Malorean Corp.

Ein Roman zum Wiedererkennen

Ein Panorama zum Verstehen

Entwickler, KI und kleines Unternehmen : eine (fast) wahre Geschichte der Transformation

Was Teams eines ERP-SaaS heute mit KI wirklich erleben



LAURENT DESÉCHALLIERS

Malorean Corp

> *Noch vor Kurzem brauchte man manchmal Wochen, um aus einer Idee einen Prototyp zu machen. Heute kann ein Mensch mithilfe künstlicher Intelligenz dasselbe in wenigen Tagen schaffen. Das ist eine bemerkenswerte Veränderung. Doch der eigentliche Umbruch könnte ganz woanders stattfinden: in der Art, wie wir zusammenarbeiten.*

Laurent Deséchalliers

Hinweis zur Übersetzung

Diese Ausgabe ist eine maschinelle Übersetzung, von einer künstlichen Intelligenz aus dem französischen Text erstellt. Sie wurde nicht von einem menschlichen Übersetzer geprüft. Die Gedanken, der Aufbau und die Beispiele sind dem Original treu; einzelne Formulierungen können dagegen ungelenk klingen, und manche Stellen tragen noch Spuren der Ausgangssprache.

Das ist der Preis dafür, dass dieses Buch in mehreren Sprachen existiert und nicht nur auf Französisch. Bei Zweifeln an einer Formulierung gilt die französische Ausgabe.

Einleitung

1

Was mich an KI am meisten beeindruckt, ist nicht, dass sie Code schreiben kann. Es ist die Geschwindigkeit, mit der sie Dinge möglich macht, für die man noch vor Kurzem mehrere Tage oder Wochen gebraucht hätte.

Aus einer Idee kann ein Prototyp werden, noch bevor die Diskussion über die Machbarkeit abgeschlossen ist. Ein Entwickler kann einen Ansatz ausprobieren, ohne darauf warten zu müssen, dass ein ganzes Team Zeit hat. Manchmal kann sogar jemand, der nicht programmieren kann, eine erste Version seiner Idee verwirklichen. Das bedeutet weder, dass ein Prototyp schon ein zuverlässiges Produkt ist, noch dass die Komplexität verschwunden wäre. Doch allein die Möglichkeit, etwas auszuprobieren, verändert unseren Blick darauf, was machbar erscheint.

Unsere Unternehmen wurden allerdings nicht für eine Welt aufgebaut, in der eine erste Lösung so schnell entstehen kann. Sie haben sich nach Kompetenzen, verfügbarer Zeit, Übergaben und den Entscheidungen organisiert, die nötig waren, damit die Arbeit vorankam. Diese Abläufe haben gute Gründe. Wenn sich jedoch die Fähigkeiten der Menschen verändern, werfen sie neue Fragen auf. Müssen wir weiterhin in derselben Reihenfolge arbeiten? Auf dieselben Freigaben

warten? Dieselben Tätigkeiten denselben Berufen vorbehalten?

Diese Fragen betreffen nicht nur Entwickler. Auch Product Manager, Designer, Vertriebsmitarbeiter, Teamleiter und viele andere entdecken neue Arbeitsweisen. Manche können heute selbst Aufgaben übernehmen, für die sie früher Kollegen brauchten. Bei anderen verändert sich die Art ihrer Arbeit. Was zunächst wie eine Frage der Werkzeuge aussah, betrifft inzwischen unsere Zusammenarbeit und die Art, wie wir Entscheidungen treffen.

Es wäre verlockend, eine Revolution auszurufen, deren Ausgang bereits feststeht. Ich glaube nicht, dass wir so weit sind.

Manche Unternehmen erproben eine neue Verteilung von Arbeit und Verantwortung. Andere gehen vorsichtiger vor, manchmal aus guten Gründen. Einige Praktiken erscheinen vielversprechend, doch noch fehlt uns die Erfahrung, um zu wissen, welche sich im Alltag bewähren. Und was in einem kleinen Team funktioniert, ist nicht automatisch für ein größeres, stärker reguliertes oder traditionsreicheres Unternehmen geeignet.

