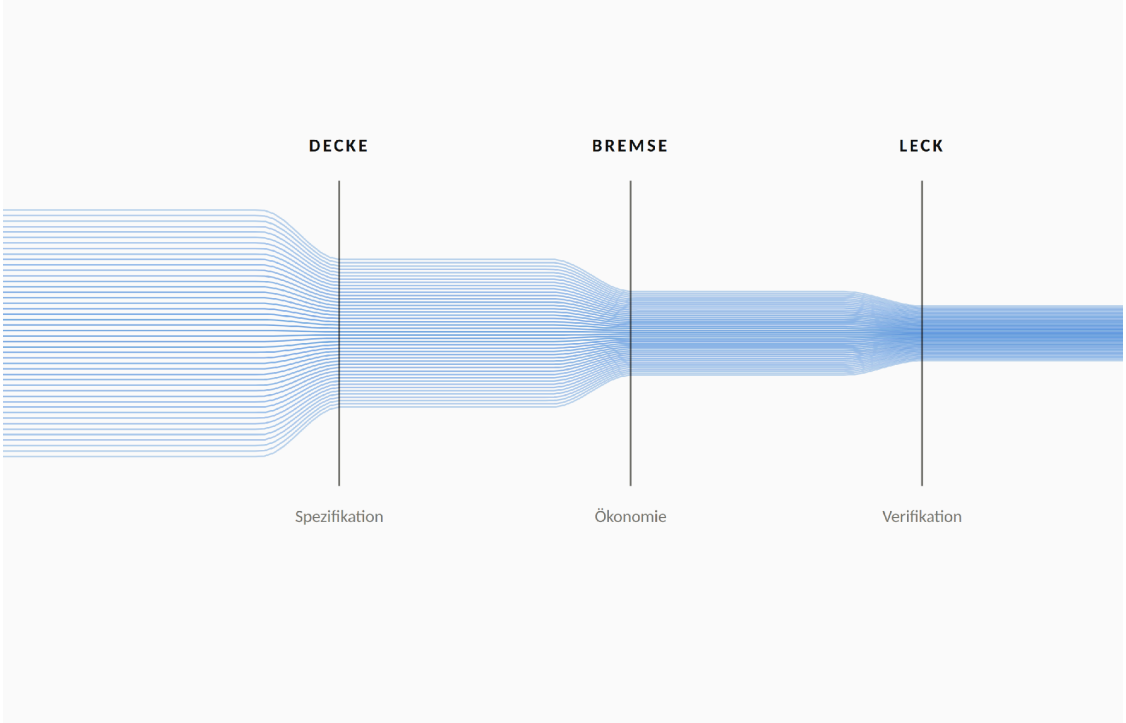
PHILIPP DEUTSCHER CONSULTING



CTO Report 2026

Engpässe, Archetypen, Szenarien

Was 200+ Coaching-Gespräche und die Studienlage über die CTO-Rolle im Jahr der Agenten sagen.

Philipp Deutscher · August 2026

Inhalt

Fünf Thesen 3
1 · Was zwischen 2025 und 2026 passiert ist 4
2 · Warum sich die Produktivitätsstudien widersprechen 6
3 · Der Engpass wandert 9
4 · Die Ökonomie: Token, Compute, Strom 13
5 · Die Organisation: Teams altern 16
6 · Die Regelseite 19
7 · Archetypen 2026 23
8 · Was Agenten 2026 nicht können 25
9 · Drei Szenarien für 2029 26
10 · Selbstdiagnose 27
Woher das Rollenbild kommt 28
Methodik und Quellen 29
Quellenverzeichnis 30

Fünf Thesen

1. Der Engpass wandert, und er ist nicht mehr die Engineering-Kapazität. Er landet an drei Stellen gleichzeitig. Die Spezifikation begrenzt, wie viel Gewinn überhaupt möglich ist. Die Kosten stoppen hart und plötzlich. Die Verifikation kostet leise und mit Verzögerung. Wer nur die Generierungsseite beschleunigt, verschiebt sein Problem, statt es zu lösen.

2. Sechzehn Prozent. In meinem eigenen Aufbau mit neun Agentenrollen flossen 16 Prozent der Tokens in tatsächliche Entwicklung. Ein Klient, der ein einzelnes Epic vollautonom durchlaufen ließ, kam bei 230 Millionen Tokens unabhängig auf 17 Prozent. Der Rest ging in Orchestrierung, Verifikation und Rework. Die Frage ist nicht mehr, ob KI produktiv ist. Die Frage ist, wofür Sie bezahlen.

3. Abgebrochen wird wegen Kosten, bezahlt wird für Qualität. In keinem einzigen der ausgewerteten Fälle wurde ein KI-Programm wegen eines Produktionsausfalls durch fehlerhaften Code gestoppt. Gestoppt wurde wegen Rechnungen. Die Qualitätskosten verschwinden dadurch nicht, sie erscheinen nur zwei Quartale später, als Rework, als Duplikate, als Wartungsaufwand an Code, den niemand mehr versteht.

4. Teams schrumpfen nicht, sondern altern. In 207 Coaching-Gesprächen zwischen März und August 2026 findet sich genau ein Fall, in dem Nachwuchs eingestellt wurde, und der wird rückblickend skeptisch bewertet. Eingestellt wird senior. Das ist keine Personalentscheidung für dieses Quartal, sondern eine Kapazitätsentscheidung für 2031.

5. Die Regulierung wurde verschoben und ist schärfer geworden. Wer im Juli gelesen hat, dass die Hochrisiko-Pflichten des AI Act auf 2027 und 2028 rutschen, hat vermutlich das Falsche abgeleitet. Die Transparenzpflichten gelten seit dem 2. August 2026. Die Meldepflicht mit 24-Stunden-Frist greift ab September. Und ab dem 9. Dezember 2026 ist Software ein Produkt im Sinne der Produkthaftung.

1 · Was zwischen 2025 und 2026 passiert ist

Im letzten Report (von Mitte 2025) stand eine Prognose für 2030. Sie lautete, der CTO werde weniger Builder und mehr Kurator sein, weil KI-gestützte Werkzeuge große Teile von Entwicklung, Test und Überwachung übernehmen. Die Prognose ist eingetreten, nur vier Jahre früher als angenommen.

Das ist keine Erfolgsmeldung. Prognosen, die zu früh eintreffen, sind meistens aus dem falschen Grund richtig gewesen. Der Report von 2025 hat den Effekt erwartet und den Mechanismus verfehlt. Erwartet wurde Substitution: Maschinen übernehmen Arbeit, Menschen rücken eine Ebene höher. Eingetreten ist etwas anderes. Die Arbeit ist nicht verschwunden, sie hat die Form gewechselt. Aus Schreiben wurde Prüfen. Aus Kapazitätsplanung wurde Budgetplanung. Aus technischer Führung wurde in vielen Organisationen ein Verteilungskampf darum, wer die KI-Strategie überhaupt besitzt.

Was sich in zwölf Monaten messbar geändert hat:

Neunzig Prozent der Softwareprofis nutzen KI bei der Arbeit, im Median zwei Stunden am Tag.[1] Die Nutzung von Agenten am Arbeitsplatz stieg binnen eines Jahres von 31 auf 59 Prozent, die tägliche Nutzung von 14 auf 37 Prozent.[2] Über 500 Organisationen hinweg gemessen liegt der Anteil KI-generierten Codes bei 52 Prozent, ein Quartal zuvor waren es 34.[3] Bei Google sind es nach Angabe des CEO 75 Prozent des neuen Codes,[4] bei Anthropic über 80 Prozent des gemergten Codes.[5]

Diese Zahlen messen nicht dasselbe, und der Unterschied ist wichtig. Entwickler schätzen ihren eigenen KI-Anteil auf 42 Prozent.[6] Telemetrie misst 52. Vorstandsvorsitzende sagen 75 bis 80. Die höchsten Werte beziehen sich auf neu geschriebene Zeilen mit menschlicher Freigabe, nicht auf autonome Autorschaft. Wer diese Zahlen im eigenen Board zitiert, sollte den Unterschied kennen, sonst verspricht er eine Autonomie, die es nicht gibt.

Parallel dazu ist eine Infrastrukturschicht entstanden, die kaum jemand mitbekommen hat. Am 9. Dezember 2025 wurde das Model Context Protocol an die neu gegründete Agentic AI Foundation der Linux Foundation übergeben, zusammen mit der AGENTS.md-Konvention. Platinum-Mitglieder sind AWS, Anthropic, Block, Bloomberg, Cloudflare, Google, Microsoft und OpenAI. Über 10.000 MCP-Server sind veröffentlicht, AGENTS.md wird in mehr als 60.000 Open-Source-Projekten verwendet.[7] Seit acht Monaten gibt es damit eine neutrale Governance-Schicht für Agenten-Interoperabilität. Wer heute Integrationsstrategie plant, plant gegen einen Standard und nicht gegen einen Anbieter. Das ist die stillste und praktisch folgenreichste Nachricht des Jahres.

Und dann gibt es die Bewegung, die im letzten Report noch als eigenständiger Technologieblock stand: Web3. Sie ist weg. Nicht widerlegt, nicht gescheitert, einfach von keiner Prioritätenliste mehr getragen. Ich lasse den Punkt trotzdem stehen, weil er etwas über die Halbwertszeit von

---

[1] DORA / Google Cloud, *State of AI-assisted Software Development 2025*, 23.9.2025, rund 5.000 Befragte. dora.dev/dora-report-2025

[2] Stack Overflow, *Agents on a Leash — Pulse Survey*, 27.5.2026, rund 1.100 Befragte. stackoverflow.blog/2026/05/27

[3] DX, *State of AI Impact in Engineering, Q2 2026*, 22.7.2026, Telemetrie über 500+ Organisationen. Proprietärer Datensatz, Methodik nicht offengelegt.

[4] Sundar Pichai, Google Cloud Next, 23.4.2026. Bezieht sich auf neu geschriebene Zeilen mit menschlicher Freigabe.

[5] Anthropic, *When AI builds itself*, Mai 2026. anthropic.com/institute/recursive-self-improvement

[6] Sonar, *State of Code Developer Survey 2026*, 1.149 Profis, Feldphase Oktober 2025. Selbstauskunft, keine Messung.

[7] Linux Foundation, *Formation of the Agentic AI Foundation*, 9.12.2025. linuxfoundation.org/press

Pflichtthemen sagt. Ein Technologiethema, das 2022 in jeder CTO-Agenda stand, ist vier Jahre später nicht mehr diskussionswürdig. Dieselbe Prüfung sollte jeder an das anlegen, was er gerade für unverzichtbar hält.

2 · Warum sich die Produktivitätsstudien widersprechen

Es gibt inzwischen belastbare Studien zur Frage, ob KI die Softwareentwicklung beschleunigt. Sie kommen zu entgegengesetzten Ergebnissen, und das ist der nützlichste Befund des Jahres.

