares Ergebnis zu produzieren, sondern es wird auch ein Lerneffekt durch das Generieren von Code beschrieben: „*And, probably, you're learning and becoming more capable over time as you do it. Lots of programmers fake it until they make it*” (Thread 8: 20-21). Auch erfahrene Entwicklerinnen und Entwickler können demnach von generativer KI profitieren, um neue Fähigkeiten außerhalb ihres gelernten Bereichs zu erschließen:

“I find ChatGPT most useful to me when I need to learn something new, like a library. I can give it my general context and it will save me hours of googling because I don’t have to follow a trail of breadcrumbs. And I learn by example best” (Thread 8: 25).

Die Aussagen dieses Typus deuten in vielerlei Hinsicht auf eine Kompetenzentwicklung hin: Die professionellen Fähigkeiten werden einigen Beschreibungen zufolge durch die Nutzung von KI potenziert und/oder erweitert. Während einige Beiträge betonen, dass KI vor allem dann wirksam unterstützt, wenn bereits Vorkenntnisse vorhanden sind, heben andere hervor, dass insbesondere in unbekannten Bereichen ein starker Lerneffekt entsteht.

5.2.2 Deprofessionalisierung

Zentrale Aussagen:

- „loss of skills, dependency“
- „knowledge monopoly“
- „shame of use“
- „maintain job without skills“

Beschreibung:

Bei der Deprofessionalisierung wird ganz im Gegenteil zum zuvor genannten Typus die Sorge vor einem Verfall professioneller Kompetenzen benannt. Wenn sprachbasierte Chatbots zur Codegenerierung genutzt werden, erfolgt die Arbeit über Prompts in natürlicher Sprache statt direkt in der Programmiersprache. Dadurch kann Routine verloren gehen und mit ihr wichtige Lerneffekte. Besonders die nachträgliche Feinjustierung oder Anpassung des Codes wird dann erschwert, weil der erzeugte Code nicht aus dem eigenen Schreibprozess stammt und somit fremde Strukturierungs- und Denkweisen aufweist. “For small projects, you're cheating yourself of the learning experience, or backing yourself into a corner where you can't go back and extend/debug/modify without significant trouble” (Thread 8: 35). Die Problematik liegt den Kommentaren zufolge allem darin, dass ein Code kein Produkt ist, das einmal fertig ist und dann für sich steht, sondern er vielmehr ständig aktualisiert und modifiziert werden muss. Für diese kontinuierliche Arbeit ist ein Verständnis der Architektur des Codes essentiell. Ein User schreibt dazu:

“I know it can seem elitist/gatekeepy, but it's not even about that, it's the fact that you as the owner of a thing you've made SHOULD know how that thing works intimately. [...] What you've made WILL stop working sooner or later, and when it does, you can't rely on AI to debug itself, or aim for compatibility, especially as older systems get more obscure” (Thread 8: 36).

Das bedeutet: Auch wenn das Ergebnis prompt-basierter Entwicklung zunächst funktioniert, wird die spätere Wartung des Codes durch erneute textbasierte Anweisungen nahezu unmöglich.

Gerade diese Wartung ist jedoch ein zentrales Element professioneller Softwareentwicklung, da Code langfristig angepasst, erweitert und verstanden werden muss. “[T]he problem with that is you'll still need to learn how to code, like lets say AI makes u an app or something and its not exactly how u imagined it. you won't know how to interact with the app” (Thread 8: 38). Eine andere Person begründet ihre Aversion zu KI-Nutzung: mit “it made me feel disconnected from thinking through my code” (Thread 6: 25).

Der Programmierberuf ist nicht allein durch technisches Fachwissen geprägt, sondern in hohem Maße auch durch kreative Problemlösungsprozesse, Flexibilität und die Fähigkeit, neuartige Lösungswege zu entwickeln. Diese kreativen Anteile machen einen wesentlichen Bestandteil der Softwareentwicklung aus. Wenn jedoch zunehmend Inhalte durch KI erzeugt werden, kann genau dieser kreative Kern der Tätigkeit in den Hintergrund treten. In der gegenwärtigen Nutzungspraxis übernehmen KI-Systeme vor allem standardisierte Abläufe und Routinetätigkeiten, während die Entwicklung origineller, kontextsensitiver Lösungen nur eingeschränkt unterstützt wird. Ein Reddit-User schreibt beispielsweise, „[i am] feeling like I am losing creativity [and] problem solving skill while i am using it“ (Thread 7: 28). Ein anderer User drückt es polemischer aus und schreibt, wer KI nutzt sei „[...] eminently employable .. yes, but essentially just a trained rat that is good at navigating one particular maze and finding the cheese“ (Thread 8: 60-62).

In der Konsequenz machen diejenigen, die mit der generativen KI arbeiten und darüber ihre fachlichen Fähigkeiten verlieren, sich auch hochgradig abhängig von den Produkten der Big-Tech-Companies: “At some point, you will hit a roadblock where even AI cannot help you. Then you will regret why didn't you learn about the library” (Thread 7: 26).

Durch die Nutzung von KI wird außerdem von mehreren Usern besonders betont, dass diese eine Entfremdung von der technischen Informatik eine ganze Generation betrifft, weshalb ein gesellschaftlicher Kohorteneffekt in der Softwareentwicklung zu erwarten ist: „We never expected gen X and millennials (and probably early gen Z) would be the only computer literate generations as a whole“ (Thread 8: 11-15). Was den Kommentaren zufolge in der Konsequenz dazu führt, dass die Produkte dieser Generation erhebliche Mängel ausweisen, zu dessen Lösung das technische Verständnis fehlt: “In fact, I believe LLMs will create a generation of shitty devs who can't actually reason through problems without it, and will create a tsunami of bugs that will require devs with critical thinking skills to solve” (Thread 4: 42). „Richtiges“ Programmieren wäre der Argumentation nach eine Tätigkeit die nostalgischen Hobbyprogrammierinnen und -programmierern vorbehalten bleibt: “In 100 years we'll have twisted mustache, waist coat wearing, nostalgists

relearning and writing c++ in their sheds because the quality of the mass produced crap that's available out there just isn't good enough" (Thread 4: 34-35).

Die Aussagen blicken eher pessimistisch auf die Zukunft der Programmierung. Man versteht sich selbst als möglicherweise letzte Generation, die die erforderlichen technischen Kompetenzen noch umfassend beherrscht. Die User befürchten, dass der verbreitete Einsatz von KI dazu führt, dass nachfolgende Entwicklergenerationen durch die ständige Exposition zu KI-Hilfen deutlich weniger lernen, den Bezug zur technischen Informatik verlieren und dadurch langfristig schlechtere Software produzieren.

5.2.3 Statusbewahrung

Zentrale Aussagen:

- „no use/need“
- „ai can't replace a good programmer“

Beschreibung:

Ähnlich wie bei den zuvor diskutierten, eher KI-skeptischen Positionen im Zusammenhang mit professioneller Identität wird hier der Anspruch formuliert, menschliche Tätigkeiten klar von maschinellen Leistungen abzugrenzen. Diese Aussagen stellen allerdings nicht primär die Zukunft der Profession, sondern die technische oder praktische Leistungsfähigkeit der KI in Frage und betonen, dass sie nicht in der Lage ist, menschliche Entwicklerinnen und Entwickler zu ersetzen oder ernsthaft zu unterstützen.

Einige Programmierer haben hier noch eine verhältnismäßig egalitäre Einstellung und betonen, dass sie KI zwar ab und an nutzen, diese aber nicht wirklich brauchen: „I mean technically I use copilot [...] but [I] barely use chat and I accept a suggestion from time to time. It's ok but it didn't changed much and I wouldn't cry if it would be gone tomorrow" (Thread 6: 63). Die meisten Kommentare dieser Kategorie suggerieren aber vielmehr, dass die Nutzung von KI die eigene Arbeit sogar substanziell verschlechtern. Beispielsweise wird betont, dass die KI-gestützte Arbeit schlicht viel länger dauert, als die manuelle Arbeit: “[...] in the time it takes to craft a specific enough prompt to pass as "good" code I could probably just write it myself" (Thread 6: 13). Dazu kommt, dass die KI den Workflow der Entwicklerinnen und Entwickler ablenkt und damit zusätzliche Probleme verursacht: „I don't use it. I find its suggestions frustrating as I want to often do

something utterly different [...] from what it suggests, and find it slows me down to have to figure out if its suggestions are actually useful or not” (Thread 7: 25). Andere Kommentare betonen die besonderen Anforderungen ihrer Tätigkeiten und zweifeln die Automatisierung der Softwareentwicklung als Kreativ- und Wissensarbeit durch KI an:

“Where I am, AI isn't going to help you. LLMs can only generate permutations of their training data as output. Garbage in, garbage out. So if they've never been trained on it, they can't do it. At all. This means LLMs can never produce anything new or unique. What I need from engineers is going to be pretty new and unique” (Thread 8: 51).

Ein noch grundlegenderes Problem der KI-Nutzung ist, dass generative KI-Systeme oft ungenauen, unbrauchbaren oder fehlerhaften Code produzieren und damit die manuelle Arbeit sogar nachträglich noch torpedieren können. Ein User schreibt zum Beispiel, er probiert Copilot immer mal wieder aus und “every single time it has produced nonsensical continuations of previous code and nothing useful. Maybe it helps if you exclusively make CRUD shit, but for my library dev purposes it is actively harmful” (Thread 5: 52-53). Eine Übernahme von KI-generiertem Code, welcher nicht vorher durch einen erfahrenen Softwareentwickler sorgfältig geprüft und getestet wurde, kann beispielsweise bei Anwendung in der kritischen Infrastruktur sogar richtig gefährlich werden:

„Anything remotely complicated is useless, and frankly dangerous for anything that is actually critical. I've found I can do it just as quickly, if not more so, because it's largely done right the first time instead of having to unpick copilots gibberish. People who have used copilot are seemingly forever fixing bugs and having to deal with pissed off users, and so they banned it completely at work for all uses. The truth is that for serious work it is untrustworthy. I've found those who are using copilot successfully are not on serious or complicated projects” (Thread 5: 43-45).

Nicht zuletzt fallen auch ethische Bedenken in die Distinktion von KI-Nutzung in die Bewertung mit ein: „I'm against the use of AI tools for environmental and ethical reasons. I think its use in society should be restricted to where it has a clear benefit” (Thread 6: 70). Hinter dieser Grenzziehung steht eine implizite Vorstellung von menschlicher Kreativität, Erfahrung und Verantwortungsbewusstsein, die durch KI als nicht reproduzierbar bewertet wird.

Dieser Typ sieht keinen wirklichen Nutzen in KI-Tools und betont, dass gute Programmiererinnen und Programmierer durch sie weder ersetzt noch wesentlich unterstützt werden können. KI wird hier als ineffizient, unzuverlässig oder sogar gefährlich wahrgenommen, sodass ihre Nutzung die eigene Arbeit eher behindert als verbessert.

5.2.4 Anpassungsdruck

Zentrale Aussagen:

- „fear of losing job“
- „we have to adapt to ai / use is necessary“

Beschreibung:

In diesem Cluster wird die Nutzung von KI nicht als individuelle Wahl oder Überzeugung dargestellt, sondern als berufliche Notwendigkeit und unausweichlicher Trend. Die Kommentare in dieser Kategorie verweisen auf einen externen Veränderungsdruck, der durch technologische Entwicklung, Branchendynamiken und soziale Vergleichsprozesse entsteht.

Auf der einen Seite stehen hier Aussagen mit pragmatischer Einsicht. So heißt es etwa: „Typically I have a distrust for these things but I figure why fight it. It’s coming. If it can lighten my load, I’ll be better off for it“ (Thread 6: 31). Hier zeigt sich eine Haltung, die zwischen Skepsis und Anpassung pendelt. KI wird akzeptiert, die nur darin begründet liegt, dass die KI ohnehin kommt und potenziell Entlastung bietet. Andere vertreten noch stärker die Ansicht, dass die Nutzung notwendig ist. Die Kommentierenden beschreiben KI als festen Bestandteil professioneller Praxis, dessen Einsatz geradezu als selbstverständlich gilt. Wer sich dem entzieht, riskiert, den Anschluss zu verlieren. „If you aren’t using LLMs to enhance your productivity and knowledge, you are missing out“ (Thread 4: 23).

Mit derselben Begründung wird der Druck zur Nutzung auch individualisiert: „If you aren’t using LLMs to increase your productivity, you are only hurting yourself“ (Thread 4: 91). Hier erscheint die Anpassung an technologische Entwicklungen als persönliche Verantwortung, während mangelnde Nutzung als selbstverschuldete Ineffizienz gilt. Dazu zeigt sich ein impliziter sozialer Druck: Nichtnutzung wird mit Rückständigkeit und Verpasstem gleichgesetzt. Dieser Druck wird zusätzlich durch eine normative Aufladung verstärkt, die KI-Nutzung als Status quo markiert und Nichtnutzende abwertet. Ein Kommentar formuliert dies explizit: „In 2024 it is stupid NOT to use automation tools“ (Thread 8: 79), eine weitere, ähnlich normative Aussage lautet: “It’s a tool, use

it. And adapt as the world changes” (Thread 5: 27-28). Die Verwendung von KI erscheint hier nicht mehr als freiwillige Option, sondern als Zeichen rationalen und professionellen Handelns. Zugleich wird die Nutzung von KI mit professionellem Selbstverständnis verknüpft. Die Aussage „As IT specialists we should be the best at using these tools“ (Thread 7: 15) verdeutlicht, dass KI-Kompetenz als Teil der fachlichen Identität und des Anspruchs an den Berufsstand gilt. In diesem Verständnis gehört der souveräne Umgang mit neuen Technologien zum Kern professioneller Expertise.