Deshalb wollte ich kein Handbuch für die vermeintlich richtige Organisation im Zeitalter der KI schreiben. Mit diesem Buch möchte ich verstehen, was sich tatsächlich verändert: Was die Werkzeuge ermöglichen, wie Unternehmen darauf reagieren,

was sie gewinnen, was sie verlieren könnten und welche Fragen sie entscheiden müssen.

Dafür habe ich zwei Wege gewählt, die sich kreuzen.

Der erste ist die Beobachtung. Wir betrachten reale Unternehmen, öffentlich beschriebene Praktiken, Studien und Erfahrungen. Dabei versuche ich, gesicherte Erkenntnisse von Hypothesen zu unterscheiden und eine überzeugende Ankündigung nicht mit einem nachgewiesenen Ergebnis zu verwechseln.

Der zweite ist eine Geschichte. Wir begleiten Malorean Corp., ein fiktives Unternehmen, dessen Mitarbeiter nach und nach entdecken, wie die neuen Möglichkeiten ihren Arbeitsalltag verändern. Sie haben keinen Plan zur Neuerfindung ihrer Organisation. Sie haben Kunden, Gewohnheiten, dringende Aufgaben, Überzeugungen und mitunter Meinungsverschiedenheiten. Die Fiktion beweist nichts; sie lässt uns diesen Wandel aus der Perspektive der Menschen betrachten, die ihn erleben.

Beide Blickwinkel sind notwendig. Reale Beispiele helfen uns, Vermutungen nicht für Tatsachen zu halten. Die Geschichte erinnert uns daran, dass sich organisatorische Veränderungen nicht nur in Organigrammen abspielen. Sie betreffen Vertrauen, Kompetenzen, Verantwortung und den Platz jedes Einzelnen.

Im Kern zieht sich eine Frage durch die folgenden Seiten:

Was geschieht in einem Unternehmen, wenn seine

Mitarbeiter entdecken, dass sie weit mehr tun können, als ihr Beruf ihnen bisher ermöglicht hat?

Über den Autor

Ich arbeite seit mehreren Jahrzehnten in der Softwarebranche und bin heute als Product Manager tätig. Mein Weg führte mich von der Softwareentwicklung und Architektur über die Organisation von Teams zum Produktmanagement, zum operativen Produktgeschäft und zum Marketing.

Diese Erfahrung bietet mir einen besonderen Blick auf die Entwicklung der künstlichen Intelligenz und ihre Auswirkungen auf unsere Arbeit. Ich erlebe sie täglich in meinem Beruf, aber auch im Austausch mit anderen Product Managern. Wir teilen Erfahrungen, Entdeckungen und Fragen. Bei diesen Gesprächen geht es oft um mehr als nur um Werkzeuge: Was verändert sich wirklich in unseren Berufen, bei unseren Entscheidungen und in unserer Zusammenarbeit?

Mit der Zeit wurden aus diesen Beobachtungen und Diskussionen immer mehr Fragen. Ich wollte Abstand gewinnen, unsere Eindrücke mit den Erfahrungen anderer Unternehmen vergleichen und meine Überlegungen teilen. So entstand dieses Buch.

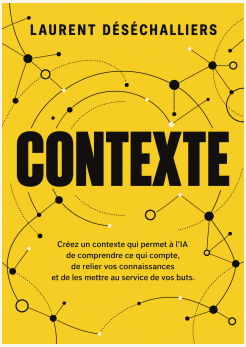
Das Schreiben begleitet mich schon lange. Ich habe Beiträge für Fachmedien verfasst und für M2 Édition und Eyrolles als technischer Lektor gearbeitet. Heute schreibe ich auf

malorean.net über Product Management, KI, Arbeitsmethoden und Informationssysteme.
Außerdem schreibe ich für *Proposition Marketing*, ein unabhängiges Magazin über strategisches und operatives Marketing. Für Leser aus Marketing und Unternehmensführung zu schreiben, motiviert mich, Produktfragen verständlich zu machen, ohne gemeinsames Fachvokabular vorauszusetzen.