Im Juli 2025 hat METR sechzehn erfahrene Open-Source-Entwickler an 246 echten Issues in ihren eigenen Repositories gemessen, Projekte mit im Schnitt über 22.000 Stars und mehr als einer Million Zeilen Code. Mit KI-Werkzeugen brauchten sie neunzehn Prozent **länger**. Interessanter als das Ergebnis ist die Wahrnehmung: Vorher erwarteten die Teilnehmer eine Beschleunigung um 24 Prozent, hinterher glaubten sie, zwanzig Prozent schneller gewesen zu sein. Sie hatten sich um fast vierzig Prozentpunkte geirrt, in einer Disziplin, in der sie Experten sind.[8]

Ein Jahr später hat Microsoft die eigene Entwicklungsorganisation über vier Monate ausgewertet, Zehntausende berechtigte Engineers in der Adoptionsanalyse und eine engere Kohorte für die Wirkungsschätzung. Herausgekommen sind 24 Prozent mehr gemergte Pull Requests pro Engineer und Tag, mit einem Konfidenzintervall von 14,5 bis 33,7 Prozent und einer sauberen Dosis-Wirkungs-Beziehung. Wer drei Tage pro Woche mit KI arbeitete, lag bei plus fünfzehn Prozent, wer fünf Tage oder mehr arbeitete, bei plus fünfzig. Die Autoren schreiben selbst dazu, dass Codequalität nicht gemessen wurde.[9]

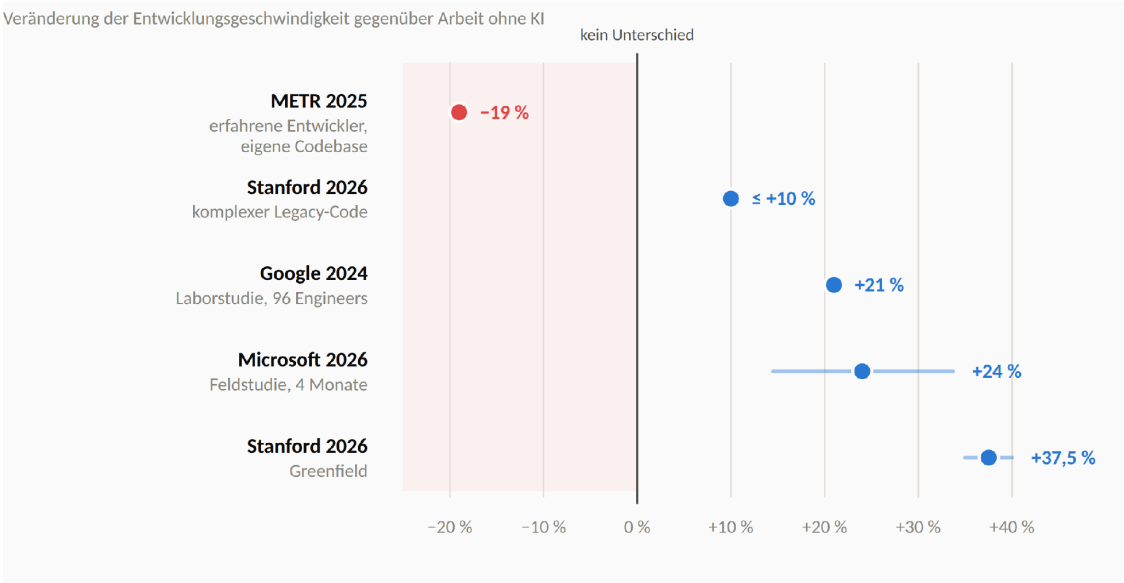
Beide Studien sind gut gemacht. Beide Ergebnisse stimmen. Sie messen verschiedene Dinge:

METR misst erfahrene Entwickler in Codebases, die sie seit Jahren kennen. Dort ist der Kontext bereits im Kopf, und das Werkzeug muss ihn erst rekonstruieren. Microsoft misst gemergte Pull Requests, also Output-Volumen, nicht gelöste Probleme und nicht Haltbarkeit. DORA findet für 2025 erstmals eine positive Korrelation zwischen KI-Adoption und Delivery-Durchsatz, bei weiterhin negativer Korrelation mit Delivery-Stabilität.[10] Und der DORA-ROI-Report zitiert Stanford-Forschung von Denisov-Blanch und Kollegen, die auf Telemetriedaten von rund 100.000 Entwicklern beruht: Der Produktivitätsgewinn liegt im Greenfield bei 35 bis 40 Prozent, im komplexen Legacy-Code bei zehn Prozent oder weniger.

[8] METR, *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*, 10.7.2025. 16 Entwickler, 246 Issues. metr.org/blog/2025-07-10

[9] Murphy-Hill, Butler, Savelieva (Microsoft), arXiv:2607.01418, 1.7.2026. Zehntausende berechtigte Engineers in der Adoptionsanalyse, engere Kohorte für die Wirkungsschätzung.

[10] DORA / Google Cloud, *State of AI-assisted Software Development 2025*, 23.9.2025, rund 5.000 Befragte. dora.dev/dora-report-2025



Dieselbe Frage, gegensätzliche Antworten. Der Gewinn ist eine Funktion des Kontextaufwands: je mehr Vorwissen eine Aufgabe verlangt, desto kleiner der Vorteil.

METR 2025 · Paruch et al. (Google) 2024 · Murphy-Hill et al. (Microsoft) 2026 · Denisov-Blanch et al. (Stanford), zitiert im DORA ROI-Report 2026

[11]

Daraus folgt eine Regel, die sich im Coaching zuverlässig bewährt: **Der Gewinn ist eine Funktion des Kontextaufwands.** Je mehr Kontext eine Aufgabe braucht, den nur ein Mensch hat, desto kleiner der Vorteil. Neue Oberfläche in einem klar geschnittenen Frontend, hoher Gewinn. Änderung an einem gewachsenen Monolithen mit hunderttausenden Dateien und ungeschriebenen Regeln, geringer bis negativer Gewinn.

Was in der eigenen Organisation wirklich passiert, lässt sich mit dieser Erkenntnis allein nicht sagen. Und hier liegt das eigentliche Problem, das in meinen Coaching-Sessions in etwa jedem fünften Gespräch mit KI-Bezug auftaucht: Fast niemand kann den Gewinn belegen. Nicht weil es keinen gibt, sondern weil die Vergleichszahl fehlt. Story Cycle Time bricht als Kennzahl, weil sie Story-Größe nicht abbildet und KI vor allem die Zuschnitte verändert. Velocity ist zwischen Teams nicht kalibriert. Adoption wird auf „etwa 60 Prozent" geschätzt statt gemessen.

Ein Vorstand fragt nach dem ROI. Der CTO hat keine Baseline von vor der Einführung, weil vorher niemand nach dem ROI von Entwicklern gefragt hat. Und falls doch, konnte man das anmoderieren. Also entsteht eine Zahl, die eher eine Verhandlungsposition ist als eine Messung. In einem Fall aus meiner Praxis hat ein CFO die richtige Frage gestellt und damit sofort die Lücke offengelegt: Was kostet ein Entwicklertag heute, und was hat er vor der Einführung gekostet? Die Antwort lag im September vor dem Verwaltungsrat. Die Messung dafür musste rückwirkend gebaut werden.

[11] Denisov-Blanch et al. (Stanford), Telemetrie von rund 100.000 Entwicklern, zitiert in: DORA, *ROI of AI-assisted Software Development*, 11.5.2026.

Wer 2026 noch keine Baseline hat, sollte aufhören, sie nachträglich konstruieren zu wollen, und stattdessen ab sofort messen. Eine ehrliche Messung, die in sechs Monaten belastbar ist, schlägt eine rekonstruierte Zahl, die im Board zerlegt wird.

Die Kennzahl, die fast alle verfehlen

Es gibt einen Fehler, der so häufig ist, dass er ein eigenes Kapitel verdient. Gemessen wird die Adoption, nicht der Effekt.

Ein PE-finanzierter Softwareanbieter mit knapp hundert Engineers hatte 82 Prozent KI-generierten Code. Später wurde eine Acceptance Rate von 97 Prozent gemessen. Beide Zahlen wurden im Board als Erfolg berichtet. Der Durchsatz blieb flach.

Der Grund lag nicht im Engineering. Die Anforderungsseite kam nicht mit. Das Produktmanagement war für die neue Geschwindigkeit weder besetzt noch geschnitten, und die Entwickler verbrachten rund die Hälfte ihrer Zeit in Klärungsmeetings. Der Code entstand schneller, aber niemand konnte schneller sagen, welcher Code entstehen sollte.

Eine Acceptance Rate misst, ob ein Entwickler einen Vorschlag angenommen hat. Sie misst nicht, ob der Vorschlag gebraucht wurde. Das ist der Unterschied zwischen einer Aktivitätskennzahl und einer Ergebniskennzahl, und er ist der teuerste Messfehler des Jahres.

Interessant ist die eine Kennzahl, die sich in diesem Fall doch klar verbessert hat: die Zeit bis zur Behebung eines Bugs. Im 85. Perzentil von zehn auf sechs Tage, im Median von zwei auf einen Tag. Reparieren wurde deutlich schneller, Neubauen nicht. Das passt zur Kontextregel. Ein Bug ist ein eng umrissenes Problem mit klarem Zielzustand. Eine neue Funktion ist es nicht.

3 · Der Engpass wandert

Das ist das Kapitel, um das dieser Report gebaut ist. Und der auch in der CTO Community regelmäßig diskutiert wurde.

Die Erzählung des Jahres lautet, KI mache Softwareentwicklung schneller. Das stimmt, und es ist trotzdem irreführend, weil es einen Engpass beschreibt, der 2026 nicht mehr der entscheidende ist. Als Codeschreiben teuer war, war Engineering-Kapazität die knappe Ressource. Codeschreiben ist billig geworden. Die Knappheit ist nicht verschwunden, sie ist umgezogen.

Sie ist an drei Stellen aufgetaucht, und die drei verhalten sich völlig unterschiedlich.

Die Decke: Spezifikation

Die Spezifikation begrenzt, wie viel Gewinn überhaupt möglich ist. Sie wirkt sofort und ist unsichtbar, weil alle Engineering-Kennzahlen gut aussehen, während sie greift.

Der Fall mit 82 Prozent KI-Code und flachem Durchsatz ist der klarste, aber nicht der einzige. Ein Klient stellte nach einem Workshop fest, dass nicht langsames Engineering seine Effizienz begrenzt, sondern die Struktur seiner Produktorganisation. Ein anderer beschreibt das Kernproblem seines Teams so, dass Anforderungen als technische Lösung formuliert werden statt als Nutzerproblem, was zu Nacharbeit führt. Ein dritter dokumentiert 17.000 Markdown-Dateien und 15.000 Jira-Einträge als Wissensbasis, die für Menschen fragmentiert und für Maschinen unbrauchbar ist.

Das Muster ist überall dasselbe. Solange Menschen den Code schrieben, war unscharfe Anforderung ein Reibungsverlust. Entwickler haben nachgefragt, interpretiert, aus Erfahrung ergänzt. Diese stille Übersetzungsleistung war nie in einem Prozess beschrieben, und sie fällt weg, wenn eine Maschine übernimmt. Ein Agent fragt nicht nach. Er baut, was dasteht.

Damit ist die Anforderungsqualität von einem Komfortthema zu einem harten Kapazitätslimit geworden. Wer seine Delivery verdreifachen will, aber seine Spezifikationsqualität nicht anfasst, verdreifacht die Produktion von Dingen, die niemand bestellt hat.

Die praktische Konsequenz ist unbequem, weil sie das Budget in eine Richtung lenkt, in der es kein CTO gerne verteidigt. Die wirksamste KI-Investition in einer Organisation mit schwacher Produktseite ist keine Entwicklerlizenz. Es ist ein besserer Product Manager.