Zugleich artikulieren andere Beiträge die Angst, dass der Wandel berufliche Existenzen gefährden könnte. Die Metapher „You can ride the wave, or you can drown in it. Your choice“ (Thread 4: 12–19) bringt diese Angst auf den Punkt: Anpassung erscheint als einziger Weg, um nicht von der technologischen Welle überrollt und anschließend vermeintlich unbrauchbar zu werden. KI wird so zum Symbol eines unausweichlichen Strukturwandels, der Gewinner und Verlierer hervorbringt. Manche reagieren auf diesen Druck mit Resignation oder Umorientierung: „Probably work in pubs instead, a place for everyone to drown out the misery of their redundancies“ (Thread 2: 23). Diese Kommentare spiegeln die Sorge um Arbeitsplatzsicherheit und berufliche Zukunft wider, die den Anpassungsdruck zusätzlich verstärkt.

Insgesamt zeichnet dieser Typ ein Bild von erzwungener Akzeptanz: Die Nutzung von KI wird als notwendige Bedingung verstanden, um in einem sich rapide wandelnden Berufsfeld relevant zu bleiben. Zwischen pragmatischer Nützlichkeit, sozialer Erwartung und existenzieller Angst entsteht ein Diskurs, in dem technologische Anpassung zum Imperativ wird: Wer nicht mitmacht, riskiert, unterzugehen.

5.2.5 Zwischenfazit: Dimension Identität

Die Dimension Identität zeigt, wie tiefgreifend generative KI in das berufliche Selbstverständnis von Softwareentwicklerinnen und -entwicklern eingreift und wie unterschiedlich die daraus entstehenden Dynamiken sind. Die vier kodierten Typen lassen sich jeweils als Pole zweier Spektren bewerten: Ein Spektrum bewegt sich entlang der Frage, ob KI das Kompetenzspektrum des Berufs eher erweitert oder schmälert, das andere Spektrum entlang der Frage, ob die Nutzung von KI überhaupt notwendig und nützlich ist.

Einige sehen in KI eine Möglichkeit, die ihre Lernprozesse erweitert, ihnen neue Fähigkeiten erschließt und sie fachlich voranbringt. Diese Perspektive ist von Aufbruch geprägt: KI wird als Motor persönlicher Entwicklung verstanden. Dem entgegengesetzt steht eine deutlich düstere Sichtweise, der zufolge KI nicht zu mehr, sondern zu weniger Kompetenz führt. Hier entsteht die Sorge,

dass Lerngelegenheiten verloren gehen, technische Tiefe verschwindet und der Beruf insgesamt an Professionalität einbüßt. Während der erste Typ also eine wachsende Expertise erwartet, rechnet der zweite mit einer schleichenden Erosion fachlicher Standards.

Die zweite Dynamik zeigt sich in der Haltung jener, die ihre eigene professionelle Position durch den Einsatz von KI nicht gefährdet sehen. Sie betonen die Überlegenheit menschlicher Expertise, zweifeln am praktischen Nutzen von KI und verteidigen damit implizit ein traditionelles Berufsethos. Diese Haltung ist stabilisierend: Der Status quo wird gegen den technologischen Wandel behauptet. Ganz anders erleben jene den Wandel, die einen erheblichen Anpassungsdruck verspüren. Für sie erzeugt KI weniger Chancen als vielmehr Erwartungen, denen sie sich nicht entziehen können. Sie fürchten, ohne KI den Anschluss zu verlieren, und erleben die technologische Entwicklung als Zwang zur beruflichen Neuausrichtung.

In der Zusammenschau lassen sich diese vier Typen als zwei zentrale Spannungsfelder lesen: Auf der einen Seite steht die Erwartung wachsender Kompetenz im Kontrast zur Befürchtung einer umfassenden Deprofessionalisierung; auf der anderen Seite treffen die Abwertung der KI als vermeintlich überflüssiges Werkzeug auf das Gefühl, sich unter hohem Druck anpassen zu müssen. Diese gegensätzlichen Interpretationen verdeutlichen, wie stark KI nicht nur Arbeitsprozesse verändert, sondern auch Spannungsfelder entlang der Deutung professioneller Identität erzeugt.

5.3 Dimension Rollenzuschreibung

Diese Dimension erfasst, welche Rollenzuschreibungen der KI im Arbeitsprozess zugeschrieben werden. Aus insgesamt elf kodierten Nutzeräußerungen ließen sich drei zentrale Typen der Rollenzuschreibung von KI in der Mensch-Maschine-Interaktion herausarbeiten. Erstens wird KI als Werkzeug verstanden, welches nur solche Aufgaben automatisiert, welche genauen Anweisungen folgen. Zweitens erscheint sie als Assistenzsystem, das augmentierend den menschlichen Denk- und Arbeitsprozess ergänzt und dabei interaktiv bedient wird. Drittens tritt sie als Strukturierungsinstanz auf, bei der KI vorgebend, bestätigend oder korrigierend in den Arbeitsablauf einbezogen wird.

5.3.1 KI als Werkzeug

Zentrale Aussagen:

- „ai can only do simple code“
- „ai takes boring, repetitive tasks“
- „user controls the output of ai“

Beschreibung:

Dieser Typus kumuliert Aussagen, die KI als Werkzeug benennen, welches ausschließlich und direkt auf Anweisungen reagiert. Die Handlungsmacht liegt dabei vollständig beim Menschen, das heißt, entscheidungsbezogene und kreative Steuerungsleistungen werden nicht an die Technologie delegiert, sondern verbleiben bei der handelnden Person.

Ein wiederkehrendes Motiv ist dabei die Überzeugung, dass KI sowieso nur einfache Aufgaben übernehmen kann, die selbst kaum Abstraktion oder Kontextverständnis erfordern. Ein User betont etwa: „Simple and redundant code that doesn't require much brain power is written by the co-pilot. I write more complex logic myself“ (Thread 6:52). Diese Aussage verdeutlicht eine kognitive Trennung zwischen Menschen und Maschine: Während die KI dieser Ansicht nach nur eine phlegmatische Logik bedienen kann, bleibt die intellektuell anspruchsvolle und kreative Tätigkeit beim Menschen. Ähnlich schätzt ein anderer User den Nutzen des Systems nur im Bereich von trivialen Tätigkeiten ein: „Copilot saves typing obvious stuff, but that's about it“ (Thread 5:40), ein anderer schreibt: „Personally it saves me time by doing some of the mundane stuff“ (Thread 7:49). Beide Zitate zeigen, dass KI hier als Unterstützung betrachtet wird, welche nur einfache Prozesse beschleunigt.

In eine ähnliche Kerbe schlagen Kommentare, die den Nutzen der KI nur bei monotonen Tätigkeiten hervorheben. Damit sind Aufgaben gemeint, die zwar langweilig und repetitiv sind, aber nicht unbedingt in ihrer Logik trivial sein müssen. So beschreibt ein Entwickler: „I mostly use it for repetitive work, refactoring, and cleanup“ (Thread 7:29). Solche Aussagen unterstreichen zwar eine die produktivitätssteigernde, aber dennoch klar begrenzte Rolle der KI. Sie fungiert als Automatisierungshilfe, die es dem Anwendenden ermöglicht sich anderen Kreativtätigkeiten zu widmen: „AI cuts out all that wasted time and just hands me the answer. It's a tool that speeds up production time so you can spend more time being creative and less time doing menial code“ (Thread 3:22). Hier wird KI explizit als Zeitsparinstrument beschrieben, das die Effizienz menschlicher Arbeit steigert, ohne dessen schöpferische Dimension zu ersetzen.

Ein interessantes Zitat legt seiner Argumentation eine Unterscheidung zwischen „coding“ (tippen von Code) und „programming“ (konzeptionelles Problemlösen) zugrunde:

“ChatGPT cannot program. It can produce code that someone else has already written, and mix and match to string it together for the prompt you give it, but this means it's only usable for the most menial coding tasks. Note that I said coding,

not programming. Coding is to programming as typing is to writing, coding is the physical act of writing code. ChatGPT is hopelessly lost when you try to tell it do something with any level of complexity or basically anything that isn't a StackOverflow question“ (Thread 2: 13).

Darüber hinaus wird die Abhängigkeit der KI von menschlicher Steuerung betont, was die Vorstellung von KI in der Rolle eines passiven Werkzeugs stärkt. Ein User schreibt: „if my current code quality is ass, then it's still ass but autocompleted ass“ (Thread 6:47), damit wird deutlich, dass KI weder Qualität noch Kreativität „von selbst“ erzeugen kann, sie bedient lediglich die vom Anwender bestehenden Fähigkeiten: “CoPilot is just a tool, it is up to your use” (Thread 7: 30).

Insgesamt zeigen diese Aussagen ein konsistentes Muster: KI wird von vielen Usern auf Reddit nicht als kreativer oder autonomer Akteur wahrgenommen, sondern als Werkzeug, das klar vorgegebene Aufgaben ausführt, die ihm der Mensch vorgibt. Die Kommentare suggerieren, dass menschliche Kontrolle, Urteilkraft und kreative Kompetenz weiterhin als zentral gelten. Die KI bleibt ein technisches Hilfsmittel, nützlich, effizient, aber grundsätzlich passiv.

5.3.2 KI als Assistenz

Zentrale Aussagen:

- „use as an assistant“
- „ai completes my thoughts“
- „ai needs to get a strong direction“

Beschreibung:

Die hier beschriebenen Aussagen machen deutlich, dass insbesondere KI-Systeme, insbesondere dialogische Modelle, nicht länger ausschließlich als neutrale Werkzeuge verstanden werden, sondern als Formen der Assistenz, die in ein dialogisches Verhältnis zum Nutzer treten. Während im Werkzeugmodell eine klare Input-Output-Logik dominiert, verschiebt sich im Assistenzmodell die Beziehung hin zu einem interaktiven Verhältnis. Die KI wird hier als eine Art Begleitung imaginiert, die den eigenen Denkprozess ergänzt und antizipiert.

Ein zentrales Merkmal dieses Verständnisses ist die Vorstellung einer gemeinsamen, aber gleichsam asymmetrischen Kooperation: Der Mensch gibt eine Richtung vor, die KI assistiert. Wie ein User beschreibt, sei es „like having my own personal assistant I can dictate how to solve the problem“ (Thread 4: 29). Diese Aussage verweist darauf, dass die Deutungshoheit über den

Arbeitsprozess klar beim Menschen verbleibt. Dennoch unterscheidet sich das Verhältnis grundlegend vom klassischen Werkzeuggebrauch, da die KI hier nicht als neutrales Instrument verstanden wird, sondern als interaktive Instanz.

Oft wird explizit betont, dass die KI nicht nur Anweisungen ausführt, sondern den eigenen Denkprozess fortsetzt oder vervollständigt. Ein User formuliert dies als „it completes my thoughts“, während eine andere Person beschreibt, das System sei „able to just sort of autocomplete what I would be typing next for the other classes in the project“ (Thread 6: 45). Die User erleben also, dass die KI ihre Gedanken fortschreibt, mögliche nächste Schritte antizipiert und so die kreative oder konzeptionelle Arbeit unterstützt. Eine weitere Aussage verdeutlicht dies: „I define the strategies and it fills in the more tedious bits“ (Thread 6: 53). Hier wird nahegelegt, dass die KI vor allem operative Aufgaben übernimmt, während die strategische Kontrolle und das kreative Entscheiden beim Menschen verbleiben.

Insgesamt entsteht in diesem Typus das Bild einer partnerschaftlichen Zusammenarbeit zwischen Mensch und Maschine, die weder vollständig instrumentell noch autonom ist. Die KI fungiert als Assistenzsystem, das dabei aber zentral auf externe Leitung angewiesen ist, dabei jedoch in der Lage ist, die Gedanken, Routinen und Intentionen des Menschen fortzusetzen. Damit markieren diese Beschreibungen die KI in einer Zwischenposition zwischen Werkzeug und eigenständigem Akteur. Sie betonen, dass der Mensch die Richtung vorgibt und die KI den Weg dorthin mitgestaltet.

5.3.3 KI als Strukturierungsinstanz

Zentrale Aussagen:

- „ai as a mentor“
- „ai sets the flow“
- „ai doesnt judge me“
- „ai is wrong confidently“

Beschreibung:

Während in den ersten beiden Rollenzuschreibungen der Mensch zwingend die Richtung vorgibt und die KI lediglich auf die ein oder andere Weise dabei unterstützt, wird hier eine Arbeitsweise beschrieben, in welcher die Handlungsträgerschaft teilweise abgegeben wird. Die KI gibt dabei beispielsweise den Arbeitsfluss vor, korrigiert Prozesse oder widerspricht menschlichen Annahmen. Diese Vorstellung deutet auf eine partielle Umkehrung der Steuerungslogik hin: Die KI wird

nicht länger nur als Werkzeug oder Assistenz des Menschen verstanden, sondern als aktive, einlenkende Instanz im Arbeitsprozess.

Ein zentrales Motiv in den Beschreibungen ist die Wahrnehmung der KI als “starting point” (Thread 8: 23), um den Arbeitsprozess einzuleiten. “Getting into flow quicker is incredibly understated. [...] It doesn't get it right all the time, but it helps bring the code and the coder into better alignment” (Thread 5: 22-24). Auch wenn die KI generierten Suggestionen also falsch sind, helfen sie, den schwierigen Anfang zu überwinden:

“I've never thought about it this way but you're absolutely describing me. I'll type a bit and Copilot will write a first draft. Occasionally it's perfect but more likely I have to tweak or replace it. Even when it's dead wrong, it frees me from writer's block -- now I'm fixing its mistakes instead of staring through the wall thinking about how to solve the issue” (Thread 5: 29-30).