Laurent Deséchalliers

- laurent@malorean.net
- https://malorean.net

Vom selben Autor



Le Contexte — Wie man einen Kontext aufbaut, der es der KI ermöglicht zu verstehen, worauf es ankommt, das eigene Wissen zu verknüpfen und das Gelernte im Dienst der eigenen Ziele zu nutzen.
Erschienen: Oktober 2026.



NO AI — *Verstehen lässt sich nicht delegieren*. Methoden ohne KI, um unser Gedächtnis zu pflegen, unser Denken zu vertiefen und unser kritisches Urteilsvermögen zu üben. Praktiken, um die eigenen Fähigkeiten zu entwickeln und die KI besser zu nutzen, ohne ihr unser Verstehen zu überlassen.
Erscheint voraussichtlich Ende 2026.

Mehr dazu auf https://malorean.net.

Danksagung

Zuerst möchte ich den Product Managern danken, mit denen ich mich regelmäßig austausche. Wir vergleichen unsere Sichtweisen auf die Entwicklung des Product Managements, der Berufe in der Softwareentwicklung und, darüber hinaus, auf die Auswirkungen künstlicher Intelligenz auf die Arbeit und die Organisation von Unternehmen. Diese manchmal lebhaften Gespräche sind eine unerschöpfliche Quelle für Fragen, neue Sichtweisen und Ideen. Manche dieser Überlegungen sind in dieses Buch eingeflossen. Danke für eure Neugier, eure Erfahrungen und eure Lust, miteinander zu diskutieren.

Und ich möchte Christine danken, meiner Frau. Nach meinem ersten Buch, *Le Contexte*, hatte ich ihr eine Pause versprochen. Stattdessen habe ich fast unmittelbar mit diesem Buch weitergemacht. Zum Glück hat uns dieses zweite Abenteuer weniger gemeinsame Abende und Wochenenden gekostet als das erste. Danke, Christine, für deine Geduld und Unterstützung – selbst dann, wenn aus meinen Gedanken schon wieder ein neues Buch wird.

Hinweis zu den Beispielen

Die Überlegungen und Beispiele in diesem Buch beruhen auf mehr als dreißig Jahren Berufspraxis in der IT. Sie gehen auf Situationen aus unterschiedlichen Projekten und Arbeitsumfeldern sowie auf zahlreiche Gespräche mit Product Managern, CTOs, Engineering Managern und anderen Fachleuten aus verschiedenen Unternehmen zurück. Sie stammen damit aus mehreren Organisationen und verschiedenen Phasen meines Berufswegs.

Für dieses Buch habe ich solche Situationen verändert, verallgemeinert oder neu zusammengesetzt. Die Anwendungsfälle, Zahlen und Kennzahlen veranschaulichen wiederkehrende Probleme und was ich daraus gelernt habe. Es sind konstruierte Beispiele: Keines beschreibt einen realen Kunden, ein reales Produkt oder eine reale Organisation. Ebenso wenig geben sie Daten, Arbeitsweisen oder Positionen meines Arbeitgebers wieder.

Die Analysen, Methoden und Meinungen in diesem Buch sind persönlich und geben ausschließlich meine eigene Auffassung wieder.

Wurde dieses Buch mit KI geschrieben?

Ja. Vollständig in Zusammenarbeit mit künstlicher Intelligenz.

Ich weiß, dass diese Frage heute eine besondere Bedeutung hat. Man fragt sich, ob ein Artikel, ein Roman oder ein Buch mit KI geschrieben wurde. Manche lassen Texte sogar von Erkennungsprogrammen prüfen, als müsse man unbedingt unterscheiden, was von einem Menschen und was von einer Maschine stammt.

Ich ziehe es vor, offen damit umzugehen: KI war an allen Phasen dieses Buches beteiligt – von der Recherche über die Strukturierung der Ideen und die Prüfung der Quellen bis zum Schreiben und zum Aufspüren blinder Flecken. Doch diese Zusammenarbeit verdient eine Erklärung. Denn allein daraus geht nicht hervor, wer die Gedanken hinter diesem Buch entwickelt hat.