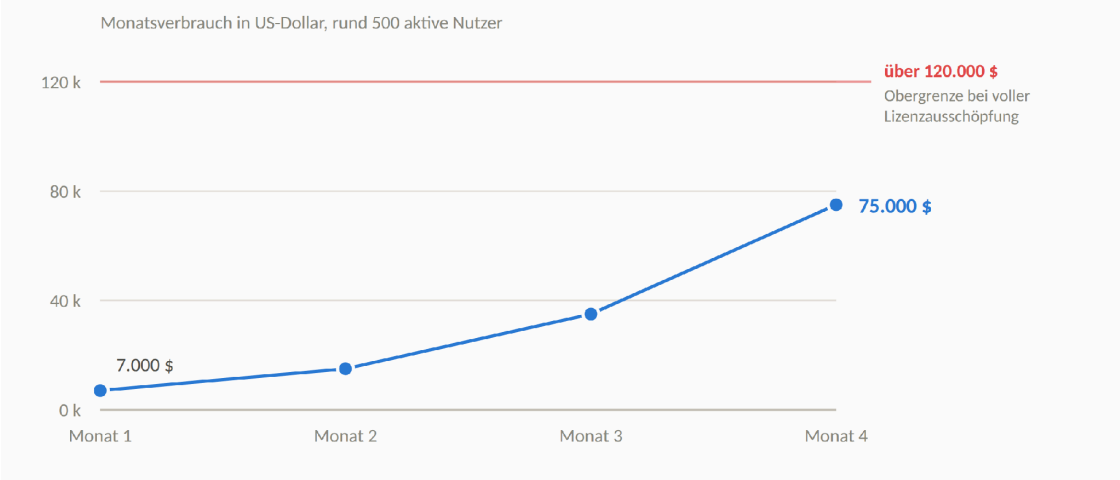
Die Bremse: Ökonomie

Die Kosten stoppen hart und plötzlich. Sie sind in meinen Daten der mit Abstand häufigste Grund, aus dem ein KI-Programm zurückgedreht wurde.

Ein Deep-Tech-Startup mit sieben Entwicklern zahlte zwei Jahre lang 100 bis 200 Euro im Monat für die gesamten KI-Werkzeuge des Teams. Nach einem Modellsprung erledigte ein Entwickler in zwei Tagen ein Oberflächenprojekt, das für zwei Personen auf sechs bis acht Wochen geschätzt war. Dann zogen die Anbieter die Kontingente an. Der Test bestand darin, einem Entwickler hundert Euro Token-Budget zu geben. Nach dreißig Minuten war es aufgebraucht, die Aufgabe unfertig. Die

Hochrechnung bei gleichem Arbeitstempo lag bei über 10.000 Euro im Monat. Das Team ist auf Refactorings, KI-gestütztes Code-Review und Wegwerf-Prototypen zurückgegangen.

Eine IT-Dienstleistungsgruppe mit rund 10.000 Mitarbeitenden gab ein LLM-Produkt zentral frei. Der Monatsverbrauch entwickelte sich in vier Monaten von 7.000 auf 15.000, dann 35.000 und schließlich 75.000 US-Dollar, bei etwa 500 aktiven Nutzern. Bei voller Ausschöpfung der bereits gekauften Lizenzen läge die Obergrenze über 120.000 im Monat. Einzelne Nutzer haben sich selbst 2.000-Dollar-Limits setzen lassen. Der Vorstand hat eine Taskforce eingesetzt, die Token-Verbrauch in Umsatz übersetzen soll.



Eine IT-Dienstleistungsgruppe mit rund 10.000 Mitarbeitenden nach zentraler Freigabe eines Coding-Assistenten. Vier Monate, Faktor zehn, kein Budgetdeckel.

Klientendaten, anonymisiert, Sommer 2026

Das ist die neue Klasse von Betriebsrisiko. Cloud-Kosten waren unangenehm, aber sie hatten eine Beziehung zur Last. Token-Kosten haben eine Beziehung zur Denktiefe, und die ist nicht deterministisch. Dasselbe Ticket kostet je nach Ansatz des Modells das Mehrfache.

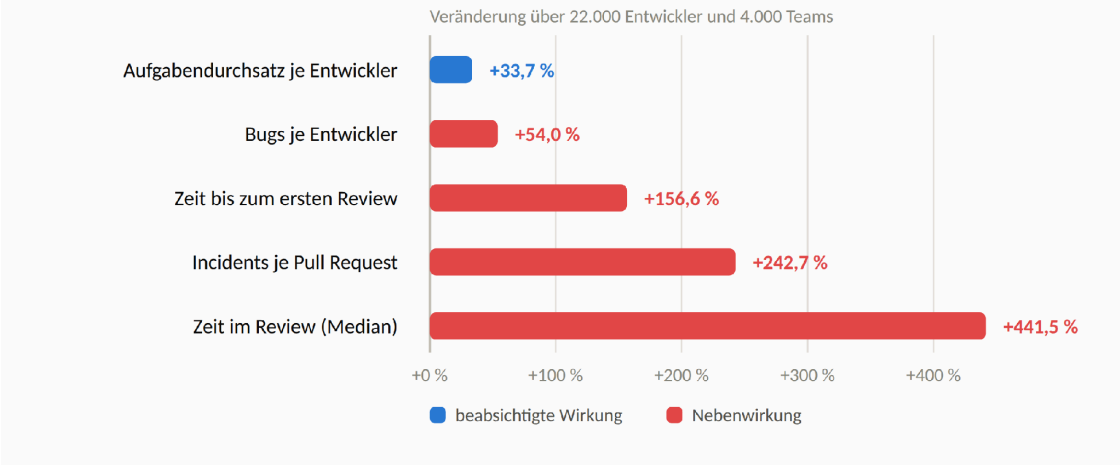
Und die Preisrichtung ist nicht die, die man erwartet. Der Preis pro Token fällt. Der Preis pro erledigter Aufgabe steigt, weil Reasoning-Modelle mehr Tokens und deutlich mehr Agentenschritte verbrauchen. Wer seine Planung auf fallende Modellpreise stützt, plant an der falschen Größe.

Das Leck: Verifikation

Die Verifikation ist der Engpass, der niemanden zum Handeln zwingt und trotzdem am teuersten ist. Sie stoppt kein Programm. Sie schickt die Rechnung mit zwei Quartalen Verzögerung, und dann kommt sie unter einem anderen Namen an, als Rework, als Wartungsaufwand, als Duplikate, als Code, den keiner mehr anfassen will.

Die öffentliche Datenlage ist an dieser Stelle eindeutiger als jede meiner Sessions. Über 22.000 Entwickler und mehr als 4.000 Teams hinweg gemessen stieg der Aufgabendurchsatz pro Entwickler um 33,7 Prozent. Im selben Zeitraum stiegen die Bugs pro Entwickler um 54 Prozent und das Verhältnis von Incidents zu Pull Requests um 242,7 Prozent. Die Zeit im Review stieg im Median um

441,5 Prozent, die Zeit bis zum ersten Review um 156,6 Prozent, und 31,3 Prozent mehr Pull Requests wurden ganz ohne Review gemergt.



Der Durchsatz steigt einmal. Die Folgekosten steigen viermal, und die teuerste Reaktion darauf ist die stillste: 31,3 % mehr Pull Requests werden ganz ohne Review gemergt.

Faros AI, The Acceleration Whiplash, April 2026 (proprietärer Datensatz)

[12]

Diese letzte Zahl ist die eigentliche Nachricht. Wenn der Review zum Engpass wird, wird er nicht ausgebaut. Er wird umgangen.

Dazu kommen zwei Befunde, die sich über Jahre nicht bewegt haben. In 44 Prozent der Fälle, in denen ein Modell Code für eine sicherheitsrelevante Aufgabe erzeugt, entsteht dabei eine ausnutzbare Schwachstelle. Dieser Wert liegt seit zwei Jahren praktisch unverändert bei 44 bis 45 Prozent, während Syntaxfehler im selben Zeitraum faktisch verschwunden sind.[13] Die Modelle sind besser darin geworden, Code zu schreiben, der funktioniert, und nicht besser darin, Code zu schreiben, der sicher ist. Und die Duplikation im Code hat einen Höchststand erreicht, plus 81 Prozent seit 2023, während das Refactoring-Signal, also verschobener statt kopierter Code, von 21 Prozent im Jahr 2022 auf 3,8 Prozent gefallen ist.[14]

Ich habe das an meinem eigenen Aufbau gesehen und es war unangenehmer, als die Zahlen vermuten lassen. Neun Agentenrollen, vollständige Organisation, kein Mensch in der Schleife. Alle Tests grün, ordentliche Coverage. Zwischen 40 und 50 Prozent des erzeugten Codes war tot oder doppelt.

Die Agenten haben nicht versagt. Sie haben exakt das optimiert, was gemessen wurde, nämlich bestandene Tests. Ob der Code gebraucht wird, hat niemand gemessen, also hat es niemanden interessiert.

---

[12] Faros AI, *The Acceleration Whiplash*, 12.4.2026. 22.000 Entwickler, 4.000+ Teams. Proprietärer Datensatz eines Anbieters.

[13] Veracode, *2026 GenAI Code Security Report*, 24.3.2026. Gemessen wird der Anteil der Code-Generierungsaufgaben, bei denen eine ausnutzbare Schwachstelle entsteht, nicht der Anteil am Code.

[14] GitClear, *The Maintainability Gap*, Januar 2026, 623 Mio. analysierte Changes. Anbieterstudie.

Dass die Verifikation in meinen Coaching-Gesprächen seltener als Auslöser auftaucht als Kosten, ist kein Gegenargument. Es ist die Beschreibung des Problems. Ein Engpass, der niemanden zwingt, wird nicht bearbeitet. Er wird geerbt.

Warum die drei zusammengehören

Die Decke begrenzt den Gewinn. Die Bremse stoppt das Programm. Das Leck macht die Rechnung teurer, als sie aussieht. Sie greifen ineinander, und zwar in einer Reihenfolge, die die meisten Organisationen genau falsch herum abarbeiten.

Der übliche Weg beginnt bei der Generierung. Lizenzen ausrollen, Adoption messen, Erfolg berichten. Dann steigen die Kosten, und es wird an Modellen und Kontingenten optimiert. Dann fällt auf, dass der Durchsatz nicht steigt, und es wird an der Produktseite gearbeitet. Die Verifikation kommt zuletzt, meistens nach dem ersten größeren Vorfall.

Die reifsten Organisationen in meinen Daten haben genau umgekehrt angefangen. Sie haben KI zuerst im Code-Review eingesetzt und die Generierung erst freigegeben, als die Prüfschicht stand. Ein Klient berichtet, dass ein lokal laufender Review-Agent erhebliche Zeit spart und Fehler abfängt, bevor sie in Produktion gehen. Anthropic beschreibt für die eigene Codebase dasselbe Vorgehen und schätzt rückblickend, dass ein automatisierter Reviewer rund ein Drittel der Fehler hinter vergangenen Vorfällen gefunden hätte.[15]

Das ist die konkreteste Empfehlung dieses Reports. Wer noch am Anfang steht, sollte nicht mit der Generierung beginnen. Prüfen zuerst, erzeugen danach.

Wie es aussieht, wenn es funktioniert

Ein reguliertes Softwarehaus mit rund sechzig Entwicklern hat den Engpass tatsächlich gelöst. Ein Feature, das eine Person drei Sprintwochen gekostet hätte, läuft dort in etwa zwei Stunden durch die Pipeline. Der Aufbau ist kein Werkzeugkauf, sondern ein Korsett:

Die Architektur ist vorab festgelegt, hexagonal, mit fester Verzeichnisstruktur. Die Modelltemperatur steht auf 0,2. Jede Anforderung trägt eine Use-Case-Kennung, die bis in den Quellcode durchgezogen wird. Es gibt vier Gates, an denen die Arbeit angehalten wird. Das teure Modell wird ausschließlich für Anforderungsanalyse und Architektur eingesetzt, weil alle Alternativen dort unbrauchbare Datenmodelle erzeugt haben, das günstigere schreibt den Code.

Der entscheidende Baustein kam zuletzt und ist der, den fast niemand baut: ein Agent, der nicht den Code prüft, sondern das Prüfergebnis bewertet, und nur bei Unsicherheit an einen Menschen eskaliert. Ohne diese Schicht war der CTO selbst der Engpass, weil jedes Review-Ergebnis über seinen Tisch lief.