Nicht nur der Start wird dabei erleichtert, sondern der gesamte Prozessfluss wird unterstützt und teilweise vorgegeben. „It's not a bad tool to have for improving your workflow” (Thread 8: 72). Auch wenn hier weiterhin die Bezeichnung „tool“ verwendet wird, entspricht die hier beschriebene interaktive Dynamik nicht dem klassischen Werkzeugcharakter. Zum Beispiel wird KI im Arbeitsprozess genutzt um sich zu vergewissern, dass man noch auf einem guten Weg ist: “I also think AI is a good tool for getting confident in what you know. It's pretty good with general concepts and as someone who just graduated it makes it easier to make sure I'm on the right track” (Thread 8: 49).

Doch diese Verschiebung hat auch ihre Schattenseiten: Einige erleben dagegen den durch die KI initiierten Arbeitsfluss als Kontrollverlust. So heißt es etwa:

“[AI's] auto suggesting make me confuse and lose focus on writing code. After try for a couple hours I am no longer tend to read official documentation for deeply learning about the library I use. Instead, I just followed AI suggesting” (Thread 7: 9).

Die Programmierinnen und Programmierer folgen, einigen Beiträgen nach zu urteilen teilweise wie automatisiert den Impulsen der Maschine, oft ohne es zu bemerken. Der Arbeitsfluss, den die KI anstößt, kann damit ebenso produktiv wie entfremdend wirken.

“You have to be on the lookout though. There is this thing that happens now and then when I think I know what I want it to write before it writes it and when I read what it writes it is so similar to what I thought I wanted it to write that I now believe it wrote what I wanted but still it is not correct but now I forgot what I initially wanted it to write. Then you just sit there for a minute watching the code and your brain is half fried” (Thread 6: 60-61)

In einer weiteren, für die Analyse zentralen Sinngebung erscheint die Interaktion mit LLM-basierten Chatbots als eine Lernbeziehung, in der die KI die Rolle eines mentorhaften Gegenübers einnimmt. Sie tritt nicht als bloßes Werkzeug in Erscheinung, sondern als begleitende Instanz, die Denkprozesse ordnet, Fragen auffängt und Lernbewegungen strukturiert:

„Everyone is so locked into it writing code for them. Yeah, it can write code but that's not what makes it great. Being able to talk through problems, give examples back and forth, and actually increase your understanding is what makes it great“ (Thread 4: 55).

Das Zitat zeigt, dass KI die Rolle eines dialogischen Lernpartners zugewiesen wird, damit dieser die eigenen Denkprozesse begleitet und das Verständnis fördert. Diese offen dialogische Nutzung kann in der Praxis beispielsweise auch helfen eigene Ideen zu realisieren: „when I often don't know what I am doing but I know what I am trying to do and those tools can help me work through it” (Thread 8: 42). Auch für weniger erfahrene Entwicklerinnen und Entwickler erfüllt KI die Funktion eines mentalen oder didaktischen Ankers, insbesondere dann, wenn menschliche Mentorrollen fehlen. Ein User rät Juniors folgendes: „Maybe use it to gather ideas if you are stuck on how to solve something and no mentors are available“ (Thread 7: 38). Ein User beschreibt sogar, dass er seinen Lernprozess gezielt daran ausrichtet, in der Interaktion mit der KI die von ihr generierten Inhalte nachzuvollziehen und zu verstehen. „And if you don't understand how it fixed the issue... You can ask it until you understand!“ (Thread 4: 56). Die KI wird damit zu einer didaktischen Instanz, die unermüdlich antwortet, ein Aspekt, der von vielen als besonders hilfreich erlebt wird. Ein User bringt diesen Aspekt auf den Punkt: „I've literally spent 30mins just asking it what does this do, why did you do that, why didn't you do this and it's like having a big brother programmer to explain everything“ (Thread 1: 10–12).

Dieses Zitat führt direkt zu einem weiteren zentralen Grund, warum KI als Mentor adressiert wird: Die Abwesenheit sozialer Bewertung. Viele vergleichen die KI-Interaktion dabei mit früheren Erfahrungen auf Plattformen wie Stack Overflow: „Personally what I find to be the coolest thing about having Chat GPT help you with code is that it actually tries to help you with code AND not

be obnoxious about it (cough stack overflow cough)“ (Thread 1: 13). Den Beschreibungen der User folgend wird die Erfahrung mit Chatbots, anders als teilweise bei Stack Overflow, nicht als herablassend, sondern im Gegenteil als respektvoll und bestätigend wahrgenommen. „It answers like I am a human with thoughts and feelings“ (Thread 1: 14-15). Ein anderer User schreibt rückblickend: “now that we have chatgpt, thinking about SO gives me creeps. I felt like I was judged every-time I posted a question over there” (Thread 1: 14-15). Damit wird die KI zu einem vermeintlich sicheren Raum, in dem Fragen ohne die Angst vor sozialer Sanktion oder Bewertung gestellt werden können. In dieser Funktion ersetzt die KI nicht nur die Rolle eines Mentors, der geduldig erklärt, ohne zu urteilen, sondern übernimmt zugleich Aufgaben, die sonst dem sozialen Umfeld zukommen: nämlich Lernprozesse durch Rückmeldung, Grenzziehung und soziale Bewertung zu steuern.

“Cut to Stack Overflow and if I asked half the things I did to Chat GPT there, i would be raked across the coals, scoffed at endlessly, and told I should give up coding in any capacity. Chat GPT actually treats me more like a person than actual people” (Thread 1: 16-17).

Gleichzeitig deuten zahlreiche Beiträge darauf hin, dass die zugewiesene Mentorenrolle der KI nicht frei von Ambivalenzen ist. So schildern User immer wieder, dass das System seine Antworten mit großer sprachlicher Sicherheit präsentiert, obwohl sie eigentlich falsch sind: „Most of the time what it says is just wrong. Very elaborate and convincing but completely wrong“ (Thread 7: 31). Noch pointierter fasst es ein anderer User zusammen: „AI, has one huge problem. It lies, and it does it with supreme confidence“ (Thread 4: 39–40).

Insgesamt wird die KI in den hier subsumierten Aussagen als eine strukturierende Instanz, in einigen Fällen sogar als Mittel der Selbstkontrolle adressiert. Besonders weniger erfahrende Entwicklerinnen und Entwickler profitieren davon, die KI als selbstgewählten Mentor zu nutzen, der erklärt, korrigiert und Orientierung bietet, ohne zu bewerten. Dadurch entsteht ein geschützter Raum, in dem Unsicherheiten, Fehler und Unwissen offen gezeigt werden können, ohne soziale Sanktionen wie auf Plattformen wie Stack Overflow zu befürchten. Gleichzeitig verschiebt sich die Kontrolle teilweise zur KI, die den Arbeitsfluss vorgibt und mitunter selbstbewusst falsch agiert.

5.3.4 Zwischenfazit: Dimension Rolle

Die Ergebnisse zeigen ein ausdifferenziertes Spektrum der Rollenzuschreibungen von KI in der Mensch-Maschine-Interaktion. Die drei herausgearbeiteten Typen Werkzeug, Assistenz und Strukturierungsinstanz markieren qualitativ verschiedene Vorstellungen darüber, wie Verantwortung, Kontrolle, Expertise und Kreativität zwischen Menschen und Maschine verteilt werden sollen. Das Werkzeugmodell knüpft an ein traditionelles Technikverständnis an: KI wird hier als effizienter, schneller oder bequemer Ersatz für Routinetätigkeiten gesehen, aber nicht als Partner im Denken oder Entscheiden. Die Kommentare, die auf diese Rollenzuweisung verweisen zeichnen sich oft auch durch Verweise auf ihre eigene Erfahrung aus, was darauf hindeutet, dass professionelle Expertise als Anker dient. Wer die Regeln des Programmierens beherrscht, fühlt weniger Bedarf, Entscheidungsanteile an eine Maschine abzugeben. Ganz anders das Assistenzmodell. Hier wird KI als Instanz wahrgenommen, die nicht nur ausführt, sondern mitdenkt, ergänzt und vorschlägt. Diese Form der Zusammenarbeit erzeugt ein dialogischeres Bild: Der Mensch setzt den Rahmen, und die KI hilft, ihn auszufüllen. Damit werden Verantwortung und Gestaltung nicht abgegeben, sondern geteilt, allerdings stets unter Führung des Menschen. Dieses Verständnis weist darauf hin, dass sich eine neue Art digitaler Kooperation etabliert: technologiegestützt, aber nicht technologiegeführt. Zum dritten Typ gehören Aussagen, die KI eine strukturierende Rolle im Arbeitsprozess zuweisen. Hier wird KI nicht primär genutzt, um Arbeit zu vereinfachen, sondern um Orientierung zu erhalten. Dadurch kann die KI stark strukturierend Einfluss nehmen. Einige beschreiben das System sogar als eine Art Mentor, welcher strukturiert, erklärt und Grenzen setzt. Durch Rückfragen, Beispiele und iterative Erklärungen wird ein Lernrahmen geschaffen, der Unsicherheit reduziert und das eigenständige Verständnis fördert. Die strukturierende Wirkung der KI liegt somit nicht in der Vorgabe fertiger Lösungen, sondern in der Ordnung und Rahmung des Arbeits- und Lernprozesses selbst.

6 Jenseits der Effizienz: Strukturierende Effekte von KI

Ein unmittelbarer Einfluss von KI auf die Arbeit zeigt sich dort, wo ihr Einsatz direkt mit sichtbaren Ergebnissen verknüpft ist, etwa wenn durch fehlerhafte Vorschläge, Halluzinationen oder technische Einschränkungen zusätzlicher Korrekturaufwand entsteht oder Arbeitsabläufe automatisiert und damit verschnellert werden. In diesen Fällen ist der Einfluss klar lokalisierbar, kausal nachvollziehbar und eng an konkrete Nutzungssituationen gebunden. KI besteht hier primär als Werkzeug, dessen Leistungsfähigkeit oder Fehlfunktion den Arbeitsprozess und dessen Ergebnis unmittelbar beeinflusst. Die daraus resultierenden Effekte lassen sich grundlegend mit den in der Dimension Arbeit erfassten Aussagen beschreiben.

Davon zu unterscheiden ist ein impliziter und subtiler Einfluss von KI auf Arbeit, der nicht aus einzelnen Fehlleistungen oder technischer Automatisierung resultiert, sondern aus der weiter reichenden Einbindung der Technologie in alltägliche Arbeitspraktiken. Dieser Einfluss zeigt sich insbesondere darin, wie Entwicklerinnen und Entwickler beginnen, ihr Verhalten an die Logiken, Erwartungen und Grenzen der KI anzupassen. Im Umgang mit generativer KI entstehen spezifische Strategien, etwa darüber, sich mit der KI rückzuversichern und einzelne Strukturentscheidungen an die KI abzugeben. Das geht über eine einfache Delegation hinaus und äußert sich in einer interaktiven Mensch-Maschine-Interaktion. Solche Veränderungen verfestigen sich im alltäglichen Umgang und wirken dann reaktiv auch auf umliegende Bereiche, wie Lernprozesse oder das professionelle Selbstverständnis. In diesem Sinne beeinflusst KI die Arbeit nicht nur durch das, was sie konkret leistet oder nicht leistet, sondern auch dadurch, wie sie Wahrnehmungen von guter Arbeit, angemessenen Vorgehensweisen und professioneller Verantwortung mitprägt.

Diese Effekte sind weniger sichtbar, entfalten ihre Wirkung jedoch langfristig, indem sie Entscheidungspraktiken stabilisieren, Handlungsoptionen rahmen und bestimmte Formen des Arbeitens normalisieren. Gerade dieser implizite Einfluss ist analytisch besonders relevant, da er die Arbeitswelt nicht punktuell verändert, sondern ihre Struktur auf subtile Weise mitformt. Besonders deutlich wird dies in Nutzungspraktiken, in denen der KI Prüf-, Instruktions- und Bewertungsaufgaben delegiert werden. Diese Nutzung kann als eine Art *stellvertretendes Management* verstanden werden. In dieser Funktion eröffnet sie einen geschützten Raum, in dem Entscheidungen, Lösungswege und Vorgehensweisen erprobt, überprüft und abgesichert werden können. Das beeinflusst dann nicht nur die konkrete Ausführung von Arbeit, sondern auch Erkenntnisprozesse, soziale Rückversicherung und Rezeptionen professioneller Anforderungen sowie die Art und Weise, wie Verantwortung wahrgenommen und zugeschrieben wird.

Aus diesem Grund richtet die abschließende Analyse ihren Fokus gezielt auf sich selbst strukturierende und kontrollierende Nutzungsweisen generativer KI. Diese Perspektive erlaubt es, den impliziten Einfluss von KI auf Arbeit systematisch zu erfassen. Die Analyse knüpft deshalb an die deskriptive Kartierung der empirischen Befunde an und soll einen vertiefenden analytischen Zugriff ermöglichen. Während die Kartierung die Breite unterschiedlicher Wahrnehmungen und Nutzungsweisen sichtbar gemacht hat, konzentriert sich die Analyse auf die Rekombination ausgewählter Aussagen und deren Einbettung in die theoretische Forschungslandschaft, um den Einfluss von KI im Arbeitsprozess systematisch zu rekonstruieren.