Früher konnte man sich für ein Vorhaben dieser Größenordnung mit Menschen umgeben, die Dokumente recherchierten, Ideen abglichen, Texte gegenlasen, Inhalte strukturierten oder gestalteten. KI verschafft mir heute Zugang zu einem Teil dieser Fähigkeiten. Sie erledigt

Aufgaben schnell, schlägt Formulierungen vor, macht auf Widersprüche aufmerksam und stellt manchmal Fragen, die mir selbst nicht eingefallen wären. So kann ich in der mir zur Verfügung stehenden Zeit wesentlich mehr erkunden, als ich allein schaffen könnte.

Für dieses Buch habe ich eine Kontext-Engine eingesetzt, die auf Technologien von ChatGPT beruht und Informationen dauerhaft in Notion speichert. Quellen, Ideen, Entscheidungen und Arbeitsfassungen lassen sich dadurch über einzelne Gespräche hinweg bewahren und wiederfinden. Diesen Ansatz habe ich bereits in meinem ersten Buch, *Le Contexte*, vorgestellt. Darin erkläre ich, wie ein solches System die KI zu einer Art außergewöhnlich leistungsfähigem persönlichen Sekretär machen kann: Sie beantwortet nicht bloß Fragen, sondern behält den Überblick über Projekte und hilft mir, sie langfristig voranzubringen.

Dabei gibt es jedoch einen wesentlichen Unterschied: **Etwas ausführen zu können bedeutet nicht, die Verantwortung für das Verstehen und Entscheiden zu tragen.** Die Fragen, die dieses Buch leiten, meine beruflichen Erfahrungen, die Interpretationen, Abwägungen und schließlich die Positionen, die ich vertrete oder verwerfe, stammen von mir. KI kann mir widersprechen, meine Annahmen hinterfragen und mir helfen, Gedanken präziser auszudrücken. Sie ersetzt weder mein Urteil noch meine Verantwortung als Autor.

Ich unterscheide außerdem dieses Buch von Werken, bei denen die Sprache selbst ein grundlegender Bestandteil des künstlerischen Schaffens ist. Wenn man Shakespeare in einer Übersetzung von François-Victor Hugo liest, Cervantes oder Cizia Zykë im Original, sind Stimme, Rhythmus und Stil ebenso wichtig wie das Erzählte. Bei einem übersetzten Werk gehören auch die Entscheidungen des Übersetzers zum Leseerlebnis. Welche Rolle wir KI in einer solchen Literatur zugestehen wollen, wirft andere Fragen auf, die ich hier nicht endgültig beantworten möchte.

Dieses Buch enthält fiktive Elemente, doch sein Hauptanliegen ist es, einen technischen und organisatorischen Wandel zu verstehen. Die Sprache soll angenehm und klar sein; vor allem dient sie den Ideen, Fakten und Fragen, die ich teilen möchte. KI hat mir ermöglicht, dieses Projekt neben meiner beruflichen Tätigkeit in einem Umfang zu verwirklichen, den ich mit meinen eigenen Mitteln vermutlich nicht erreicht hätte. Sie eröffnet mir die Möglichkeit, persönliche Projekte wie dieses Buch zu entwickeln, ohne auf den Anspruch sorgfältiger Reflexion zu verzichten.

Und letztlich passt es durchaus zusammen, ein Buch über die Veränderungen durch KI zu schreiben und dabei genau so zu arbeiten.

Laurent Deséchalliers

Inhalt

Hinweis zur Übersetzung 3

Einleitung 4

Über den Autor 8

Vom selben Autor 10

Danksagung 11

Hinweis zu den Beispielen 12

Wurde dieses Buch mit KI geschrieben? 13

Malorean Corp. 17

1 Malorean Corp. — Ein Tag wie kein anderer 18

2 KI hat die Entwicklungskapazität von Softwareteams vervielfacht . 26

3 Malorean Corp. — Der Tag danach 37

4 Neue Organisationsmodelle entstehen 51

5 Malorean Corp. — Was wir tun können und was wir tun sollten . 63

6 Die Rollen verschwimmen, die Verantwortung bleibt 79

7 Malorean Corp. — Was Alice wusste, ohne es je aufgeschrieben zu haben 91

8 Mehr Autonomie braucht klarere Regeln, nicht weniger 112

9 Malorean Corp. — Zwei vernünftige Entscheidungen 130

10 Wird KI die mittleren Führungsebenen im Unternehmen verschwinden lassen? 148

11 Malorean Corp. — Die wiedergewonnene Zeit 168

12 Schlusswort 190

Malorean Corp.