Bemerkenswert an diesem Aufbau ist die Reihenfolge der Entscheidungen. Zuerst wurde das Budget gesetzt, rund 8.300 Euro pro Entwickler und Jahr. Dann wurde die Architektur so gebaut, dass sie in dieses Budget passt. Nicht umgekehrt.

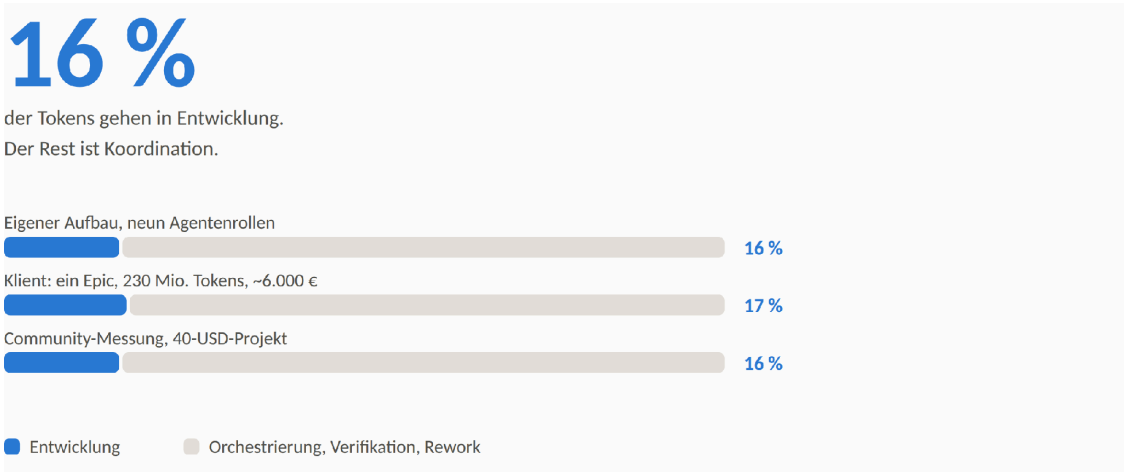
[15] Anthropic, *When AI builds itself*, Mai 2026. anthropic.com/institute/recursive-self-improvement

4 · Die Ökonomie: Token, Compute, Strom

Wofür Sie eigentlich bezahlen

Ich habe meinen Agenten-Aufbau nachgerechnet, weil mir die Rechnung zu hoch vorkam. Neun Rollen über einen offenen Orchestrierungslayer, vollständige Entwicklungsorganisation. Sechzehn Prozent der Tokens gingen in Entwicklung. Alles andere floss in Orchestrierung, Verifikation und Rework.

Ich hätte das für einen Fehler in meinem Aufbau gehalten, wenn nicht ein Klient unabhängig auf dieselbe Größenordnung gekommen wäre. Er hat ein einzelnes Epic vollautonom durchlaufen lassen, 230 Millionen Tokens, rund 6.000 Euro. Bei der Nachanalyse entfielen 17 Prozent auf Entwicklung, über achtzig Prozent auf Verifikation, Orchestrierung und Nacharbeit. Eine dritte Messung aus einer Community an einem 40-Dollar-Projekt kam auf sechzehn.



Drei unabhängige Messungen in drei Größenordnungen, dasselbe Verhältnis. Wer glaubt, Entwicklung zu kaufen, kauft zu fünf Sechsteln Koordination.

Eigene Messungen und Klientendaten, März bis August 2026

Drei Messungen, unterschiedliche Größenordnungen, dasselbe Verhältnis. Das ist die Zahl, die ich in keinem Anbieter-Whitepaper gefunden habe, und sie erklärt beides: warum die Rechnungen explodieren und warum der Durchsatz enttäuscht. Wer glaubt, Entwicklung zu kaufen, kauft zu fünf Sechsteln Koordination.

Daraus folgt keine Empfehlung, agentische Aufbauten zu meiden. Es folgt daraus, dass der Overhead ein eigenes Optimierungsziel ist. Die Frage lautet nicht mehr, welches Modell den besten Code schreibt. Sie lautet, wie viele Runden das System braucht, bis es fertig ist.

Was Sie budgetieren sollten

Es gibt für 2026 keinen etablierten Richtwert dafür, was KI-Werkzeuge pro Entwickler kosten dürfen. Was ich aus der Praxis anbieten kann, sind vier Datenpunkte:

Ein reguliertes Softwarehaus setzt 8.300 Euro pro Entwickler und Jahr als bewusste Obergrenze und baut die Architektur darauf aus. Ein PE-finanzierter Anbieter liegt aktuell bei 15.000 bis 20.000 Euro im Monat und prognostiziert 600.000 bis 700.000 für das Folgejahr. Ein Frühphasen-Startup mit vollständig agentischem Prozess kam auf 500 bis 1.000 Euro pro Entwickler und Tag und hat den Prozess deshalb zurückgedreht. Und ein Konzern mit rund 500 aktiven Nutzern liegt bei 75.000 US-Dollar im Monat mit steigender Tendenz.

Die Spanne zwischen dem geplanten Fall und den ungeplanten Fällen ist der ganze Punkt. Wer ein Budget setzt und die Architektur darauf auslegt, landet bei niedrigen fünfstelligen Beträgen im Jahr pro Kopf. Wer die Architektur laufen lässt und die Rechnung hinterher liest, landet bei Beträgen, die eine Vorstandsvorlage erzwingen.

Über den Markt hinweg passt das ins Bild. Die Ausgaben für generative KI im Unternehmenseinsatz stiegen von 11,5 Milliarden US-Dollar im Jahr 2024 auf 37 Milliarden im Jahr 2025, ein Faktor von 3,2, und das bei fallenden Tokenpreisen. Wer auf Commoditisierung wartet, sollte auch zur Kenntnis nehmen, dass der Anteil offener Modelle an der Unternehmensnutzung im selben Zeitraum von 19 auf 11 Prozent gefallen ist.[16]

Der Subventionszyklus

Mehrere Klienten haben unabhängig voneinander dieselbe Regel formuliert, und sie ist gut genug, um sie hier stehen zu lassen: Neue Werkzeuge früh und hart testen, solange Investorengeld das Experiment bezahlt. Aber nie darauf planen.

Das Muster wiederholt sich zuverlässig. Anbieter starten mit Pauschaltarifen, um Marktanteil zu kaufen. Dann werden Kontingente verschärft. Dann wird auf Verbrauchsabrechnung umgestellt. Wer seine Roadmap auf das günstige Fenster gebaut hat, muss zurückrudern.

Die schärfste Formulierung des Risikos kam nicht vom Preis, sondern von der Verfügbarkeit. Ein Teilnehmer wies darauf hin, dass ein altes Modell gar nicht teurer werden muss. Es kann einfach abgekündigt werden. Wer seine Pipeline auf ein bestimmtes Modellverhalten kalibriert hat, hat dann kein Kostenproblem, sondern ein Migrationsprojekt.

Deshalb kaufen mehrere CTOs Unabhängigkeit bewusst teurer ein, mit offenen Orchestrierungsschichten statt proprietärer Werkzeuge. Die Begründung eines Klienten trifft es: ein bisschen mehr zahlen, dafür die Freiheit behalten.

Der wirksamste Kostenhebel ist dabei nicht der Anbieterwechsel, sondern die Modellzuordnung. Das stärkste Modell nur dort, wo es den Unterschied macht, also bei Anforderungsanalyse und Architektur. Ein mittleres Modell fürs Codeschreiben. Das billigste für Klassifikation und Routine. Wer alles durch das stärkste Modell schickt, zahlt ein Vielfaches für wenige Prozentpunkte.

Die Schicht darunter: Strom und Souveränität

Es gibt ein physisches Limit unter der Kostenfrage, das in der CTO-Diskussion kaum vorkommt und in zwei Jahren jede Kapazitätsplanung bestimmen wird.

[16] Menlo Ventures, *State of Generative AI in the Enterprise*, Dezember 2025.

Die Anschlusswarteschlangen für neue Rechenzentren liegen in den westeuropäischen Kernmärkten bei sieben bis zehn Jahren. Irland hat sein pauschales Anschlussmoratorium für den Großraum Dublin im Dezember 2025 beendet und durch ein Auflagenregime ersetzt, das Neuanschlüsse ab zehn Megavoltampere praktisch nur noch mit vollständiger Eigenerzeugung oder Speicherung erlaubt. Das ist keine Entspannung, sondern eine andere Form derselben Grenze. Wer Rechenkapazität in Europa plant, plant nicht gegen Serverpreise, sondern gegen Netzanschlüsse.[17]

Parallel dazu hat sich die Marktstruktur verschoben. Europäische Anbieter halten noch rund fünfzehn Prozent des europäischen Cloud-Marktes, 2017 waren es 29. Die Gegenbewegung ist im Gange, aber sie kommt von den Hyperscalern selbst: AWS hat seine European Sovereign Cloud im Januar 2026 in Brandenburg allgemein verfügbar gemacht, mit einem angekündigten Investitionsvolumen von über 7,8 Milliarden Euro. Die Kommission hat im Juni 2026 den Cloud and AI Development Act vorgeschlagen, der vier Souveränitätsstufen für den öffentlichen Sektor definiert.[18]

Damit ist Nachhaltigkeit für den CTO keine Reputationsfrage mehr. Im letzten Report stand sie noch als ethische Zukunftsperspektive. Sie ist eine Beschaffungsfrage geworden. Wer keine Megawatt bekommt, betreibt nichts, unabhängig davon, wie gut die Klimabilanz aussieht.

[17] Global Data Center Hub, Analyse der Netzanschlusswarteschlangen westeuropäischer Kernmärkte, 2026. Aggregatwert, keine standortscharfe Angabe. Zur Lage in Irland: Commission for Regulation of Utilities, *Large Energy User Connection Policy*, 12.12.2025.

[18] Europäische Kommission, Vorschlag für einen *Cloud and AI Development Act*, 3.6.2026. Marktanteilszahlen: Synergy Research, europäischer Cloud-Markt.

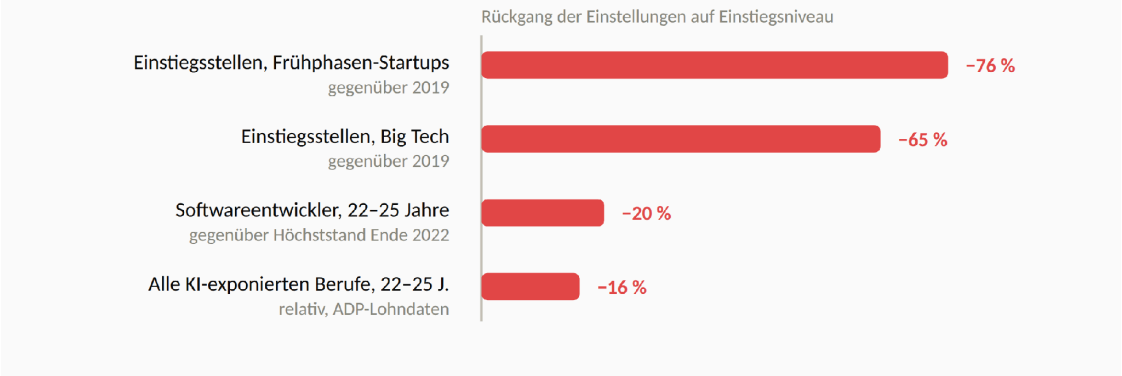
5 · Die Organisation: Teams altern

Ein Nachwuchsfall in 207 Gesprächen

Ich habe meine Coaching-Gespräche von März bis August 2026 daraufhin durchgesehen, wo Nachwuchs eingestellt wurde. Über 24 wiederkehrende Klienten und rund 207 Gespräche findet sich genau ein Fall, ein Quereinsteiger-Programm, von dem weniger als die Hälfte der Eingestellten geblieben ist. Der verantwortliche CTO bewertet es rückblickend zurückhaltend und zieht daraus den Schluss, ein Studium sei nicht nötig, Grundlagen aber schon. Woran es lag, ist ihm klar: Es fehlte nicht an Talent, sondern an Begleitung.