Im ersten Schritt wird die Nutzung von KI als stellvertretendes Management bzw. die Delegation von klassischen Managementaufgaben wie Strukturierung oder Kontrolle des Arbeitsprozesses an die KI beleuchtet. Dazu werden die empirischen Befunde und Aussagen zu Nutzungspraktiken der KI unter Bezugnahme auf die im Theorieteil bereits benannten etablierten Normen in der Softwareentwicklung, die vor der Implementierung von KI galten, verglichen und analysiert. Ein besonderer Fokus ergibt sich dabei insbesondere auf die Attraktivität dieser Praxis, etwa im Hinblick auf Entlastung, Orientierung und die Absicherung professioneller Entscheidungen. In einem zweiten Schritt werden dann die mit dieser Nutzungsweise verbundenen potenziellen Schattenseiten thematisiert, die sich in den empirischen Befunden finden lassen. Im letzten Abschnitt werden die empirischen Befunde noch vor dem Hintergrund der Konzepte der kollaborativen und kooperativen Grenzarbeit nach Langley et al. (2019) analysiert, um die Ergebnisse theoretisch einzuordnen. Die Perspektive ermöglicht es, die empirischen Befunde als Aushandlung professioneller Zuständigkeiten sowohl entlang der Grenzen zwischen beruflichen Gruppen als auch zwischen Mensch und Maschine zu verstehen.

Ziel des Analyseteils ist es, die strukturierenden Eigenschaften generativer KI im Arbeitsprozess der Softwareentwicklung in einer kontextualisierenden Beschreibung herauszuarbeiten und ihre ambivalente Bedeutung für Arbeitsprozesse und professionelle Identität sichtbar zu machen. Auf diese Weise trägt die Analyse dazu bei, generative KI nicht nur als technologische Innovation, sondern als wirksamen Bestandteil soziotechnischer Transformationsprozesse der Arbeitswelt zu verstehen.

6.1 Subjektive Autonomieerfahrung

Zwar ist Softwareentwicklung historisch mit der Open-Source-Bewegung in einer Kultur der Offenheit, des Austauschs und der kollektiven Wissensproduktion verankert (Bretthauer, 2001; DiBona & Ockman, 1999; Tozzi, 2017); tatsächlich führen aber implizite Kommunikationspraktiken

in den zentralen Foren wie Stack Overflow auch immer wieder zu Ausschlusserfahrungen (Cheriyān et al., 2021; Krishtul et al., 2022). Besonders Berufseinsteigende kann diese Situation Unsicherheiten aufwerfen: Ihnen fehlen nicht nur technische Routinen, sondern auch die Kenntnisse über implizite soziale Normen der Entwickler-Community, etwa, welche Fragen als legitim gelten und wie man sie „richtig“ formuliert. Das notwendige implizite Wissen über Kommunikationsstile und Erwartungshaltungen stellt damit eine Zugangshürde dar. Es ist dabei davon auszugehen, dass diese Hürde nicht nur in den in der Programmierpraxis der viel frequentierten Online-Foren besteht, sondern ebenso in der Interaktion mit Kolleginnen und Kollegen oder Vorgesetzten. Das Kommunizieren von Fragen wird damit auf zwischenmenschlicher Ebene schnell zu einer Prüfung ihrer Zugehörigkeit oder Kompetenz.

Ausgehend von der professionssoziologischen Perspektive nach Abbott (1988) sowie dem daran anschließenden Konzept der Grenzarbeit (Gieryn, 1983; Langley & Leyshon, 2017) lässt sich Interaktion im beruflichen Bereich grundsätzlich nicht als neutraler Austausch von Informationen begreifen. Vielmehr ist jede Form beruflicher Interaktion und damit auch der Wissensaustausch immer zugleich in Prozesse der Aushandlung professioneller Identität, Zuständigkeit und Autorität eingebettet. Vor diesem Hintergrund erhält auch das Nachfragen in Situationen beruflicher Unsicherheit eine weitergehende Bedeutung. Anfragen dienen nicht allein der Klärung sachlicher Probleme, sondern wirken zugleich als performative Akte, durch die sich Akteure innerhalb eines professionellen Gefüges positionieren. Wer fragt, signalisiert nicht nur einen Wissensbedarf, sondern setzt sich potenziell auch einer Bewertung der eigenen Kompetenz und Zuständigkeit aus. Diese Positionierung kann reale Konsequenzen für die Zuschreibung professioneller Identität haben, etwa indem sie bestehende Grenzen zwischen Expertise und Nicht-Expertise bestätigt, verschiebt oder infrage stellt. Wissensaustausch ist in diesem Sinne stets auch mit Macht- und Grenzziehungsprozessen verbunden und kann damit Auswirkungen auf die Wahrnehmung der eigenen Rolle innerhalb der Profession entfalten.

Bis vor Kurzem existierten zu diesen öffentlichen Wissensräumen kaum Alternativen. Wer Unterstützung benötigte, war in der Regel darauf angewiesen, Fragen in öffentlichen Foren zu stellen. Sie stellten den primären Ort dar, an dem Entwicklerinnen Fragen formulierten, Probleme verhandelten und kontextspezifische Lösungsansätze austauschten. Diese zentrale Rolle ergibt sich aus der besonderen Struktur softwaretechnischer Arbeit: Softwareentwicklung ist durch hohe Dynamik, schnelle technologische Veränderungen sowie durch eine Vielzahl kleinteiliger und hochspezialisierter Lösungen geprägt. Institutionalisierte Wissensbestände wie etwa in Form von Handbüchern, Normen oder formalen Supportstrukturen können mit dieser Entwicklung häufig nicht Schritt halten. In der Folge entsteht eine informelle Praxis der Wissensproduktion, in der

relevante Expertise oft nur bei einzelnen Akteurinnen verfügbar ist und über öffentliche Foren situativ zugänglich gemacht wird. (Abdalkareem et al., 2017; Barua et al., 2014; Treude et al., 2011; Wu et al., 2019). Mit dem Aufkommen generativer Sprachmodelle verändert sich diese Situation grundlegend. Wer eine Frage an eine dialogische KI Software richtet, kann dieselbe Form von situativ passenden Antworten bekommen, ist sich dabei aber bewusst, kein menschliches Gegenüber anzusprechen. Gerade darin liegt ein zentraler Unterschied zu anderen Kontexten des Wissensaustauschs. Fragen, die in sozialen oder persönlichen Interaktionen möglicherweise als redundant, banal oder unangebracht bewertet würden, können hier in einem Raum gestellt werden, der frei von unmittelbarer sozialer Bewertung ist. Entscheidend ist dabei, dass es nicht primär um tatsächliche negative Reaktionen geht, sondern um deren Antizipation im Vorfeld der Frageformulierung. Bereits die Erwartung möglicher Bewertungen kann beeinflussen, ob und wie eine Frage überhaupt gestellt wird. In der Mensch-Maschine-Interaktion entfällt diese antizipierte soziale Rückmeldung weitgehend. Da kein menschliches Gegenüber adressiert wird, müssen keine potenziellen Reaktionen, Sanktionen oder reputationsbezogenen Effekte mitgedacht werden. Dadurch entsteht ein veränderter Bezugsrahmen des Fragens, in dem Unsicherheit nicht sozial moderiert, sondern individuell artikulierbar wird. Kurz gefasst erweitert die technische Verfügbarkeit dialogischer Antworten, die ohne soziale Interaktion mit einem menschlichen Gegenüber vollzogen werden können, aus subjektiver Perspektive den Möglichkeitsraum der Fragestellungen. Sie verschiebt die Bedingungen des Fragens von einem sozial regulierten zu einem primär explorativen Akt.

Zwar erzeugen Chatbots sprachlich Antworten, die menschlicher Kommunikation ähneln, jedoch sind Modelle wie ChatGPT durch Verfahren des *instruction tuning* explizit darauf ausgelegt, bestätigend, unterstützend und kooperationsorientiert zu reagieren (Bai et al., 2022; Bardol, 2025; Ouyang et al., 2022). Die Interaktion bleibt damit von den impliziten Bewertungs- und Machtstrukturen menschlicher Kommunikation entkoppelt. Gerade diese Kombination aus formaler Nicht-Menschlichkeit und zugleich affirmativer Interaktionslogik macht generative Sprachmodelle zu einer spezifischen Form professioneller Rückversicherung. Genau an dieser Erfahrung setzten auch die Reddit-User mit dieser Aussage an: “now that we have chatgpt, thinking about [StackOverflow] gives me creeps. I felt like I was judged everytime I posted a question over there” (Thread 1: 14-15). Gerade in Situationen, in denen Fragen nach Zugehörigkeit und Kompetenz besonders drängend sind, kann KI so zu einer bevorzugten Option der Wissensaneignung und des explorativ experimentierenden Lernens werden. Die Möglichkeit, unabhängig von Vorwissen, Fehlern oder Unsicherheiten mit wertschätzenden Rückmeldungen zu rechnen, schafft eine Kommunikationsumgebung, die stärker von emotionaler Sicherheit als von inhaltlicher Präzision

geprägt ist. Für Nutzerinnen und Nutzer entsteht damit ein geschützter Raum, in dem sie sich ohne strategische Selbstdarstellung äußern können. Während der Austausch mit Fachkolleginnen und -kollegen immer auch eine soziale Performance ist, entfallen bei der KI diese Anforderungen vollständig. Damit wird verständlich, warum einige Nutzende trotz bekannter und viel hervorgehobener Fehlerrisiken der KI treu bleiben: Die technische Qualität der Antwort ist nur ein Teil des Nutzungserlebnisses; ebenso wichtig ist die soziale Qualität der Rückmeldung.

Vor dem Hintergrund dieses Befunds lohnt ein genauerer Blick auf die vier Eigenschaften, die Kellogg et al. (2020) als konstitutiv für algorithmische Kontrolle beschreiben: eine umfassende Wissensbasis (*comprehensive*), sofortige Rückmeldung (*instantaneous*), dialogische Interaktivität (*interactive*) sowie Opazität der Funktionslogik (*opacity*). In der Forschungsliteratur werden diese Merkmale überwiegend kritisch diskutiert, da sie, eingesetzt in der Plattform- oder Gigarbeit, in die Lage versetzen, Verhalten in Echtzeit zu erfassen, zu bewerten und zu steuern. Im Kontext der freiwilligen, selbstgesteuerten KI-Nutzung können diese Eigenschaften zwar genauso auftreten, gleichsam aber gegenteilig autonomiefördernd gedeutet werden: Genau jene Eigenschaften, die als potenziell entmündigend gelten, werden hier als besonders hilfreich und entlastend erlebt, was die Perspektive auf algorithmische Kontrolle deutlich erweitert. Wie dies geschieht, wird im folgenden Abschnitt erläutert.

Die umfassende Wissensbasis (*Comprehensive*) wird nicht als überwachende Datenmacht erlebt, sondern als Wissensressource: Mit der KI-Nutzung greift man auf ein breites, kontextreiches Wissensfeld zu, ohne auf soziale Schwellen und Zugangsbeschränkungen zu stoßen. Die sofortige Reaktivität (*Instantaneous*) bietet auch sofortige Antworten und damit eine selbstbestimmte Taktung der Zugänge. Entwicklerinnen und Entwickler können jederzeit auf Feedback zugreifen und sind dabei nicht länger abhängig von Reaktionszeiten Dritter oder der Bereitschaft anderer, Hilfe zu leisten. Die iterative, dialogische Struktur der Interaktion (*Interactive*) unterstützt dabei, ein exploratives, individualisiertes Lernen: Rückfragen, Varianten und Fehlversuche können unmittelbar ausprobiert werden, ohne Erwartungsdruck oder normative Bewertung durch andere. Besonders bedeutsam ist schließlich die Intransparenz der Antwortentstehung (*Opacity*). Diese Eigenschaft wird nicht als Kontrolle oder Machtasymmetrie wahrgenommen, sondern als sozialer Schutzraum. Die Opazität des Systems erlaubt es, Fragen zu stellen, die man in sozialen Kontexten möglicherweise zurückhalten oder für die man in öffentlichen Foren sanktioniert würde. Die Abwesenheit eines identifizierbaren Anderen, also das Fehlen eines menschlichen Reaktanten, schafft eine Situation, in der keine wahrnehmbare Exposition stattfindet. Auf diese Weise entsteht ein Gegenentwurf zur klassischen Logik algorithmischer Macht: ein Raum subjektiver Autonomie, in dem Ausprobieren und Scheitern scheinbar unbegrenzt erfolgen können.

Zusammenfassend zeigt sich, dass der Einsatz von KI von Entwicklerinnen und Entwicklern durchaus als autonomieerweiternd wahrgenommen wird. Indem KI als nicht-menschliche, zugleich unterstützende Instanz genutzt wird, ermöglicht sie es, Unsicherheiten zu adressieren, ohne sich sozialen Bewertungs- und Aushandlungsprozessen aussetzen zu müssen. KI fungiert damit als Form professioneller Rückversicherung, die Handlungssicherheit stabilisiert und individuelle Autonomie im Arbeitsprozess stärken kann.

6.2 Erosion von Handlungs- und Lernprozessen

Im folgenden Kapitel werden die negativen Konsequenzen dieser Nutzungsform aus den empirischen Befunden herausgearbeitet. Der durch KI eröffnete geschützte Raum erweist sich dabei als ambivalent. Während Entwicklerinnen und Entwickler die Nutzung, wie im zuvorgehenden Abschnitt beschrieben, als Quelle von Strukturierung, Rückversicherung und Selbstkontrolle beschreiben, gehen mit dieser Form der Mensch-Maschine-Interaktion zugleich Verschiebungen einher, die zentrale Handlungs- und Lernprozesse unterlaufen können. Aus dieser Ambivalenz ergeben sich zwei zentrale Problembereiche, die im Folgenden näher analysiert werden.