1 — Malorean Corp. — Ein Tag wie kein anderer

Der Hauptsitz von Malorean Corp. befand sich in Le Havre. Doch seit der Pandemie verriet die Adresse kaum noch etwas darüber, wo tatsächlich gearbeitet wurde. Die Mitarbeiter waren über ganz Frankreich verteilt. Man traf sich vor der Kamera, besprach Dinge über einen geteilten Bildschirm und musste für die meisten Aufgaben nicht mehr im selben Raum sitzen.

Malorean Corp. entwickelte ein online verfügbares ERP-System – eine jener Softwarelösungen, die nach und nach fast jeden Ablauf in einem Unternehmen berühren. Ein Kunde konnte ein Kleinstunternehmen mit wenigen Mitarbeitern und sehr konkreten Bedürfnissen sein. Ein anderer war ein großer Konzern mit komplizierten Prozessen, besonderen Anforderungen und wenig Geduld für ungenaue Lösungen.

Die meisten Kunden kamen aus Frankreich. Zwar gab es auch Nutzer in Deutschland und Spanien, doch vor allem in Tochtergesellschaften französischer Konzerne. Hubert Félix Duca, der Geschäftsführer, hatte größere Pläne. Er wollte Malorean Corp. wirklich in Europa etablieren.

Ein ERP lässt sich allerdings nicht so leicht über Landesgrenzen hinweg einsetzen wie eine einfache Unternehmenswebsite. Die Regeln ändern sich von Land zu

Land. Das gilt für die Lohnabrechnung, für Verwaltungspflichten und für die Arbeitsweise der Unternehmen. Hubert kannte diese Einschränkungen. Er kam aus der Unternehmensführung, Buchhaltung und Finanzwelt, nicht aus der Softwareentwicklung. Für ihn war eine Funktion nur dann wertvoll, wenn Kunden sie nutzen konnten, ohne dadurch in Schwierigkeiten zu geraten.

Innerhalb des Unternehmens waren die Rollen recht klar verteilt.

Zwei Teams mit jeweils fünf Entwicklern entwickelten die Software weiter. Alice und Bob, die beiden Product Manager, arbeiteten mit ihnen zusammen, um die Bedürfnisse zu verstehen, sinnvolle Verbesserungen auszuwählen und dem Produkt eine Richtung zu geben. Alice hatte früher selbst entwickelt und konnte technischen Gesprächen mühelos folgen. Bob hatte eine betriebswirtschaftliche Ausbildung und betrachtete die Themen aus einem anderen Blickwinkel. Ihre Erfahrungen unterschieden sich, aber sie arbeiteten ausgezeichnet zusammen.

Ein Vertriebsteam gewann neue Kunden und pflegte bestehende Beziehungen, unterstützt von einer Person im Marketing. Fünf Spezialisten begleiteten Interessenten vor dem Verkauf und halfen später bei der Einführung des ERP vor Ort oder aus der Ferne. Sie installierten nicht einfach die Software. Sie unterstützten die Unternehmen dabei, ihre Bedürfnisse genauer zu formulieren, Abläufe anzupassen, und

betreuten sie oft noch während der ersten sechs Monate nach dem Kauf.

Im Kundensupport arbeiteten zwölf Personen: sieben für kleinere Kunden und fünf für Großkunden. Oft erfuhren sie als Erste, dass eine Funktion den Kunden zu einem Umweg zwang, statt ihm die Arbeit zu erleichtern. Zwei Juristen kümmerten sich um die Angelegenheiten von Malorean Corp. und brachten ihr Fachwissen in die Produktentwicklung ein, insbesondere bei sensiblen Themen wie der Lohnabrechnung.