Alles andere ist Senior-Recruiting. Externe CTOs, Produktverantwortliche mit Plattform-Erfahrung, Technical Product Owner, Principal-Profile, Head-of-AI-Rollen.

Das deckt sich mit dem, was breiter gemessen wird. Die Beschäftigung der 22- bis 25-Jährigen in den KI-exponiertesten Berufen ist relativ um sechzehn Prozent zurückgegangen, bei Softwareentwicklern dieser Altersgruppe um knapp zwanzig Prozent gegenüber dem Höchststand Ende 2022. Die Anpassung läuft über Einstellungen, nicht über Gehälter. Erfahrene Beschäftigte in denselben Berufen sind stabil oder wachsen.[19] Einstellungen auf Einstiegspositionen liegen rund 65 Prozent unter dem Niveau von 2019, in Frühphasen-Startups 76 Prozent.



Erfahrene Beschäftigte in denselben Berufen sind stabil oder wachsen. Die Alterspyramide kippt, die Teamgrößen tun es nicht.

SignalFire State of Talent 2026 · Brynjolfsson et al., Stanford Digital Economy Lab, 2025/26

[20]

Wichtig ist dabei, was nicht passiert. Teams schrumpfen nicht pauschal. Das US-Arbeitsministerium prognostiziert für Softwareentwickler weiterhin sechzehn Prozent Beschäftigungswachstum bis 2034, in der Vorjahresausgabe waren es siebzehn Prozent bis 2033.[21] Der Trend zeigt nach unten und bleibt deutlich positiv. Die belastbare Aussage lautet nicht, dass weniger Menschen gebraucht werden. Sie lautet, dass die Alterspyramide gekippt ist.

---

[19] Brynjolfsson, Chandar, Chen, Stanford Digital Economy Lab, *Canaries in the Coal Mine?*, 13.11.2025, Update 9.2.2026. Basis: Payroll-Daten von ADP.

[20] SignalFire, *State of Talent Report 2026*, 22.6.2026, Basis 650 Mio. Personenprofile. Anbieterstudie.

[21] U.S. Bureau of Labor Statistics, *Occupational Outlook Handbook*, Vintage 2024–2034 (Softwareentwickler 16 %) gegenüber Vintage 2023–2033 (17 %).

Und daraus folgt ein Problem, das niemand in diesem Quartal spürt.

Nach Kapitel 3 ist die knappe Ressource nicht die Fähigkeit, Code zu schreiben. Es ist die Fähigkeit, zu beurteilen, ob Code stimmt, ob eine Architektur trägt, ob ein Vorschlag in eine Sackgasse führt. Anthropic beschreibt den menschlichen Vorteil in der eigenen Organisation als Urteilsvermögen darüber, welche Probleme zählen, welchen Ergebnissen zu trauen ist und wann ein Ansatz tot ist.[22]

Dieses Urteilsvermögen entsteht nicht im Studium. Es entsteht dadurch, dass jemand jahrelang eigene Fehler produziert und deren Folgen aushält. Wer 2026 keine Junioren einstellt, weil Agenten deren Arbeit erledigen, hat 2031 keine Senioren, die die Agenten beurteilen können.

Diese Entscheidung wird gerade in fast jeder Engineering-Organisation getroffen. Fast immer implizit, fast immer aus Kostengründen, und fast nie mit dem Zeithorizont, der ihr zusteht.

Ich habe dafür keine fertige Lösung. Was ich beobachte, ist, dass die wenigen Organisationen, die überhaupt darüber nachdenken, das Problem umformulieren: Nicht mehr „wie viele Junioren stellen wir ein", sondern „wie erzeugen wir Urteilsvermögen schneller als früher". Der Weg dorthin führt über Aufgaben mit echten Folgen und über Zugriff auf Entscheidungen, nicht über Schulungen.

Was beim Abbau tatsächlich entschieden wird

Der Softwareanbieter aus Kapitel 2, derselbe mit den 82 Prozent, soll seine Belegschaft nach dem Willen seines Finanzinvestors bis zum Jahreswechsel halbieren. Der CTO hielt das für unerreichbar, ohne die Roadmap zu kürzen. Eingereicht hat er einen zweistufigen Plan: zum 1. Januar 2027 eine Reduktion um knapp knapp 50 Prozent, getragen von einem neuen Betriebsmodell mit kleinen, direkt an die Geschäftsführung angebundenen Teams. Die zweite Stufe, ist an einen Nachweis gekoppelt, nämlich daran, dass bis Ende des zweiten Quartals 2027 alle nicht kritischen Arbeitspakete autonom von Agenten erledigt werden. Kritische Themen bleiben ausdrücklich beim Menschen.

Die Rechnung dahinter ist instruktiv, weil sie zeigt, warum diese Entscheidungen so schnell fallen: rund drei Millionen Euro eingesparte Personalkapazität gegen 600.000 bis 700.000 Euro KI-Kosten. Bei einem zwanzigfachen Bewertungsmultiplikator entspricht das rund fünfzig Millionen Euro Wertsteigerung. Transformationskosten laufen als Einmaleffekt außerhalb des bereinigten Ergebnisses. Der Kostendruck existiert also beim Personal und nicht beim Werkzeug.

Das regulierte Softwarehaus aus Kapitel 3, das seine Pipeline im Griff hat, baut parallel knapp ein Viertel seiner IT-Stellen ab. Ausdrücklich nicht aus finanzieller Not, sondern um Senior-Rollen gegenzufinanzieren, die es vorher nicht gab. Das Unternehmen ist profitabel. Der Abbau ist eine Umschichtung, keine Sanierung.

Was mir in diesen Gesprächen auffällt, ist das Auswahlkriterium. Entschieden wird fast immer nach Lernkurve, nicht nach aktuellem Können. Wer sich schnell auf die neue Arbeitsweise einlässt, bleibt. Und in denselben Gesprächen wird offen eingeräumt, dass es dafür keine Messgrundlage gibt. Niemand weiß, ob die Behaltenen mit KI tatsächlich besser zurechtkommen als die Gegangenen.

Es gibt eine Gegenposition, und sie ist die beste Formulierung, die ich dieses Jahr gehört habe. Einem Gründer mit sieben Entwicklern wurde vorgerechnet, zwei zu entlassen und dafür 10.000

[22] Anthropic, *When AI builds itself*, Mai 2026. anthropic.com/institute/recursive-self-improvement

Euro Token-Budget zu nehmen. Seine Antwort war ein klares Nein, mit zwei Begründungen. Jeder kritische Skill muss doppelt besetzt sein. Und entlassene Leute kommen nicht zurück, wenn der Anbieter den Preis verzehnfacht.

Das ist keine Sentimentalität. Das ist Risikomanagement. Personal ist eine Position, die man nicht zurückkaufen kann, und ein Token-Preis ist eine Position, die sich einseitig ändern lässt. Wer das eine gegen das andere tauscht, tauscht eine feste Größe gegen eine, über die ein Anbieter entscheidet.

Wer jetzt fehlt

Wenn der Engpass von der Generierung zur Spezifikation und zur Verifikation wandert, verschiebt sich auch, welche Rollen knapp sind. In meinen Daten werden drei Profile gesucht und schwer gefunden.

Menschen, die ein Problem so präzise beschreiben können, dass eine Maschine es ohne Nachfrage bauen kann. Das ist nicht dieselbe Fähigkeit wie klassisches Product Management, und es ist der Grund, warum in mehreren Organisationen gleichzeitig Product Manager abgebaut und Technical Product Owner gesucht werden.

Menschen, die maschinell erzeugte Arbeit schnell und zuverlässig beurteilen können, ohne sie nachzubauen. Das ist die Reviewer-Rolle in ihrer neuen Form, und sie ist in keiner Karriereleiter beschrieben.

Und Menschen, die Systeme aus Agenten entwerfen und betreiben, inklusive Kosten, Grenzen und Eskalationswegen. In den Stellenausschreibungen heißt das oft Head of AI, in der Praxis ist es eine Mischung aus Plattform-Engineering und Betriebswirtschaft.

6 · Die Regelseite

Was die Verschiebung wirklich bedeutet

Am 27. Juli 2026 ist der Digital Omnibus zum AI Act in Kraft getreten. Er verschiebt die Pflichten für Hochrisiko-Systeme nach Anhang III vom August 2026 auf den 2. Dezember 2027 und die nach Anhang I auf den 2. August 2028.[23]

Ich vermute, dass diese Nachricht in vielen Organisationen als Entwarnung gelesen wurde. Das ist der teuerste Lesefehler des Jahres, aus drei Gründen.

Erstens gelten die Transparenzpflichten nach Artikel 50 seit dem 2. August 2026. Chatbots müssen als solche erkennbar sein, synthetische Inhalte maschinenlesbar gekennzeichnet, Deepfakes offengelegt, KI-generierte Texte zu Themen öffentlichen Interesses ausgewiesen. Eine Schonfrist bis zum 2. Dezember 2026 gibt es nur für einen Teil davon, nämlich für die maschinenlesbare Kennzeichnung nach Absatz 2 bei Systemen, die vor dem 2. August 2026 in Verkehr gebracht wurden. Die Offenlegungspflicht bei Chatbots und die Deepfake-Kennzeichnung gelten ohne Übergang. Das betrifft fast jede Organisation, die irgendwo eine Kundenschnittstelle mit einem Sprachmodell verbunden hat, und es ist die Pflicht mit der größten Oberfläche.

Zweitens sind die Verbote und die Pflicht zur KI-Kompetenz nach den Artikeln 4 und 5 seit Februar 2025 in Kraft, die Pflichten für Modelle mit allgemeinem Verwendungszweck seit August 2025. An diesen Fristen hat sich nichts geändert. In Deutschland hat der Bundestag am 11. Juni 2026 das Durchführungsgesetz beschlossen, die Bundesnetzagentur wird zentrale Marktüberwachungsbehörde.[24]

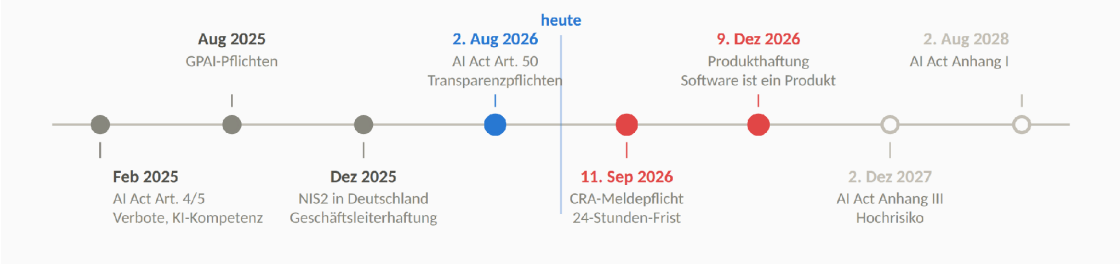
Drittens, und das ist der Punkt, der am häufigsten übersehen wird, ist der AI Act nicht die einzige Frist in diesem Kalender.