Das erste zentrale Risiko dieser Nutzungspraxis liegt in der schrittweisen Verlagerung von Einfluss- und Steuerungspotenzialen (Agency) im Arbeitsprozess auf KISysteme. Aus zahlreichen Reddit-Beiträgen wird deutlich, dass die durch KI generierten Vorschläge spezifische Handlungsmöglichkeiten bzw. „Affordanzen“ (Hutchby, 2001) erzeugen, welche den Arbeitsfluss zunehmend strukturieren und vortaktieren. Die KI „setzt den Flow“, indem sie fortlaufend nahelegt, welche nächsten Schritte sinnvoll oder naheliegend erscheinen. Dadurch verschiebt sich das Verhältnis zwischen Unterstützung und Steuerung in grundlegender Weise: Sichtbar eingeblendete Vorschläge, automatische Vervollständigungen und dialogisch formulierte Impulse wirken wie Handlungseinladungen, die den Arbeitsfluss subtil steuern. Mit zunehmender Nutzungsdauer steigt zudem die Bereitschaft, diese Vorschläge automatisch zu übernehmen: “After try for a couple hours I am no longer tend to read official documentation for deeply learning about the library I use. Instead, I just followed AI suggesting.” (Thread 7: 9). Hier wird sichtbar, wie das System Lern- und Entscheidungsprozesse subtil umlenkt, weg von einer aktiv gestaltenden Arbeitshandlung hin zu einer stärker reaktiven Orientierung an maschinellen Impulsen. Der Arbeitsfluss wird dabei teilweise durch die kontinuierlichen Einblendungen der KI vorstrukturiert: Sie rahmt, welche Lösungsmöglichkeiten überhaupt in den Blick geraten. Ähnliche Berichte beschreiben, dass Vorschläge unabsichtlich akzeptiert werden, weil sie sich automatisch einblenden: “I sometimes find myself accidentally accepting a terrible suggestion because it pops up just as I was using the tab key or something for another purpose” (Thread 7: 21).

Gerade weil der Mensch formal weiterhin die entscheidende Instanz im Arbeitsprozess bleibt, ist diese soziotechnische Form der Einflussnahme besonders schwer zu erkennen. Die KI liefert Suggestionen, die häufig eng an das anschließen, was die Person möglicherweise selbst gedacht hätte, wenn sie den Raum gehabt hätte, diesen Gedanken eigenständig weiterzuentwickeln. Dadurch, dass der Vorschlag jedoch früher erscheint, wird der eigene Gedankengang unterbrochen oder mit dem maschinell generierten Inhalt überlagert. Die Grenze zwischen selbst hervorbrachter Lösung und algorithmischer Anregung wird damit unscharf, wie auch der folgende Kommentar verdeutlicht:

“You have to be on the lookout though. There is this thing that happens now and then when I think I know what I want it to write before it writes it and when I read what it writes it is so similar to what I thought I wanted it to write that I now believe it wrote what I wanted but still it is not correct but now I forgot what I initially wanted it to write. Then you just sit there for a minute watching the code and your brain is half fried” (Thread 6: 60-61).

Das Besondere liegt darin, dass die Verlagerung des Denkprozesses nicht bewusst wahrgenommen wird. Die Person im Zitat beschreibt, wie sie den eigenen ursprünglichen Gedanken vergisst, sobald der Vorschlag der KI plausibel erscheint. Damit verliert der Mensch den Zugriff auf den inneren Referenzpunkt seiner eigenen Entscheidung. Hier zeigt sich ein Mechanismus, der subtiler wirkt als klassische Formen algorithmischer Kontrolle: Die KI reduziert nicht Handlungsspielräume, sondern überlagert sie.

Vor diesem Hintergrund ist es besonders problematisch, dass KI-Systeme häufig mit hoher sprachlicher Überzeugung auch falsche oder unsichere Inhalte generieren. Ein User bemerkt: “AI, has one huge problem. It lies, and it does it with supreme confidence” (Thread 4: 39-40). Die sprachliche Form vermittelt Glaubwürdigkeit, selbst wenn die Substanz fehlt. Dadurch entsteht ein trügerischer Eindruck von Richtigkeit, der die Entwicklerinnen und Entwickler dazu verleiten kann, Ergebnisse zu übernehmen, ohne sie ausreichend zu prüfen, oder Fehler nicht einmal mehr zu bemerken. Die Konsequenzen reichen dabei über einzelne, übernommene Fehler hinaus, wie der folgende Beitrag bemerkt: „LLMs are going to teach people a lot. They also are false too many times to be teaching anything... The amount of unsecure code that gets pushed out will be amazing and wild“ (Thread 4: 67). Das zentrale Problem liegt somit nicht allein in der Übernahme einzelner fehlerhafter Vorschläge, sondern in der Möglichkeit einer systematischen Verbreitung vermeintlich plausibler Lösungswege, die wiederum als Lerngrundlage genutzt werden könnten. Entwicklerinnen und Entwickler lernen im Arbeitsalltag fortlaufend anhand der ihnen präsentierten

Beispiele, etwa indem sie syntaktische Muster, Lösungslogiken oder Strukturierungen übernehmen. Werden solche Inhalte als korrekt wahrgenommen, obwohl sie fehlerhaft sind, besteht die Gefahr, dass sich diese Fehler schleichend im eigenen Wissensbestand verfestigen. Genau dieser Befund führt zum zweiten zentralen Risiko: Dequalifizierung.

Die mögliche Dequalifizierung erweist sich insbesondere insofern als problematisch, als sie mit einer zunehmenden operativen Abhängigkeit von KI-gestützten Systemen einhergeht. Mit jeder erfolgreichen Interaktion verstärkt sich der Eindruck, dass das System schneller, besser oder verlässlicher arbeitet, wodurch die Motivation sinkt, alternative Lern- und Reflexionswege zu beschreiten. Was zunächst als pragmatische Arbeitserleichterung erlebt wird, führt mittelfristig zu einer Form epistemischer Trägheit: Wissen über Lösungswege wird nicht mehr aktiv erarbeitet, sondern passiv konsumiert. Mehrere Nutzerstimmen beschreiben diesen Effekt, indem sie auf eine subjektiv erlebte reduzierte Verständnissfähigkeit verweisen: „I feel like if everything is auto completed for you you don't get the same level of understanding you would writing it yourself” (Thread 6: 25). Hier wird deutlich, dass nicht nur ein Teil der Arbeit ausgelagert wird, sondern auch der Lernprozess, der beim selbstständigen Formulieren, Schreiben und Fehlerkorrigieren entsteht. Dann gelingt es einem auch nicht mehr, selbstständig an dem generierten Code weiterzuarbeiten. „For small projects, you're cheating yourself of the learning experience, or backing yourself into a corner where you can't go back and extend/debug/modify without significant trouble” (Thread 8: 35). Die Abhängigkeit von generierten Vorschlägen zeigt sich spätestens dann als problematisch, wenn die KI an Grenzen stößt. Ohne eigenes Fundament wird die technische Unterstützung zur unverzichtbaren Krücke: „At some point, you will hit a roadblock where even AI cannot help you. Then you will regret why didn't you learn about the library” (Thread 7: 26). Besonders zugespitzt wird diese Sorge in pessimistischen Prognosen artikuliert, die nicht nur individuelle Kompetenzverluste, sondern strukturelle Auswirkungen auf die Profession befürchten: „In fact, I believe LLMs will create a generation of shitty devs who can't actually reason through problems without it, and will create a tsunami of bugs that will require devs with critical thinking skills to solve” (Thread 4: 42).

Zusammenfassend konnten auf Grundlage der analysierten Reddit-Beiträge¹⁷ zwei zentrale Risiken identifiziert werden. Zum einen verweisen zahlreiche Aussagen auf Veränderungen ihres

¹⁷ Wichtig ist dabei festzuhalten, dass diese kritischen Aspekte von unterschiedlichen Usern artikuliert werden die nicht zwingend in direktem Zusammenhang miteinander stehen. Ebenso wenig lassen sich aus den Daten kausale Schlüsse dahingehend ziehen, dass die als autonomieerweiternd beschriebene Nutzung von KI als geschützten Lernraum oder Selbstkontrollinstrument zwangsläufig zu einem Verlust von

Denk und Arbeitsprozesses, welche aus den Affordanzen KI-gestützter Systeme resultieren. Durch kontinuierliche Vorschläge, automatische Vervollständigungen und implizite Handlungseinladungen wird Arbeitsprozess zunehmend vorstrukturiert. Zum anderen wird ein schleichender Verlust fachlicher Kompetenz thematisiert, der sich aus der fortlaufenden Nutzung KI-generierter Inhalte ergeben kann. Wenn vermeintlich plausible, aber fehlerhafte Lösungen nicht ausreichend reflektiert werden, besteht die Gefahr, dass sich diese in Lernprozessen verfestigen und langfristig zu einer operativen Abhängigkeit von KI-Systemen führen.

6.3 Grenzarbeit im Kontext KI-gestützter Softwareentwicklung

Zur theoretischen Einordnung der Ergebnisse bietet sich das Konzept der Grenzarbeit („*Boundary Work*“) nach Gieryn (1983) an, welches beschreibt, wie Akteure Grenzen ziehen, verschieben oder aushandeln, um berufliche Expertise und Verantwortlichkeiten zu definieren. Diese Grenzarbeit kann kooperative, kompetitive oder konfigurative Form annehmen (Langley et al. 2019). Diese kommen, wie Faulconbridge et al. (2023) feststellen im Kontext der Arbeit mit KI in verschränkter Form („*Intertwined Boundary Work*“) vor. Analysiert man die in dieser Arbeit empirisch gefundenen Aussagen entlang dieser drei Kategorien, wird dabei deutlich, dass die verschiedenen und miteinander verschränkten Formen der Grenzarbeit nicht nur *wegen der KI* gegenüber anderen Menschen stattfinden, etwa um Expertise zu verteidigen oder Verantwortlichkeiten neu zu verhandeln, sondern auch *gegenüber der KI*: Die Aushandlung betrifft hier nicht mehr nur zwischenmenschliche Rollen, sondern die Frage, welche Aufgaben, Entscheidungen und Deutungsmacht menschlichen Akteuren verbleiben und welche an KI-Systeme delegiert werden können. Grenzarbeit bezieht sich also nicht ausschließlich auf Beziehungen zwischen menschlichen Akteursgruppen, sondern zunehmend auch entlang menschlicher Arbeit gegenüber technologischen Systemen. Diese beiden Ebenen sollen in diesem Abschnitt anhand der empirischen Befunde gezeigt werden.

Grenzarbeit *zwischen* Menschen infolge der Einführung von KI zeigt sich beispielsweise dort, wo erfahrene Entwicklerinnen und Entwickler ihre professionelle Expertise gegenüber weniger qualifizierten Akteuren, etwa sogenannten „Prompt- oder Vibe-Codern“¹⁸, die durch KI neue Zugänge

Handlungsträgerschaft oder zu Abhängigkeitsverhältnissen führt. Die empirischen Befunde machen vielmehr die gleichzeitige Existenz unterschiedlicher, teils widersprüchlicher Erfahrungsberichte sichtbar, ohne deren Verhältnis abschließend festlegen zu können.

¹⁸ Mit „Prompt- oder Vibe-Codern“ werden Entwicklerinnen und Entwickler bezeichnet, die KI-gestützte Systeme primär durch natürlichsprachliche Eingaben (Prompts) nutzen, um Code zu generieren oder zu modifizieren; oft ohne die zugrunde liegenden Funktionsweisen, Logiken oder Strukturen des Codes im

zur Softwareentwicklung erhalten, behaupten. So lassen sich die in Abschnitt 5.2.2 (Deprofessionalisierung) formulierten Prognosen, wonach künftige Generationen den etablierten Berufsstandards nicht mehr entsprechen könnten, weil sie sich zu stark auf KI-generierte Ergebnisse stützen, als Ausdruck einer normativen Grenzziehung gegenüber anderen Personengruppen lesen. Die Technologie der KI steht in diesem Zusammenhang als Auslöser der Grenzarbeit. Ebenso zeigen die Aussagen aus Abschnitt 5.1.2 (Anforderungszuwachs), oder 5.2.4 (Anpassungsdruck) wie die User versuchen Kriterien fachlicher Kompetenz im Zuge der KI-Nutzung zu verhandeln. Die Fähigkeit, KI-Systeme sinnvoll und effizient einzusetzen, wird zunehmend selbst zum Maßstab professioneller Expertise. Professionelle Autorität legitimiert sich damit nicht mehr ausschließlich über tiefes, exklusives Wissen, sondern verstärkt über die Fähigkeit, Mensch-Maschine-Interaktionen kompetent zu gestalten und zu steuern. Der kompetente Umgang mit KI wird dabei zu einer normativen Erwartung, anhand derer sich professionelle Anerkennung bemisst: Nicht nur Programmierfähigkeit, sondern auch die Fähigkeit, KI zu „managen“, wird zum Kriterium legitimer Expertise. Die Kommentare suggerieren, dass man neue Fähigkeiten braucht und verorten damit die Grenze um den Beruf als Programmierer neu.