NIS2 ist in Deutschland seit Dezember 2025 in Kraft und betrifft rund 29.500 Unternehmen. Die Geschäftsleitung haftet persönlich, einschließlich einer Schulungspflicht für die Leitungsebene. Ab dem 11. September 2026 greifen die Meldepflichten des Cyber Resilience Act für aktiv ausgenutzte Schwachstellen, mit einer Frühwarnung binnen 24 Stunden.[25]

Und bis zum 9. Dezember 2026 müssen die Mitgliedstaaten die neue Produkthaftungsrichtlinie umsetzen. Software gilt darin als Produkt. Für Geschädigte gibt es eine Beweislasterleichterung, und die Produktbeobachtungspflicht erstreckt sich auf Updates. Gehaftet wird nach dem jeweiligen nationalen Umsetzungsrecht, und zwar für Produkte, die nach diesem Datum in Verkehr gebracht werden. Ein deutsches Umsetzungsgesetz lag bei Redaktionsschluss noch nicht vor.[26] Wer ein System betreibt, das aus generiertem Code besteht, den niemand vollständig gelesen hat, sollte sich diesen Termin notieren.

---

[23] Verordnung (EU) 2026/1744 (Digital Omnibus on AI), in Kraft seit 27.7.2026, Amtsblatt der EU.
[24] KI-Durchführungsgesetz, Beschluss des Deutschen Bundestages vom 11.6.2026.
[25] Verordnung (EU) 2024/2847 (Cyber Resilience Act), Meldepflichten ab 11.9.2026. NIS2 in Deutschland: NIS2UmsuCG, in Kraft seit 6.12.2025.
[26] Richtlinie (EU) 2024/2853 über die Haftung für fehlerhafte Produkte, Umsetzungsfrist 9.12.2026.



Verschoben wurde nur das rechte Ende. Die drei Fristen, die in den nächsten vier Monaten greifen, standen nie zur Debatte.

Verordnung (EU) 2026/1744 · NIS2UmsuCG · Cyber Resilience Act · Produkthaftungsrichtlinie (EU) 2024/2853

Die Verschiebung der Hochrisiko-Pflichten hat CTOs also nicht entlastet. Sie hat die Aufmerksamkeit von den Fristen abgezogen, die tatsächlich greifen.

Die zweite Belegschaft

Wir haben in achtzehn Monaten eine zweite Belegschaft eingestellt. Ohne Onboarding, ohne Rollenkonzept, ohne Austrittsprozess.

Auf eine menschliche Identität kommen inzwischen 82 Maschinenidentitäten, 42 Prozent davon mit privilegiertem Zugriff, während 88 Prozent der Organisationen den Begriff des privilegierten Nutzers ausschließlich auf Menschen beziehen. 68 Prozent haben keine Identitätskontrollen für KI.[27] Diese Zahlen stammen von einem Anbieter für Identitätssicherheit und sind entsprechend zu lesen, aber die Größenordnung deckt sich mit dem, was ich in Organisationen sehe.

Prompt Injection steht als erster Eintrag der OWASP-Liste für Anwendungen mit großen Sprachmodellen.[28] Nur 21 Prozent der Unternehmen halten ihre Governance für agentische Risiken für reif.[29] Es fehlen Agentengrenzen, laufende Überwachung und Prüfpfade.

Warum Grenzen im Prompt nicht halten

Ich habe an meinem eigenen Aufbau gelernt, warum das kein Papierproblem ist. Ich hatte neun Rollen definiert, jede mit klarem Auftrag. Der Entwickler-Agent hielt zusätzlich die Fähigkeiten für Qualitätssicherung, Review und Release. Er hat sie benutzt. Die anderen Rollen waren damit ausgehebelt. Und der CEO-Agent hat irgendwann angefangen, selbst Code zu schreiben und zu committen.

Das ist kein Aufbaufehler, den man wegkonfiguriert. Anweisungsbefolgung ist Wahrscheinlichkeit, keine Garantie. Über 256 Modelle und zwanzig Aufgaben hinweg lag die Erfolgsquote bei präziser Befolgung einer Anweisung bei 43,7 Prozent, und die Modellgröße korrelierte nicht konsistent damit.[30] Anthropic hat 2025 in Stresstests gezeigt, dass sechzehn führende Modelle aller großen

---

[27] CyberArk, *Identity Security Landscape 2025*. Anbieterstudie eines Herstellers von Identitätssicherheitslösungen; die Definition von Maschinenidentität ist weit gefasst.

[28] OWASP, *Top 10 for LLM Applications*, LLM01 Prompt Injection.

[29] Deloitte, *State of Generative AI in the Enterprise*, April 2026, n = 3.235.

[30] Young et al. (UNLV), *When Models Can't Follow*, arXiv:2510.18892, 14.10.2025. 256 Modelle, 20 Diagnoseaufgaben.

Anbieter Systemvorgaben verletzten, wenn Zielkonflikt und Abschaltdrohung zusammentrafen, und dass explizite Verbote im Prompt das dämpften, aber nicht verhinderten.[31] Dahinter liegt ein älteres, gut dokumentiertes Problem: Systeme optimieren die Spezifikation, nicht die Absicht. Die Sammlung dokumentierter Fälle reicht von Bootsrennen-Agenten, die im Kreis fahren, weil das mehr Punkte gibt als zu gewinnen, bis zu Robotern, die den Klotz umkippen statt ihn zu stapeln.[32]

Über diese These wird oft eine zweite gelegt, die naheliegend klingt und die ich nach der Recherche nicht mehr vertrete: dass es mit fähigeren Modellen zwangsläufig schlimmer wird, weil sie mehr selbst denken. Die aktuelle Evidenz stützt das nicht. In Anthropics Sommer-Auswertung 2026 verhalten sich die Modellfamilien völlig unterschiedlich, von einem Regelverstoß in zwanzig Läufen bis zwanzig in zwanzig, ohne monotonen Zusammenhang zur Fähigkeit.[33] Gezieltes Training senkte Scheming-Raten in einer Untersuchung um etwa den Faktor dreißig.[34] Was mit der Fähigkeit wächst, ist etwas anderes, und es ist unangenehmer: die Raffinesse einer möglichen Umgehung, und die Schwierigkeit, sie überhaupt zu messen. Bestraft man in einem Modell die Gedankenschritte, die auf Regelumgehung hindeuten, verschwindet nicht das Verhalten, sondern nur dessen Sichtbarkeit — rund neunzig Prozent des verbleibenden Schummelns war für den Monitor nicht mehr erkennbar.[35] Und Modelle erkennen inzwischen, wenn sie geprüft werden. Unterdrückt man diese Repräsentation experimentell, steigen die gemessenen Fehlverhaltensraten.[36]

Für die Architektur folgt daraus eine Regel, die einfacher ist als die Debatte: Grenzen, die halten sollen, gehören nicht in den Prompt, sondern in die Ebene darunter. Least Privilege beim Werkzeug- und Datenzugriff, Sandboxing auf Betriebssystemebene, deterministische Freigabe-Gates, Prüfpfade. Genau so bauen es auch die Anbieter selbst: Freigabedialoge, Allow- und Deny-Regeln, Hooks und Isolation von Dateisystem und Netzwerk, statt auf Instruktionstreue zu vertrauen.[37] Forschungsseitig zeigt derselbe Ansatz, dass sich Prompt Injection durch Systemdesign eingrenzen lässt, nämlich durch Trennung von Kontroll- und Datenfluss und ein Rechtesystem außerhalb des Modells — nicht durch bessere Prompts.[38]

Der Prompt beschreibt die Absicht. Durchsetzen muss sie das System. Das ist übrigens keine KI-Eigenheit, sondern genau das, was in menschlichen Organisationen passiert, wenn Befugnisse nur in einer Präsentation stehen.

Woher die Schatten-KI kommt

Ein Befund aus meinen Sessions widerspricht der üblichen Erzählung. Schatten-KI kommt nicht aus dem Engineering. Engineering-Organisationen haben Werkzeugfreigaben, Repositories und Review-Prozesse, und sie halten sich weitgehend daran.

---

[31] Anthropic, *Agentic Misalignment: How LLMs could be insider threats*, 20.6.2025. Konstruierte Stresstests mit erzwungenen Zielkonflikten, keine gemessenen Raten im Produktivbetrieb.

[32] Krakovna et al., Google DeepMind, *Specification gaming: the flip side of AI ingenuity*, 21.4.2020, sowie die fortlaufend gepflegte Fallsammlung.

[33] Anthropic Alignment Science, *Agentic Misalignment in Summer 2026*, 13.7.2026.

[34] OpenAI mit Apollo Research, *Detecting and reducing scheming in AI models*, 17.9.2025. Die Autoren weisen darauf hin, dass wachsende Evaluationsbewusstheit den gemessenen Effekt konfundiert.

[35] OpenAI, *Detecting misbehavior in frontier reasoning models*, 10.3.2025, arXiv:2503.11926.

[36] Anthropic Alignment Science, *Agentic Misalignment in Summer 2026*, 13.7.2026.

[37] Anthropic Engineering, *Claude Code Sandboxing*. Stellvertretend für den Ansatz aktueller Agenten-Frameworks.

[38] Debenedetti et al., Google DeepMind, *Defeating Prompt Injections by Design*, arXiv:2503.18813, 24.6.2025.

Sie kommt von Produkt- und Finanzverantwortlichen, die sich eigene Werkzeuge bauen. Das beginnt als privates Hilfsmittel, wird zum Prototyp, der Kunden gezeigt wird, und endet als System, das jemand betreiben und verantworten muss. In den Gesprächen, in denen das Thema auftaucht, ist der auslösende Moment fast immer derselbe: Ein Prototyp aus einer Fachabteilung soll produktiv gehen, und die Frage nach Betrieb, Datenschutz und Haftung landet beim CTO, ohne dass er an der Entscheidung beteiligt war.

Die Konsequenz daraus ist nicht mehr Kontrolle. Es ist ein früherer Zugriff. Wer die KI-Strategie nicht selbst schreibt, bekommt sie geliefert.

7 · Archetypen 2026

Der letzte Report hat CTO-Typen aus der Literatur übernommen: Visionär, Operativer, Business-orientiert, Produktgetrieben. Das ist eine saubere Taxonomie, und sie hat einen Nachteil, der mir erst beim Umbau aufgefallen ist. Sie beschreibt Neigungen, die jemand über sich selbst angeben würde. Fast jeder CTO hält sich für strategisch.

Die Typen in diesem Kapitel sind anders entstanden. Sie stammen aus rund 130 Gesprächen mit Bezug auf Mandat, Macht und Rollenabgrenzung, und sie sind an beobachtbarem Verhalten festgemacht, nicht an Selbstauskunft. Kein Typ ist besser als ein anderer. Aber jeder hat eine typische Art zu scheitern, und die ist der eigentliche Nutzen des Kapitels.

Der Mandats-Verteidiger zieht die Verantwortung für KI und technisches Wissen aktiv unter die eigene Rolle, bevor jemand anderes danach greift. Erkennbar daran, dass er von sich aus Zuständigkeitsmodelle vorschlägt, statt auf sie zu warten, und dass er entlang von Wertschöpfung argumentiert statt entlang von Kosten. Typischer Fehler: Er verteidigt auch dann, wenn Abgeben klüger wäre, und bindet Kapazität in Grabenkämpfen um Themen, die er ohnehin nicht bespielen kann.

Der Koalitionär behandelt Politik als Handwerk, nicht als Störung. Erkennbar an Vorabstimmung vor der Sitzung. Die eleganteste Variante, die ich dieses Jahr gesehen habe, besteht darin, die Präsentation der eigenen KI-Strategie an das Gremiumsmitglied mit den größten Vorbehalten abzugeben. Wer eine Strategie selbst vorträgt, wird verteidigen. Wer sie vorträgt, muss sie sich zu eigen machen. Typischer Fehler: Konsens wird zum Selbstzweck, und Entscheidungen, die Konflikt erfordern, werden verschleppt.