Im Unterschied zur Grenzarbeit zwischen Menschen richtet sich die Grenzarbeit *gegenüber KI* unmittelbar auf das Verhältnis zwischen menschlicher Arbeit und technologischen Systemen. Einige Entwicklerinnen und Entwickler positionieren sich beispielsweise gegenüber der Technologie, indem sie deren Leistungsfähigkeit relativieren oder die eigene fachliche Expertise hervorheben. Aussagen wie in Abschnitt 5.2.3 (Statusbewahrung), die betonen, dass KI den eigenen Beruf nicht ersetzen könne, lassen sich als defensive Jurisdiktionsansprüche im Sinne von Abbott (1988) interpretieren. Die KI wird hierbei als potenzielle Konkurrenz adressiert, gegenüber der sich die Akteure distinktiv positionieren. Diese Abgrenzung erfolgt häufig über die Zuschreibung genuin menschlicher Fähigkeiten wie Kreativität, Kontextsensitivität oder Urteilkraft, die der KI als nur eingeschränkt zugänglich dargestellt werden. Daneben finden sich Aussagen, in denen Grenzarbeit nicht gegen, sondern in Zusammenarbeit mit der KI, aber immer noch gegenüber der KI erfolgt. In den Aussagen zu 5.1.1 (Produktivitätsgewinne) wie auch in den Aussagen aus Abschnitt 5.2.1 (Kompetenzentwicklung) wird suggeriert, dass man die menschliche Arbeitskraft Zusammenarbeit augmentiert wird und damit die Aufgaben, die den Berufsstand ausmachen, erweitert werden. Das zeigt, dass die Nutzung KI-gestützter Systeme mit einer Verschiebung des

Detail nachzuvollziehen. Der Begriff „Vibe Coding“ wird dabei informell verwendet, um eine praxisorientierte, experimentelle Form der Softwareentwicklung zu beschreiben, bei der das iterative Ausprobieren und Anpassen von KI-generierten Vorschlägen im Vordergrund steht.

alltäglichen Aufgaben- und Tätigkeitsspektrums einhergeht. Besonders deutlich wird diese Dynamik in den in Abschnitt 5.3.3 (KI als Strukturierungsinstanz) beschriebenen Praktiken, die sich als konfigurative Grenzarbeit zwischen Mensch und Technologie interpretieren lassen. Die KI strukturiert den Arbeitsprozess zunehmend durch ihre Affordanzen (Hutchby 2001), indem sie nahelegt, welche Schritte als sinnvoll gelten, welche Optionen priorisiert werden und welche Ergebnisse als korrekt erscheinen. Der Mensch agiert in diesem Gefüge nicht mehr primär als steuernde Instanz, sondern übernimmt eine überwachende, prüfende und bestätigende Rolle. Seine Aufgabe besteht zunehmend darin, die von der KI vorgeschlagenen Handlungspfade zu bewerten, zu kontextualisieren und zu legitimieren.

7 Fazit

Die vorliegende Arbeit hat explorativ untersucht, wie Entwicklerinnen und Entwickler den Einsatz von KI in ihrer täglichen Arbeit auf der Plattform Reddit beschreiben, bewerten und aushandeln. Auf Grundlage einer netnographischen Analyse wurden wiederkehrende Deutungsmuster und Praktiken rekonstruiert. Aus der Kartierung dieser Beschreibungen lässt sich übergreifend ableiten, dass KI in der professionellen Softwareentwicklung nicht additiv wirkt, sondern restrukturierend. Die Implementierung von KI verändert alltägliche Tätigkeiten, Lernpraktiken und kollegiale Kommunikation nicht punktuell oder ergänzend, sondern greift in deren Zusammenspiel als Ganzes ein. Produktivitätsgewinne, Lernchancen und neue Formen der Unterstützung gehen systematisch mit einer Verdichtung von Arbeit, steigenden Anforderungen an Kontrolle und neuen Abhängigkeiten einher.

Ein zentral zu hervorgebendes Ergebnis der empirischen Befunde betrifft die Nutzung von KI-Systemen als Form des stellvertretenden Managements. Einige Entwicklerinnen und Entwickler greifen auf KI-Systeme zurück, um sich Handlungsschritte vorschlagen zu lassen, den eigenen Code zu überprüfen oder ihre Entscheidungsprozesse zu validieren. Damit schreiben sie der KI typische Managementaufgaben zu. Diese Praxis als Selbstkontrolle durch KI wird dabei teilweise wie ein Akt der Selbstermächtigung gerahmt, da sie ein Gefühl von Sicherheit, Orientierung und Autonomie vermittelt. Charakteristische Merkmale algorithmischer Kontrolle (Kellogg et al., 2020) werden hier positiv gewendet und produktiv genutzt. Die KI fungiert als jederzeit verfügbare, nicht wertende und geduldige Instanz, die es ermöglicht, Unsicherheiten offen zu artikulieren und Fehler zu machen, ohne soziale Sanktionen befürchten zu müssen. Auf diese Weise entsteht ein digitaler und geschützter Raum des Lernens, Ausprobierens und Experimentierens.

Gleichzeitig zeigt sich eine ambivalente Dynamik: Indem KI-Systeme den Arbeitsprozess durch ihre *Affordanzen* (Hutchby, 2001) vorstrukturieren, nehmen sie zunehmend Einfluss auf die Prozesssteuerung. Nutzerinnen und Nutzer folgen den Vorschlägen der KI oft automatisch, ohne deren Logik oder Begründung vollständig nachzuvollziehen. Der Arbeitsfluss wird damit weniger aktiv gestaltet, sondern stärker von der Technologie gelenkt. Auf diese Weise verschiebt sich die Kontrolle subtil in Richtung der KI, während zentrale Lernmomente wie etwa das selbstständige Schreiben, Reflektieren und Überarbeiten von Code zunehmend wegfallen. Auch traditionelle soziale Korrektive wie kollegiales Feedback oder gemeinschaftliche Lernprozesse, die bislang ein wichtiger Bestandteil kollektiver Wissensproduktion waren, werden teilweise umgangen. Da viele Prozesse so nicht mehr vollständig durchdrungen oder verstanden werden können, verlieren einige Entwicklerinnen und Entwickler das Gefühl, den Arbeitsablauf selbst kontrollieren zu können. In den beobachteten Diskursen äußern einige sogar ein Gefühl der Ohnmacht, wenn es darum geht, den von der KI erzeugten Code gezielt und intentional zu verändern. Sie erleben sich als von ihrem eigenen Schaffensprozess entfremdet oder äußern eine steigende Abhängigkeit.

Unter dem Titel „*Kontrolle ohne Korrektiv*“ lässt sich eine zentrale Ambivalenz im Umgang mit generativer KI bündeln. Einerseits nutzen Entwicklerinnen und Entwickler KI-Systeme gezielt als Kontrollinstanz, um ihre Arbeit zu überprüfen, Entscheidungen abzusichern und Unsicherheiten zu reduzieren. Diese Form der Kontrolle greift jedoch nur begrenzt korrektiv in die Arbeitspraxis ein, da sie aufgrund der Architektur von KI-Systemen weitgehend ohne Widerspruch, Irritation oder soziale Aushandlung auskommt und damit zentrale Momente von Kritik und Korrektur suspendiert. Andererseits greift KI zugleich kontinuierlich in steuernder und strukturierender Weise in den Arbeitsprozess ein, indem sie Vorschläge priorisiert, Handlungsschritte vorordnet und Problemlösungen rahmt. Die nun reflexive Aufgabe der Entwicklerinnen und Entwickler, diese zu prüfen und kontextualisieren zeigt sich in der Praxis an einigen Stellen immer seltener wirklich korrektiv, da algorithmische Vorschläge zunehmend routiniert übernommen werden, ohne sie systematisch zu hinterfragen. Während KI den Arbeitsprozess fortlaufend steuert und „kontrolliert“, kann die Bereitschaft und Fähigkeit zur kritischen Auseinandersetzung mit ihren Vorgaben abnehmen.

Zusammenfassend macht die vorliegende Arbeit deutlich, dass der Einsatz generativer KI in der professionellen Softwareentwicklung einen tiefgreifenden Strukturwandel professioneller Praxis markiert. KI wirkt nicht additiv, sondern restrukturierend, indem sie Arbeitsprozesse, Lernformen und professionelle Selbstverständnisse in ihrem Zusammenspiel neu ordnet. Besonders die Nutzung von KI als algorithmisch vermittelte Form der Selbstkontrolle verweist auf eine zentrale Ambivalenz: Während KI als entlastende, orientierende und scheinbar objektive Instanz erfahren

wird, entzieht sie professioneller Praxis zugleich jene sozialen und reflexiven Korrektive, die für Kritik, Verantwortungsübernahme und kollektive Qualitätssicherung konstitutiv sind. Unter dem Begriff „*Kontrolle ohne Korrektiv*“ verdichten sich die Befunde zu der Einsicht, dass Kontrolle subjektiv als intensiviert wahrgenommen werden kann, zugleich jedoch von ihrer kritisch-reflexiven und sozialen Dimension entkoppelt wird.

Aus normativer Perspektive wirft diese Entwicklung grundlegende Fragen nach der Zukunft professioneller Arbeit unter KI-Bedingungen auf. Wenn Kontrolle zunehmend algorithmisch vermittelt und individualisiert wird, stellt sich die Frage, wie professionelle Korrektive, kritische Reflexion und kollektive Verantwortungsstrukturen langfristig aufrechterhalten werden können. Zukünftige Forschung sollte daher nicht nur untersuchen, *ob* KI Arbeitsprozesse effizienter gestaltet, sondern unter welchen Bedingungen sie professionelles Handeln im Sinne reflexiver Urteilskraft und verantwortlicher Praxis unterstützt oder unterminiert. Darüber hinaus bedarf es einer systematischen Auseinandersetzung mit Gestaltungsprinzipien für KI-gestützte Arbeit, die soziale Aushandlung, Widerspruch und Kritik nicht ersetzen, sondern gezielt integrieren. Schließlich verweisen die beobachteten Einflüsse auf die Notwendigkeit, professionelle Standards, organisationale Rahmenbedingungen und ethische Leitlinien neu zu denken, um sicherzustellen, dass KI nicht zur stillen Instanz normativer Setzung wird, sondern als gestaltbarer Bestandteil professioneller Praxis begriffen bleibt.

8 Literaturverzeichnis

- 11 *Best Programming Forums*. (2025). Daily Dev. Abgerufen am 15. Januar 2025, unter <https://daily.dev/blog/11-best-programming-forums-2024>
- Abbott, A. (1988). *The system of professions: An essay on the division of expert labor* (S. xvi, 435). The University of Chicago Press.
- Abdalkareem, R., Shihab, E., & Rilling, J. (2017). What Do Developers Use the Crowd For? A Study Using Stack Overflow. *IEEE Software*, 34(2), 53–60. <https://doi.org/10.1109/MS.2017.31>
- About GitHub*. (2025). GitHub. Abgerufen am 3. September 2025, unter <https://github.com/about>
- Abramson, J., Ahuja, A., Barr, I., Brussee, A., Carnevale, F., Cassin, M., Chhaparia, R., Clark, S., Damoc, B., Dudzik, A., Georgiev, P., Guy, A., Harley, T., Hill, F., Hung, A., Kenton, Z., Landon, J., Lillicrap, T., Mathewson, K., ... Zhu, R. (2021). *Imitating Interactive Intelligence* (No. arXiv:2012.05672). arXiv. <https://doi.org/10.48550/arXiv.2012.05672>
- Arntz, M., Gregory, T., & Zierahn, U. (2016, Mai 13). *The Risk of Automation for Jobs in OECD Countries. A Comparative Analyses*. OECD. https://www.oecd.org/en/publications/the-risk-of-automation-for-jobs-in-oecd-countries_5jlz9h56dvq7-en.html
- Atkinson, D. P., & Barker, D. R. (2023). AI and the social construction of creativity. *Convergence*, 29(4), 1054–1069. <https://doi.org/10.1177/13548565231187730>
- Atkinson, P., Coffey, A., Delamont, S., Lofland, J., & Lofland, L. (2001). *Handbook of Ethnography*. SAGE Publications Ltd. <https://doi.org/10.4135/9781848608337>
- Autor, D. H. (2015). Why Are There Still So Many Jobs? The History and Future of Workplace Automation. *Journal of Economic Perspectives*, 29(3), 3–30. <https://doi.org/10.1257/jep.29.3.3>
- Autor, D. H., Levy, F., & Murnane, R. J. (2003). The Skill Content of Recent Technological Change: An Empirical Exploration. *The Quarterly Journal of Economics*, 118(4), 1279–1333. <https://doi.org/10.1162/003355303322552801>
- Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., Joseph, N., Kadavath, S., Kernion, J., Conerly, T., El-Showk, S., Elhage, N., Hatfield-Dodds, Z., Hernandez, D., Hume, T., ... Kaplan, J. (2022). *Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback* (No. arXiv:2204.05862). arXiv. <https://doi.org/10.48550/arXiv.2204.05862>
- Bakshy, E., Messing, S., & Adamic, L. A. (2015). Exposure to ideologically diverse news and opinion on Facebook. *Science*, 348(6239), 1130–1132. <https://doi.org/10.1126/science.aaa1160>

- Bardol, F. (2025). *ChatGPT Reads Your Tone and Responds Accordingly—Until It Does Not—Emotional Framing Induces Bias in LLM Outputs* (No. arXiv:2507.21083). arXiv. <https://doi.org/10.48550/arXiv.2507.21083>
- Barley, S. R. (1986). Technology as an Occasion for Structuring: Evidence from Observations of CT Scanners and the Social Order of Radiology Departments. *Administrative Science Quarterly*, 31(1), 78–108. <https://doi.org/10.2307/2392767>
- Barua, A., Thomas, S. W., & Hassan, A. E. (2014). What are developers talking about? An analysis of topics and trends in Stack Overflow. *Empirical Software Engineering*, 19(3), 619–654. <https://doi.org/10.1007/s10664-012-9231-y>
- Becker, J., Rush, N., Barnes, E., & Rein, D. (2025). *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity* (No. arXiv:2507.09089). arXiv. <https://doi.org/10.48550/arXiv.2507.09089>
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer. <https://link.springer.com/9780387310732>
- Boltanski, L., & Chiapello, È. (2006). *Der neue Geist des Kapitalismus* (1. Auflage). Herbert von Halem.
- Braganza, A., Chen, W., Canhoto, A., & Sap, S. (2021). Productive employment and decent work: The impact of AI adoption on psychological contracts, job engagement and employee trust. *Journal of Business Research*, 131, 485–494. <https://doi.org/10.1016/j.jbusres.2020.08.018>
- Braverman, H. (1998). *Labor and Monopoly Capital: The Degradation of Work in the Twentieth Century*. NYU Press. <https://www.jstor.org/stable/j.ctt9qfrkf>
- Bretthauer, D. (2001). Open Source Software: A History. *Published Works*. https://digitalcommons.lib.uconn.edu/libr_pubs/7
- Brubaker, P. J., Montez, D., & Church, S. H. (2021). The Power of Schadenfreude: Predicting Behaviors and Perceptions of Trolling Among Reddit Users. *Social Media + Society*, 7(2), 20563051211021382. <https://doi.org/10.1177/20563051211021382>
- Brynjolfsson, E., Li, D., & Raymond, L. (2025). Generative AI at Work*. *The Quarterly Journal of Economics*, 140(2), 889–942. <https://doi.org/10.1093/qje/qjae044>
- Butollo, F. (2024). *The Rebound Effects of Automation*. Weizenbaum Institute. <https://doi.org/10.34669/WI.WS/37>
- CCC | *Hacker Ethics*. (o.J.). Chaos Computer Club. Abgerufen am 3. September 2025, unter <https://www.ccc.de/en/hackerethik>

- Chang, H.-F., & Shokrolah Shirazi, M. (2025). A Systematic Approach for Assessing Large Language Models' Test Case Generation Capability. *Software*, 4(1), 5. <https://doi.org/10.3390/software4010005>
- Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. de O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., ... Zaremba, W. (2021). *Evaluating Large Language Models Trained on Code* (No. arXiv:2107.03374). arXiv. <https://doi.org/10.48550/arXiv.2107.03374>
- Cheng, J., Bernstein, M., Danescu-Niculescu-Mizil, C., & Leskovec, J. (2017). Anyone Can Become a Troll: Causes of Trolling Behavior in Online Discussions. *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, 1217–1230. <https://doi.org/10.1145/2998181.2998213>
- Cheriyian, J., Savarimuthu, B. T. R., & Cranefield, S. (2021). *Norm violation in online communities—A study of Stack Overflow comments* (Bd. 12298, S. 20–34). https://doi.org/10.1007/978-3-030-72376-7_2
- Conover, M., Ratkiewicz, J., Francisco, M., Goncalves, B., Menczer, F., & Flammini, A. (2011). Political Polarization on Twitter. *Proceedings of the International AAAI Conference on Web and Social Media*, 5(1), Article 1. <https://doi.org/10.1609/icwsm.v5i1.14126>
- Cuskley, C., Woods, R., & Flaherty, M. (2024). The Limitations of Large Language Models for Understanding Human Language and Cognition. *Open Mind*, 8, 1058–1083. https://doi.org/10.1162/opmi_a_00160
- Datta, S., & Adar, E. (2019). Extracting Inter-Community Conflicts in Reddit. *Proceedings of the International AAAI Conference on Web and Social Media*, 13, 146–157. <https://doi.org/10.1609/icwsm.v13i01.3217>
- Dell'Acqua, F., McFowland III, E., Mollick, E. R., Lifshitz-Assaf, H., Kellogg, K., Rajendran, S., Kraye, L., Candelon, F., & Lakhani, K. R. (2023). *Navigating the Jagged Technological Frontier: Field Experimental Evidence of the Effects of AI on Knowledge Worker Productivity and Quality* (SSRN Scholarly Paper No. 4573321). Social Science Research Network. <https://doi.org/10.2139/ssrn.4573321>
- DiBona, C., & Ockman, S. (1999). *Open Sources: Voices from the Open Source Revolution*. O'Reilly Media, Inc.
- Driver jobs in Berlin—Choose Your Hours*. (o. J.). Uber. Abgerufen 26. Oktober 2025, von <https://www.uber.com/de/en/e/drive/berlin-be-de/>
- Edwards, R. (1979). *Contested Terrain*. Basic Books.
- Elmaresa, M. V., & Irwansyah. (2025). Human and Artificial Intelligence Interaction from the Perspective of Social Construction of Technology. *EKSPRESI DAN PERSEPSI : JURNAL ILMU KOMUNIKASI*, 8(1), 52–64. <https://doi.org/10.33822/jep.v8i1.9099>

- Eloundou, T., Manning, S., Mishkin, P., & Rock, D. (2023). *GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models – Own Your AI*. ownyourai. <https://ownyourai.com/gpts-are-gpts-an-early-look-at-the-labor-market-impact-potential-of-large-language-models/>
- Evetts, J. (2014). The Concept of Professionalism: Professional Work, Professional Practice and Learning. In S. Billett, C. Harteis, & H. Gruber (Hrsg.), *International Handbook of Research in Professional and Practice-based Learning* (S. 29–56). Springer Netherlands. https://doi.org/10.1007/978-94-017-8902-8_2
- Faulconbridge, J., Sarwar, A., & Spring, M. (2023). How Professionals Adapt to Artificial Intelligence: The Role of Intertwined Boundary Work. *Journal of Management Studies*, n/a(n/a). <https://doi.org/10.1111/joms.12936>
- Festinger, L. (1962). Cognitive Dissonance. *Scientific American*, 207(4), 93–106.
- Freidson, E. (1984). The Changing Nature of Professional Control. *Annual Review of Sociology*, 10(Volume 10, 1984), 1–20. <https://doi.org/10.1146/annurev.so.10.080184.000245>
- Freidson, E. (2017). *Professional Dominance: The Social Structure of Medical Care*. Routledge.
- Frey, C. B., & Osborne, M. A. (2017). The future of employment: How susceptible are jobs to computerisation? *Technological Forecasting and Social Change*, 114, 254–280. <https://doi.org/10.1016/j.techfore.2016.08.019>
- Garrett, R. K. (2009). Echo chambers online?: Politically motivated selective exposure among Internet news users. *Journal of Computer-Mediated Communication*, 14(2), 265–285. <https://doi.org/10.1111/j.1083-6101.2009.01440.x>
- Gen AI in software. (2024, Dezember 20). *Capgemini UK*. <https://www.capgemini.com/gb-en/insights/research-library/gen-ai-in-software/>
- Gieryn, T. F. (1983). Boundary-Work and the Demarcation of Science from Non-Science: Strains and Interests in Professional Ideologies of Scientists. *American Sociological Review*, 48(6), 781–795. <https://doi.org/10.2307/2095325>
- González Barman, K., Lohse, S., & De Regt, H. W. (2025). Reinforcement Learning from Human Feedback in LLMs: Whose Culture, Whose Values, Whose Perspectives? *Philosophy & Technology*, 38(2), 35. <https://doi.org/10.1007/s13347-025-00861-0>
- Goos, M., Manning, A., & Salomons, A. (2014). Explaining Job Polarization: Routine-Biased Technological Change and Offshoring. *American Economic Review*, 104(8), 2509–2526. <https://doi.org/10.1257/aer.104.8.2509>
- Hannak, A., Sapiezynski, P., Molavi Kakhki, A., Krishnamurthy, B., Lazer, D., Mislove, A., & Wilson, C. (2013). Measuring personalization of web search. *Proceedings of the 22nd*

- Heiland, H. (2020). *Workers' Voice in platform labour: An Overview* (Research Report No. 21). WSI Study. <https://www.econstor.eu/handle/10419/225444>
- Heiland, H., & Schaupp, S. (2020). Digitale Atomisierung oder neue Arbeitskämpfe? Widerständige Solidaritätskulturen in der plattformvermittelten Kurierarbeit. *Momentum Quarterly*, 50–67. <https://doi.org/10.15203/momentumquarterly.vol9.no2.p50-67>
- Heßler, M. (2014). Die Halle 54 bei Volkswagen und die Grenzen der Automatisierung. Überlegungen zum Mensch-Maschine-Verhältnis in der industriellen Produktion der 1980er-Jahre. *Zeithistorische Forschungen - Studies in Contemporary History*, 11(1), 56–76. <https://doi.org/10.14765/zzf.dok-1495>
- Hirsch-Kreinsen, H. (2016). Digitization of industrial work: Development paths and prospects. *Journal for Labour Market Research*, 49(1), 1–14. <https://doi.org/10.1007/s12651-016-0200-6>
- Hoda, R., Noble, J., & Marshall, S. (2013). Self-Organizing Roles on Agile Software Development Teams. *IEEE Transactions on Software Engineering*, 39(3), 422–444. <https://doi.org/10.1109/TSE.2012.30>
- Hoffmann, C. H. (2022). Is AI intelligent? An assessment of artificial intelligence, 70 years after Turing. *Technology in Society*, 68, 101893. <https://doi.org/10.1016/j.tech-soc.2022.101893>
- Honneth, A. (2018). *Anerkennung: Eine europäische Ideengeschichte*. Suhrkamp Verlag.
- Husein, R. A., Aburajouh, H., & Catal, C. (2025). Large language models for code completion: A systematic literature review. *Computer Standards & Interfaces*, 92, 103917. <https://doi.org/10.1016/j.csi.2024.103917>
- Hutchby, I. (2001). Technologies, Texts and Affordances. *Sociology*, 35(2), 441–456. <https://doi.org/10.1177/S0038038501000219>
- Imai, S. (2022). Is GitHub copilot a substitute for human pair-programming? An empirical study. *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings*, 319–321. <https://doi.org/10.1145/3510454.3522684>
- Jamieson, K. H., & Cappella, J. N. (2008). *Echo Chamber: Rush Limbaugh and the Conservative Media Establishment*. Oxford University Press.
- Jemielniak, D. (2008). Software engineers or artists? Programmers' identity choices. *Tamara Journal*, 7.

- Jiang, W., Park, J., Xiao, R. (Jiqui), & Zhang, S. (2025). *AI and the Extended Workday: Productivity, Contracting Efficiency, and Distribution of Rents* (No. w33536). National Bureau of Economic Research. <https://doi.org/10.3386/w33536>
- Jordan, M., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245). <https://doi.org/10.1126/science.aaa8415>
- Kellogg, K. C., Valentine, M. A., & Christin, A. (2020). Algorithms at Work: The New Contested Terrain of Control. *Academy of Management Annals*, 14(1), 366–410. <https://doi.org/10.5465/annals.2018.0174>
- Kenney, M., & Zysman, J. (2016, April 1). The Rise of the Platform Economy. *Issues in Science and Technology*. <https://issues.org/rise-platform-economy-big-data-work/>
- Knoblauch, H. (2014). Ethnographie. In N. Baur & J. Blasius (Hrsg.), *Handbuch Methoden der empirischen Sozialforschung* (S. 521–528). Springer Fachmedien. https://doi.org/10.1007/978-3-531-18939-0_36
- Korteling, J. E. (Hans), van de Boer-Visschedijk, G. C., Blankendaal, R. a. M., Boonekamp, R. C., & Eikelboom, A. R. (2021). Human- versus Artificial Intelligence. *Frontiers in Artificial Intelligence*, 4. <https://doi.org/10.3389/frai.2021.622364>
- Kozinets, R. V. (2015). *Netnography: Redefined*. SAGE.
- Kozinets, R. V. (2019). *Netnography: The Essential Guide to Qualitative Social Media Research* (3. Aufl.). SAGE Publications Ltd. <https://www.perlego.com/book/1431879/netnography-the-essential-guide-to-qualitative-social-media-research-pdf>
- Krishtul, S., Shahin, M., Obie, H. O., Khalajzadeh, H., Gai, F., Nasab, A. R., & Grundy, J. (2022). *Human Values Violations in Stack Overflow: An Exploratory Study* (No. arXiv:2203.10551). arXiv. <https://doi.org/10.48550/arXiv.2203.10551>
- Krzywdzinski, M., Pfeiffer, S., Kuhlmann, M., Ottaiano, M., Heinlein, M., Ritter, T., Neumer, J., & Huchler, N. (2025). An uncertain elite: Professional differences and similarities between engineers and tech workers in times of digital transformation. *Work Organisation, Labour & Globalisation*, 19(1). <https://doi.org/10.13169/workorglaboglob.19.1.0084>
- Langley, Lindberg, K., Mørk, B. E., Nicolini, D., Raviola, E., & Walter, L. (2019). Boundary Work among Groups, Occupations, and Organizations: From Cartography to Process. *Academy of Management Annals*, 13(2), 704–736. <https://doi.org/10.5465/annals.2017.0089>
- Langley, P., & Leyshon, A. (2017). Platform capitalism: The intermediation and capitalisation of digital economic circulation. *Finance and Society*, 3(1), 11–31. <https://doi.org/10.2218/finsoc.v3i1.1936>