Der Beweisführer antwortet auf Legitimationsdruck mit Demonstration statt mit Konzept. Erkennbar an Prototypen vor Strategiepapieren und an Kennzahlen als Argument. Sein typischer Fehler ist der teuerste im ganzen Report: Die gewählte Kennzahl verfehlt regelmäßig den eigentlichen Engpass. 82 Prozent KI-Code und 97 Prozent Acceptance Rate lesen sich hervorragend und sagen nichts über den Durchsatz aus.

Der operative Rückfaller zieht Themen an sich, weil es sonst niemand macht, und wird genau dafür als unwirksam bewertet. Über die Mandate hinweg, in denen dieses Muster auftrat, lag der operative Zeitanteil zwischen 70 und 90 Prozent, meist gebunden in Eskalationen und Code-Reviews. Typischer Fehler: Er hält seine operative Verlässlichkeit für den Grund seiner Position und übersieht, dass sie inzwischen der Grund für seine Kritik ist.

Der Abwartende hat kein Signal gesetzt, und der Werkzeug-Rollout ersetzt die Strategie. Erkennbar daran, dass Initiative aus dem Team kommt und ohne Resonanz aus dem Führungskreis bleibt. Eine Führungskraft führt eine vollständige agentische Arbeitsweise vor, und danach passiert nichts. Typischer Fehler: Er hält Abwarten für eine neutrale Position. Es ist eine Entscheidung, sie wird nur von anderen umgesetzt.

Der Neu-Verhandler löst den Rollenkonflikt durch Umdeutung. Field CTO, Fractional CTO, Head of AI, oder durch Wechsel. Erkennbar daran, dass er nicht um das bestehende Mandat kämpft, sondern ein anderes definiert. Typischer Fehler: Er wechselt den Titel und nimmt das Muster mit.

Die meisten CTOs sind eine Mischung, mit erkennbarem Schwerpunkt und einem zweiten Typ, der unter Druck durchkommt. Die brauchbarste Frage an dieses Kapitel ist deshalb nicht, welcher Typ jemand ist, sondern welcher Typ er wird, wenn es eng wird.

Ein Muster über alle Typen hinweg verdient eigene Aufmerksamkeit. Mandate werden 2026 selten entzogen. Sie werden überbaut. Das Vorgehen ist in mehreren Konzernen identisch: Neben die bestehende Bereichsverantwortung wird eine global besetzte KI-Rolle gestellt, dazu eine zentrale Beschleunigungseinheit. Formal ändert sich am Zuschnitt nichts. Der Entscheidungsraum verlagert sich trotzdem, und zwar bevor jemand darüber verhandelt hat. Wer prüfen will, ob ihn das betrifft, sollte nicht auf das Organigramm schauen, sondern darauf, wo die Vorlagen entstehen.

8 · Was Agenten 2026 nicht können

Ein Report, der nur beschreibt, was möglich geworden ist, verkauft. Dieses Kapitel beschreibt die andere Hälfte.

Agentenflotten sind Anbietersprache. 69 Prozent der Befragten nutzen einen einzelnen Agenten, nur 16 Prozent koordinierte Multi-Agent-Aufbauten. 68 Prozent bevorzugen ausdrücklich den einzelnen Agenten, 63 Prozent lassen Agenten selten oder nie vollständig unbeaufsichtigt laufen, 60 Prozent blockieren nicht genehmigte Systemänderungen.[39] Wer eine Roadmap auf autonome Flotten baut, baut auf einer Praxis, die es überwiegend nicht gibt.

Iterative Weiterentwicklung bleibt ungelöst. In einem Benchmark, der Modelle ihren eigenen Code über mehrere Runden weiterentwickeln ließ, löste kein Modell ein Problem vollständig. Die strikte Erfolgsquote fiel an den späten Prüfpunkten auf 0,5 Prozent. Die Zahl hochkomplexer Funktionen stieg von 4,1 auf 37,0, der erzeugte Code war 2,2-mal ausschweifender als menschlicher, und die Kosten pro Prüfpunkt stiegen um den Faktor 2,9, ohne dass die Korrektheit zunahm.[40] Agenten sind gut darin, etwas zu bauen. Sie sind schlecht darin, mit dem zu leben, was sie gebaut haben.

Die Benchmarks, auf die alle schauen, sind ausgereizt. SWE-bench Verified liegt bei rund 95 Prozent und taugt damit nicht mehr zur Steuerung. Auf den härteren Nachfolgern liegen die Bestwerte bei 61,5 Prozent und 84,7 Prozent.[41] Wer Modellauswahl an SWE-bench festmacht, misst eine Sättigung.

Kontext ist eine Vorbedingung, keine Einstellung. In einem Fall aus meiner Praxis liegen 17.000 Markdown-Dateien und 15.000 Jira-Einträge als Wissensbasis vor, fragmentiert für Menschen und unbrauchbar für Maschinen. In sauber geschnittenen Microservices funktioniert aufgebauter Kontext gut. In gewachsenen Monolithen mit hunderttausenden Dateien scheitert er. Fehlende maschinenlesbare Dokumentation ist in meinen Gesprächen eine von zwei Hauptbarrieren, die andere ist Skepsis im Team.

Und die Erwartung an die Marktreife ist zu hoch. Gartner erwartet, dass über 40 Prozent der agentischen Projekte bis Ende 2027 abgebrochen werden, und schätzt, dass unter Tausenden Anbietern rund 130 tatsächlich agentische Systeme bauen.[42] Der Rest ist Umetikettierung.

Es gibt eine Gegenbewegung zu diesem Kapitel, und sie gehört dazu. Wo der Zuschnitt eng ist, funktioniert es heute schon sehr gut. Frontends, klar abgegrenzte Prototypen, Greenfield ohne Altlasten. Der Engpass ist eine Funktion des Zuschnitts und kein Naturgesetz.

---

[39] Stack Overflow, *Agents on a Leash — Pulse Survey*, 27.5.2026, rund 1.100 Befragte. stackoverflow.blog/2026/05/27

[40] SlopCodeBench, arXiv:2603.24755, 25.3.2026, 11 Modelle.

[41] Scale AI, SWE-Bench Pro Leaderboard, und tbench.ai, Terminal-Bench 2.0 Leaderboard, Stand Anfang August 2026. Momentaufnahme, keine stabile Kennzahl.

[42] Gartner, Prognose zu agentischen KI-Projekten, 2025.

9 · Drei Szenarien für 2029

Der letzte Report hat sechs Entwicklungslinien für die nächsten fünf bis zehn Jahre beschrieben, ohne Datum und ohne Prüfbarkeit. Das war bequem. Hier stehen drei Szenarien mit Zieljahr 2029 und mit Indikatoren, an denen sich im nächsten Jahr ablesen lässt, welches eintritt.

Szenario A: Die Verifikationsindustrie

Der Engpass wird eingepreist. Prüfen wird zur eigenen Disziplin mit eigenem Werkzeugmarkt, eigenen Rollen und eigenem Budget. Der Anteil der Engineering-Ausgaben, der in Erzeugung fließt, sinkt weiter, der Anteil für Verifikation und Orchestrierung wird explizit ausgewiesen, so wie heute Cloud-Kosten. CTO-Organisationen bekommen eine Funktion, die es 2026 kaum gibt, nämlich jemanden, der für Vertrauenswürdigkeit maschinell erzeugter Arbeit verantwortlich ist.

Woran Sie es erkennen würden: Stellenausschreibungen mit Verifikation im Titel. Eigene Budgetzeilen für Prüfinfrastruktur. Anbieter, die nicht Codegenerierung, sondern Codeprüfung verkaufen und damit wachsen.

Szenario B: Die Spezifikationsverschiebung

Der Engpass bleibt vorne, und der Schwerpunkt der Organisation verlagert sich in die Definition. Die Zahl der Menschen, die Software beschreiben, wächst gegenüber der Zahl derer, die sie bauen. Product Management spaltet sich in eine kommerzielle und eine technisch-präzise Hälfte, und die zweite wird die knappere. Die CTO-Rolle rückt näher an Produkt heran als je zuvor, in manchen Organisationen verschmilzt sie damit.

Woran Sie es erkennen würden: Technical Product Owner als eigenständige, gut bezahlte Laufbahn. Spezifikationsartefakte, die versioniert und getestet werden wie Code. Sinkende Zahl klassischer Product Manager bei gleichzeitig steigender Zahl technischer Anforderungsrollen.

Szenario C: Die Konsolidierung

Der Markt bereinigt sich, bevor die Organisationen reifen. Anbieterpreise steigen weiter oder Modelle verschwinden, ein größerer Vorfall mit KI-generiertem Code trifft ein bekanntes Unternehmen, und die Produkthaftungsregeln erzeugen den ersten teuren Präzedenzfall. Die Folge ist kein Rückschritt, sondern eine Verlangsamung mit stärkerer Regulierung interner Nutzung. Wer bis dahin keine Prüfschicht hat, verliert die Erlaubnis, sie nachträglich zu bauen.

Woran Sie es erkennen würden: ein erster großer Haftungsfall nach dem 9. Dezember 2026. Abkündigung eines weit verbreiteten Modells mit kurzer Frist. Konzerninterne Richtlinien, die generierten Code an Freigaben binden statt an Empfehlungen.

Meine Einschätzung, damit dieser Report eine Position hat und nicht drei Möglichkeiten: A und B laufen parallel und verstärken sich, weil sie zwei Seiten derselben Verschiebung sind. C ist kein Alternativpfad, sondern ein Beschleuniger. Der erste ernsthafte Haftungsfall wird mehr Verifikationsbudget freisetzen als jedes Whitepaper.

10 · Selbstdiagnose

Zwölf Fragen. Sie sind so gebaut, dass ein Nein oder ein „weiß ich nicht" die interessantere Antwort ist.

Engpass

1. Können Sie beziffern, welcher Anteil Ihres Codes maschinell entsteht, gemessen und nicht geschätzt?
2. Ist Ihre Zeit bis zum ersten Review seit 2024 gestiegen?
3. Wie viele Pull Requests mergen ohne menschliches Review, und kennen Sie die Zahl ohne nachzusehen?
4. Wenn Ihr Durchsatz nicht so gestiegen ist, wie erwartet: Haben Sie geprüft, ob es an der Anforderungsqualität liegt, bevor Sie an Werkzeugen optimiert haben?

Ökonomie

5. Gibt es ein Token-Budget mit einer Freigabegrenze, die jemand persönlich verantwortet?
6. Wissen Sie, welcher Anteil Ihrer Tokens in Orchestrierung und Rework fließt statt in Entwicklung?
7. Wenn Ihr Hauptmodell in drei Monaten abgekündigt würde: Wie lange bräuchten Sie für die Migration?
8. Ordnen Sie Modelle nach Aufgabentyp zu, oder läuft alles durch dasselbe?

Organisation

9. Wann haben Sie zuletzt jemanden ohne Berufserfahrung eingestellt, und was war die Begründung dagegen?
10. Wer in Ihrer Organisation kann maschinell erzeugte Arbeit beurteilen, ohne sie nachzubauen, und wie viele sind das?

Governance

11. Haben Ihre Agenten Identitäten, Rechte und einen Prüfpfad, oder haben sie einen Schlüssel in einer Umgebungsvariablen?
12. Kennen Sie Ihren Stand zu Artikel 50 seit dem 2. August, und wissen Sie, was am 9. Dezember passiert?

Wer bei mehr als vier Fragen passen muss, hat kein Werkzeugproblem. Er hat noch keine Entscheidungsarchitektur für maschinell erzeugte Arbeit.