- Lapidot-Lefler, N., & Barak, A. (2012). Effects of anonymity, invisibility, and lack of eye-contact on toxic online disinhibition. *Computers in Human Behavior*, 28(2), 434–443. <https://doi.org/10.1016/j.chb.2011.10.014>
- Latour, B. (2021). *Der Berliner Schlüssel* (G. Roßler, Übers.; Fünfte Auflage). Botopress.
- Leonardi, P. M., & Barley, W. (2011). Materiality as organizational communication: Technology, intent, and delegation in the production of meaning. In *Matters of communication: Political, cultural, and technological challenges to communication theorizing* (S. 101–124). Hampton Press.
- Levy, S. (1984). *Hackers: Heroes of the computer revolution* (1st ed). Anchor Press/Doubleday.
- Liang, J. T., Yang, C., & Myers, B. A. (2024). A Large-Scale Survey on the Usability of AI Programming Assistants: Successes and Challenges. *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, 1–13. <https://doi.org/10.1145/3597503.3608128>
- Lin, W., Gerchanovsky, A., Akgul, O., Bauer, L., Fredrikson, M., & Wang, Z. (2025). *LLM Whisperer: An Inconspicuous Attack to Bias LLM Responses* (No. arXiv:2406.04755). arXiv. <https://doi.org/10.48550/arXiv.2406.04755>
- Liu, Y., & Anwar, M. (2022). Learning Programming in Social Media: An NLP-powered Reddit Study. *2022 Fourth International Conference on Transdisciplinary AI (TransAI)*, 55–58. <https://doi.org/10.1109/TransAI54797.2022.00015>
- Lühr, T., Ziegler, A., Vogl, E., & Boes, A. (2020). *#UmbruchErleben: Wie erleben die Menschen die digitale Transformation?* [Pdf]. 96 Seiten. <https://doi.org/10.35067/XYPQ-KN61>
- Luzsa, R., & Mayr, S. (2021). False consensus in the echo chamber: Exposure to favorably biased social media news feeds leads to increased perception of public support for own opinions. *Cyberpsychology: Journal of Psychosocial Research on Cyberspace*, 15(1), Article 1. <https://doi.org/10.5817/CP2021-1-3>
- Macke, W., & Doyle, M. (2024). Testing the Effect of Code Documentation on Large Language Model Code Understanding. In K. Duh, H. Gomez, & S. Bethard (Hrsg.), *Findings of the Association for Computational Linguistics: NAACL 2024* (S. 1044–1050). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2024.findings-naacl.66>
- Mancuso, S. (2015). *The software craftsman: Professionalism, pragmatism, pride*. Prentice Hall.
- Manokha, I. (2018). Surveillance, panopticism, and self-discipline in the digital age. *Surveillance and Society*, 16(2). <https://ora.ox.ac.uk/objects/uuid:a8f5e604-0e3e-42d2-b373-a4e650b39dcb>
- Mastropaolo, A., Pascarella, L., Guglielmi, E., Ciniselli, M., Scalabrino, S., Oliveto, R., & Bavota, G. (2023). On the Robustness of Code Generation Techniques: An Empirical Study on

- GitHub Copilot. 2023 *IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, 2149–2160. <https://doi.org/10.1109/ICSE48619.2023.00181>
- McBreen, P. (2002). *Software Craftsmanship: The New Imperative*. Addison-Wesley Professional.
- Meincke, L., Mollick, E. R., Mollick, L., & Shapiro, D. (2025). *Prompting Science Report 1: Prompt Engineering is Complicated and Contingent* (SSRN Scholarly Paper No. 5165270). Social Science Research Network. <https://doi.org/10.2139/ssrn.5165270>
- Moore, P., & Robinson, A. (2016). The quantified self: What counts in the neoliberal workplace. *New Media & Society*, 18(11), 2774–2792. <https://doi.org/10.1177/1461444815604328>
- Nickerson, R. S. (1998). Confirmation Bias: A Ubiquitous Phenomenon in Many Guises. *Review of General Psychology*, 2(2), 175–220. <https://doi.org/10.1037/1089-2680.2.2.175>
- Noordegraaf, M. (2015). Hybrid professionalism and beyond: (New) Forms of public professionalism in changing organizational and societal contexts. *Journal of Professions and Organization*, 2(2), 187–206. <https://doi.org/10.1093/jpo/jov002>
- Noy, S., & Zhang, W. (2023). Experimental evidence on the productivity effects of generative artificial intelligence. *Science*, 381(6654), 187–192. <https://doi.org/10.1126/science.adh2586>
- Ohagi, M. (2024). *Polarization of Autonomous Generative AI Agents Under Echo Chambers* (No. arXiv:2402.12212). arXiv. <https://doi.org/10.48550/arXiv.2402.12212>
- Orlikowski, W. J. (2007). Sociomaterial Practices: Exploring Technology at Work. *Organization Studies*, 28(9), 1435–1448. <https://doi.org/10.1177/0170840607081138>
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., & Lowe, R. (2022). *Training language models to follow instructions with human feedback*.
- Pan, X., Fan, J., Xiong, Z., Hahami, E., Overwiening, J., & Xie, Z. (2025). *User-Assistant Bias in LLMs* (No. arXiv:2508.15815). arXiv. <https://doi.org/10.48550/arXiv.2508.15815>
- Pariser, E. (2011). *The Filter Bubble: How the New Personalized Web Is Changing What We Read and How We Think*. Penguin.
- Parsons, T. (1991). *The social system* (2nd ed). Taylor and Francis.
- Peng, S., Kalliamvakou, E., Cihon, P., & Demirer, M. (2023). *The Impact of AI on Developer Productivity: Evidence from GitHub Copilot* (No. arXiv:2302.06590). arXiv. <https://doi.org/10.48550/arXiv.2302.06590>

- Pfeiffer, S. (2010). Kapitel IV Rationalisierung von Arbeit: Technisierung von Arbeit. In F. Böhle, G. G. Voß, & G. Wachtler (Hrsg.), *Handbuch Arbeitssoziologie* (S. 231–261). VS Verlag für Sozialwissenschaften. https://doi.org/10.1007/978-3-531-92247-8_8
- Pfeiffer, S. (2020). Kontext und KI: Zum Potenzial der Beschäftigten für Künstliche Intelligenz und Machine-Learning. *HMD Praxis der Wirtschaftsinformatik*, 57(3), 465–479. <https://doi.org/10.1365/s40702-020-00609-8>
- Pignot, E. (2023). Who is pulling the strings in the platform economy? Accounting for the dark and unexpected sides of algorithmic control. *Organization*, 30(1), 140–167. <https://doi.org/10.1177/1350508420974523>
- Pinch, T. J., & Bijker, W. E. (1984). The Social Construction of Facts and Artefacts: Or How the Sociology of Science and the Sociology of Technology might Benefit Each Other. *Social Studies of Science*, 14(3), 399–441. <https://doi.org/10.1177/030631284014003004>
- Raymond, E. S. (1998, März 2). *The cathedral and the bazaar* [Text]. Valauskas, Edward J. <https://firstmonday.org/ojs/index.php/fm/article/download/1472/1387?inline=1>
- Russell, S. J., & Norvig, P. (1995). *Artificial intelligence: A modern approach*. Prentice Hall.
- Saeed, M. H., Ali, S., Blackburn, J., Cristofaro, E. D., Zannettou, S., & Stringhini, G. (2022). Troll-Magnifier: Detecting State-Sponsored Troll Accounts on Reddit. *2022 IEEE Symposium on Security and Privacy (SP)*, 2161–2175. <https://doi.org/10.1109/SP46214.2022.9833706>
- Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2025). A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications (No. arXiv:2402.07927). arXiv. <https://doi.org/10.48550/arXiv.2402.07927>
- Saks, M. (2012). Defining a Profession: The Role of Knowledge and Expertise. *Professions and Professionalism*, 2(1). <https://doi.org/10.7577/pp.v2i1.151>
- Santos, R. de S., Adisaputri, G., & Ralph, P. (2023). *Post-pandemic Resilience of Hybrid Software Teams* (No. arXiv:2303.06215). arXiv. <https://doi.org/10.48550/arXiv.2303.06215>
- Schäfer, M., Nadi, S., Eghbali, A., & Tip, F. (2024). An Empirical Evaluation of Using Large Language Models for Automated Unit Test Generation. *IEEE Transactions on Software Engineering*, 50(1), 85–105. <https://doi.org/10.1109/TSE.2023.3334955>
- Schmitt, A., Gajos, K. Z., & Mokryn, O. (2024). *Generative AI in the Software Engineering Domain: Tensions of Occupational Identity and Patterns of Identity Protection* (No. arXiv:2410.03571). arXiv. <https://doi.org/10.48550/arXiv.2410.03571>
- Schulz-Schaeffer, I. (2025). Why generative AI is different from designed technology regarding task-relatedness, user interaction, and agency. *Big Data & Society*, 12(3), 20539517251367452. <https://doi.org/10.1177/20539517251367452>

- Sennett, R. (1998). *Corrosion of Character: The Personal Consequences Of Work In The New Capitalism*. W. W. Norton & Company.
- Sharma, Liao, Q. V., & Xiao, Z. (2024). Generative Echo Chamber? Effect of LLM-Powered Search Systems on Diverse Information Seeking. *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 1–17. <https://doi.org/10.1145/3613904.3642459>
- Sharma, Tong, M., Korbak, T., Duvenaud, D., Askell, A., Bowman, S. R., Cheng, N., Durmus, E., Hatfield-Dodds, Z., Johnston, S. R., Kravec, S., Maxwell, T., McCandlish, S., Ndousse, K., Rausch, O., Schiefer, N., Yan, D., Zhang, M., & Perez, E. (2025). *Towards Understanding Sycophancy in Language Models* (No. arXiv:2310.13548). arXiv. <https://doi.org/10.48550/arXiv.2310.13548>
- Sharp, H., & Robinson, H. (2008). Collaboration and co-ordination in mature eXtreme programming teams. *International Journal of Human-Computer Studies*, 66(7), 506–518. <https://doi.org/10.1016/j.ijhcs.2007.10.004>
- Shi, E., Wang, Y., Du, L., Chen, J., Han, S., Zhang, H., Zhang, D., & Sun, H. (2022). On the evaluation of neural code summarization. *Proceedings of the 44th International Conference on Software Engineering*, 1597–1608. <https://doi.org/10.1145/3510003.3510060>
- Spicer, A. (2003). *Transforming Technology: A Critical Theory Revisited* (2nd ed.) 20032 Andrew Feenberg. *Transforming Technology: A Critical Theory Revisited* (2nd ed.) . Oxford: Oxford University Press 2002. *Information Technology & People*, 16, 378–383. <https://doi.org/10.1108/itp.2003.16.3.378.2>
- Srnicek, N. (2017). *Platform Capitalism*. John Wiley & Sons.
- Suler, J. (2004). The Online Disinhibition Effect. *Cyberpsychology & behavior : the impact of the Internet, multimedia and virtual reality on behavior and society*, 7, 321–326. <https://doi.org/10.1089/1094931041291295>
- Sun, Y., & Kok, S. (2025). *Investigating the Effects of Cognitive Biases in Prompts on Large Language Model Outputs* (No. arXiv:2506.12338). arXiv. <https://doi.org/10.48550/arXiv.2506.12338>
- Sundelin, A., Gonzalez-huerta, J., Wnuk, K., & Gorschek, T. (2021). Towards an Anatomy of Software Craftsmanship. *ACM Trans. Softw. Eng. Methodol.*, 31(1), 6:1-6:49. <https://doi.org/10.1145/3468504>
- Sunstein, C. (2002). The Law of Group Polarization. *Journal of Political Philosophy*. <https://doi.org/10.1111/1467-9760.00148>
- Tozzi, C. (2017). *For Fun and Profit: A History of the Free and Open Source Software Revolution*. The MIT Press. <https://doi.org/10.7551/mitpress/10803.001.0001>

- Treude, C., Barzilay, O., & Storey, M.-A. (2011). How do programmers ask and answer questions on the web? (NIER track). *Proceedings of the 33rd International Conference on Software Engineering*, 804–807. <https://doi.org/10.1145/1985793.1985907>
- Wason, P. C. (1960). On the Failure to Eliminate Hypotheses in a Conceptual Task. *Quarterly Journal of Experimental Psychology*, 12(3), 129–140. <https://doi.org/10.1080/17470216008416717>
- Wason, P. C. (1968). Reasoning about a Rule. *Quarterly Journal of Experimental Psychology*, 20(3), 273–281. <https://doi.org/10.1080/14640746808400161>
- Winner, L. (2007). Do Artifacts Have Politics? In *Computer Ethics*. Routledge.
- Wu, Y., Wang, S., Bezemer, C.-P., & Inoue, K. (2019). How do developers utilize source code from stack overflow? *Empirical Software Engineering*, 24(2), 637–673. <https://doi.org/10.1007/s10664-018-9634-5>
- Xiao, J., Li, Z., Xie, X., Getzen, E., Fang, C., Long, Q., & Su, W. J. (2025). *On the Algorithmic Bias of Aligning Large Language Models with RLHF: Preference Collapse and Matching Regularization* (No. arXiv:2405.16455). arXiv. <https://doi.org/10.48550/arXiv.2405.16455>
- Yeung, K. (2018). Algorithmic regulation: A critical interrogation. *Regulation & Governance*, 12(4), 505–523. <https://doi.org/10.1111/rego.12158>
- Yuan, Z., Liu, M., Ding, S., Wang, K., Chen, Y., Peng, X., & Lou, Y. (2024). Evaluating and Improving ChatGPT for Unit Test Generation. *Proc. ACM Softw. Eng.*, 1(FSE), 76:1703-76:1726. <https://doi.org/10.1145/3660783>
- Zhang, B., Liang, P., Zhou, X., Ahmad, A., & Waseem, M. (2023). *Practices and Challenges of Using GitHub Copilot: An Empirical Study*. 124–129. <https://doi.org/10.18293/SEKE2023-077>
- Zuboff, S. (2018). *Das Zeitalter des Überwachungskapitalismus*. Campus Verlag.