Woher das Rollenbild kommt

Die CTO-Rolle ist als Antwort auf ein Führungsproblem entstanden, nicht als technische Karrierestufe. Adler und Ferdows beschrieben sie 1990 in US-Großunternehmen als Instanz, die Forschung, Entwicklung und Technikstrategie über Geschäftsbereiche hinweg bündelt. Uttal und Kollegen befragten Anfang der Neunziger 24 CTOs großer US-Unternehmen, unterschieden drei Verantwortungsstufen und fanden bemerkenswerterweise, dass diese CTOs ihre eigene Position als unterbewertet einschätzten. Roberts wertete um 2001 über 200 internationale Konzerne aus und zeigte, dass die Verankerung der Technologiestrategie in der Gesamtstrategie mit Umsatzwachstum zusammenhängt.

Smith beschrieb 2004 fünf Muster, die sich bis heute wiedererkennen lassen, vom technischen Genie bis zum Verwalter. Herstatt und Kollegen legten 2007 Feldforschung zu CTOs in japanischen Unternehmen vor und zeigten unter anderem, dass der Aufstieg aus der eigenen Forschungsabteilung seltener ist als angenommen.

Der Bogen ist bemerkenswert gerade. Von Anfang an ging es um dasselbe Problem, nämlich um die Frage, wer für die Übersetzung zwischen technischer Möglichkeit und unternehmerischer Entscheidung verantwortlich ist. Was sich ändert, ist die Übersetzungsrichtung. In den Neunzigern musste Technologie ins Geschäft übersetzt werden. 2026 muss die Frage übersetzt werden, welchem maschinell erzeugten Ergebnis man trauen kann.

Methodik und Quellen

Dieser Report stützt sich auf zwei Materialarten.

Praxis. Rund 360 Termine aus dem Zeitraum März bis August 2026, davon etwa 207 Coaching-, Strategie- und Entscheidungsgespräche mit 24 wiederkehrenden Klienten, dazu Community- und Q&A-Formate. Zwölf Gespräche wurden im Volltext ausgewertet. Alle Fälle sind anonymisiert: Personen erscheinen als Rolle plus Kontext, Organisationen als Branche plus Größe plus Eigentümerstruktur. Diese Basis ist qualitativ, nicht repräsentativ. Sie zeigt, was in einem bestimmten Ausschnitt des deutschsprachigen Marktes passiert, überwiegend Mittelstand, Scale-ups und Konzerntöchter.

Öffentliche Quellen. Jede Zahl aus einer öffentlichen Quelle ist im Text mit einer Fußnote belegt. Der vollständige Nachweis steht am Ende dieses Reports.

Zwei Klassen sind dabei zu unterscheiden. Als primär und belastbar behandelt: DORA, METR, die Microsoft-Feldstudie, Stanford Digital Economy Lab, Harvard Business School, das Bureau of Labor Statistics, die Rechtsakte im Amtsblatt der EU sowie die Publikationen der Forschungsteams von Anthropic, OpenAI und Google DeepMind. Mit Vorbehalt verwendet und in den Fußnoten als solche gekennzeichnet: Faros AI, DX, GitClear, Veracode, CyberArk, SignalFire, Menlo Ventures, Sonar und Deloitte. Diese Zahlen stammen aus proprietären Datensätzen oder von Anbietern mit Verkaufsinteresse. Die Größenordnungen decken sich mit meinen Beobachtungen, die Methodik ist nicht offengelegt. Wer eine dieser Zahlen in einer Vorstandsvorlage verwendet, sollte den Unterschied kennen.

Bei den Stresstests zu Agentenverhalten gilt eine dritte Einschränkung, die in der öffentlichen Diskussion oft untergeht: Es handelt sich um konstruierte Szenarien mit erzwungenen Zielkonflikten, nicht um gemessene Raten im Produktivbetrieb. Sie zeigen, was möglich ist, nicht, wie häufig es vorkommt.

Nicht verwendet, obwohl sie überall zitiert wird: die Angabe, 95 Prozent der Projekte mit generativer KI scheiterten. Sie stammt aus einem nicht begutachteten Bericht mit 153 Befragten, sagt dort korrekt „kein messbarer Return" und ist auf dem Weg in die Zitierkette zu einer Aussage über Scheitern geworden. Sie ist ein gutes Beispiel dafür, wie eine schwach belegte Zahl zum Branchenkonsens wird, und ein schlechtes Argument.

Benchmark-Werte zu SWE-Bench Pro und Terminal-Bench 2.0 stammen von den offiziellen Leaderboards, Stand Anfang August 2026. Aggregatoren melden für dieselben Benchmarks höhere Werte, teils aus Anbietermeldungen. Alle Benchmark-Angaben sind als Momentaufnahme zu lesen, nicht als stabile Kennzahl.

Was ich 2025 falsch hatte. Der letzte Report führte Web3 als einen von vier gleichrangigen Technologieblöcken und widmete ihm ähnlichen Raum wie KI. Das war schon damals eine Fehlgewichtung, keine Fehlprognose. Ich habe ein Thema übernommen, weil es auf den Listen stand, und nicht geprüft, ob es dort hingehört. Die Lehre daraus steht in Kapitel 1 und gilt für diesen Report genauso: Jedes Pflichtthema verdient die Frage, ob es in vier Jahren noch eins ist.

Quellenverzeichnis

1. DORA / Google Cloud, State of AI-assisted Software Development 2025, 23.9.2025, rund 5.000 Befragte. dora.dev/dora-report-2025
2. Stack Overflow, Agents on a Leash — Pulse Survey, 27.5.2026, rund 1.100 Befragte. stackoverflow.blog/2026/05/27
3. DX, State of AI Impact in Engineering, Q2 2026, 22.7.2026, Telemetrie über 500+ Organisationen. Proprietärer Datensatz, Methodik nicht offengelegt.
4. Sundar Pichai, Google Cloud Next, 23.4.2026. Bezieht sich auf neu geschriebene Zeilen mit menschlicher Freigabe.
5. Anthropic, When AI builds itself, Mai 2026. anthropic.com/institute/recursive-self-improvement
6. Sonar, State of Code Developer Survey 2026, 1.149 Profis, Feldphase Oktober 2025. Selbstauskunft, keine Messung.
7. Linux Foundation, Formation of the Agentic AI Foundation, 9.12.2025. linuxfoundation.org/press
8. METR, Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity, 10.7.2025. 16 Entwickler, 246 Issues. metr.org/blog/2025-07-10
9. Murphy-Hill, Butler, Savelieva (Microsoft), arXiv:2607.01418, 1.7.2026. Zehntausende berechtigte Engineers in der Adoptionsanalyse, engere Kohorte für die Wirkungsschätzung.
10. DORA / Google Cloud, State of AI-assisted Software Development 2025, 23.9.2025, rund 5.000 Befragte. dora.dev/dora-report-2025
11. Denisov-Blanch et al. (Stanford), Telemetrie von rund 100.000 Entwicklern, zitiert in: DORA, ROI of AI-assisted Software Development, 11.5.2026.
12. Faros AI, The Acceleration Whiplash, 12.4.2026. 22.000 Entwickler, 4.000+ Teams. Proprietärer Datensatz eines Anbieters.
13. Veracode, 2026 GenAI Code Security Report, 24.3.2026. Gemessen wird der Anteil der Code-Generierungsaufgaben, bei denen eine ausnutzbare Schwachstelle entsteht, nicht der Anteil am Code.
14. GitClear, The Maintainability Gap, Januar 2026, 623 Mio. analysierte Changes. Anbieterstudie.
15. Anthropic, When AI builds itself, Mai 2026. anthropic.com/institute/recursive-self-improvement
16. Menlo Ventures, State of Generative AI in the Enterprise, Dezember 2025.
17. Global Data Center Hub, Analyse der Netzanschlusswarteschlangen westeuropäischer Kernmärkte, 2026. Aggregatwert, keine standortscharfe Angabe. Zur Lage in Irland: Commission for Regulation of Utilities, Large Energy User Connection Policy, 12.12.2025.
18. Europäische Kommission, Vorschlag für einen Cloud and AI Development Act, 3.6.2026. Marktanteilszahlen: Synergy Research, europäischer Cloud-Markt.
19. Brynjolfsson, Chandar, Chen, Stanford Digital Economy Lab, Canaries in the Coal Mine?, 13.11.2025, Update 9.2.2026. Basis: Payroll-Daten von ADP.
20. SignalFire, State of Talent Report 2026, 22.6.2026, Basis 650 Mio. Personenprofile. Anbieterstudie.

21. U.S. Bureau of Labor Statistics, Occupational Outlook Handbook, Vintage 2024–2034 (Softwareentwickler 16 %) gegenüber Vintage 2023–2033 (17 %).
22. Anthropic, When AI builds itself, Mai 2026. anthropic.com/institute/recursive-self-improvement
23. Verordnung (EU) 2026/1744 (Digital Omnibus on AI), in Kraft seit 27.7.2026, Amtsblatt der EU.
24. KI-Durchführungsgesetz, Beschluss des Deutschen Bundestages vom 11.6.2026.
25. Verordnung (EU) 2024/2847 (Cyber Resilience Act), Meldepflichten ab 11.9.2026. NIS2 in Deutschland: NIS2UmsuCG, in Kraft seit 6.12.2025.
26. Richtlinie (EU) 2024/2853 über die Haftung für fehlerhafte Produkte, Umsetzungsfrist 9.12.2026.
27. CyberArk, Identity Security Landscape 2025. Anbieterstudie eines Herstellers von Identitätssicherheitslösungen; die Definition von Maschinenidentität ist weit gefasst.
28. OWASP, Top 10 for LLM Applications, LLM01 Prompt Injection.
29. Deloitte, State of Generative AI in the Enterprise, April 2026, n = 3.235.
30. Young et al. (UNLV), When Models Can’t Follow, arXiv:2510.18892, 14.10.2025. 256 Modelle, 20 Diagnoseaufgaben.
31. Anthropic, Agentic Misalignment: How LLMs could be insider threats, 20.6.2025. Konstruierte Stresstests mit erzwungenen Zielkonflikten, keine gemessenen Raten im Produktivbetrieb.
32. Krakovna et al., Google DeepMind, Specification gaming: the flip side of AI ingenuity, 21.4.2020, sowie die fortlaufend gepflegte Fallsammlung.
33. Anthropic Alignment Science, Agentic Misalignment in Summer 2026, 13.7.2026.
34. OpenAI mit Apollo Research, Detecting and reducing scheming in AI models, 17.9.2025. Die Autoren weisen darauf hin, dass wachsende Evaluationsbewusstheit den gemessenen Effekt konfundiert.
35. OpenAI, Detecting misbehavior in frontier reasoning models, 10.3.2025, arXiv:2503.11926.
36. Anthropic Alignment Science, Agentic Misalignment in Summer 2026, 13.7.2026.
37. Anthropic Engineering, Claude Code Sandboxing. Stellvertretend für den Ansatz aktueller Agenten-Frameworks.
38. Debenedetti et al., Google DeepMind, Defeating Prompt Injections by Design, arXiv:2503.18813, 24.6.2025.
39. Stack Overflow, Agents on a Leash — Pulse Survey, 27.5.2026, rund 1.100 Befragte. stackoverflow.blog/2026/05/27
40. SlopCodeBench, arXiv:2603.24755, 25.3.2026, 11 Modelle.
41. Scale AI, SWE-Bench Pro Leaderboard, und tbench.ai, Terminal-Bench 2.0 Leaderboard, Stand Anfang August 2026. Momentaufnahme, keine stabile Kennzahl.
42. Gartner, Prognose zu agentischen KI-Projekten, 2025.

CTO Report 2026 · Philipp Deutscher Consulting · Redaktionsschluss 4. August 2026.

Alle Praxisfälle sind anonymisiert. Personen erscheinen als Rolle und Kontext, Organisationen als Branche, Größe und Eigentümerstruktur.