So funktionierte das Unternehmen. Nicht perfekt, natürlich. Wenn so viele Kunden eine Software einsetzen, sammeln sich Wünsche, Ausnahmen, schwer veränderbare Gewohnheiten und Funktionen an, die nur erhalten bleiben, weil niemand Zeit hat, sie neu zu gestalten. Doch jeder wusste ungefähr, was er zu tun hatte.

Dann tauchte ChatGPT in den Gesprächen auf.

Am Anfang stand kein großes Transformationsprogramm. Zwei besonders neugierige Menschen hatten es schon einige Monate zuvor ausprobiert: Julie Morel aus dem Vertrieb und Lucas Martin aus der Entwicklung. Zunächst nutzten sie es für ihre eigenen Aufgaben und erzählten später, was sie damit erreichen konnten.

Für Julie lag der Vorteil auf der Hand. Sie bereitete eine heikle E-Mail vor, ließ einen Text verständlicher formulieren, sortierte Notizen vor einem Termin oder machte aus ein paar Gedanken eine Gliederung für eine Präsentation. Manchmal

entwarf sie auch Inhalte für einzelne Folien. Es war nicht ChatGPT, das entschied, was dem Kunden gesagt werden sollte. Julie lieferte die Informationen, überarbeitete die Vorschläge, korrigierte Formulierungen und verschickte nur das, wofür sie selbst einstehen konnte.

Sie behauptete nicht, das Werkzeug kenne ihre Kunden besser als sie. Sie war einfach froh, nicht mehr eine halbe Stunde mit der Formulierung von Dingen zu verbringen, die sie längst wusste.

Ihr Kollege Philippe Garnier beobachtete das eher aus der Ferne. Er war ein erfahrener Vertriebler, der Großkunden betreute. Er kannte die Menschen hinter den Verträgen, Meinungsverschiedenheiten, die in keinem Protokoll standen, und beiläufige Bemerkungen, die den Verlauf einer Besprechung verändern konnten.

— Du solltest es mal für deine E-Mails ausprobieren, hatte Julie eines Morgens gesagt. Und wenn es nur um die Formulierung geht.

— E-Mails kann ich selbst schreiben.

Das war nicht böse gemeint. Er verstand nur nicht, warum er einen Teil seiner Arbeit einem Werkzeug überlassen sollte, wenn sein Wert doch gerade darin lag, seine Kunden zu kennen und mit ihnen sprechen zu können.

Julie drängte nicht weiter. Sie musste einen Termin vorbereiten.

Lucas wiederum nutzte ChatGPT auf ganz andere Weise. Der erfahrene Entwickler lebte in Lyon und wollte nicht nur ein paar Minuten bei wiederkehrenden Aufgaben sparen. Er verbrachte viel Zeit mit Experimenten, stellte immer anspruchsvollere Fragen und begann von vorn, wenn ihn eine Antwort nicht überzeugte. Julie wollte vor allem ein nützliches Ergebnis. Lucas wollte herausfinden, wie weit er gehen konnte.
Irgendwann erreichten die Erfahrungen der ersten Nutzer auch Hubert. Bei rund zwanzig Euro pro Person und Monat schien die Rechnung einfach: Wenn jeder auch nur ein wenig Zeit bei mühsamen Aufgaben sparte, würde sich die Investition lohnen. Malorean Corp. stellte ChatGPT deshalb allen zur Verfügung.
Das wurde nicht mit einer großen Präsentation als Revolution ausgerufen. Jeder konnte das Werkzeug nutzen, wenn es sinnvoll erschien. Manche öffneten es täglich, andere selten, Philippe überhaupt nicht. Für die meisten blieb KI eine Assistenz: Man erklärte ihr, was man wollte, sie machte einen Vorschlag, man überprüfte ihn und entschied dann selbst.
Der Alltag ging weiter.
Am Mittwoch kam in einem Gespräch zwischen Produkt- und Entwicklungsteam ein altes Thema wieder auf den Tisch: die Urlaubsverwaltung.
Niemand mochte das Modul besonders. Es war alt, unflexibel und bot nur wenige Möglichkeiten. Es funktionierte gerade gut

genug, um weiter genutzt zu werden, aber nicht gut genug, um Beschwerden zu vermeiden.

Wer Urlaub beantragen wollte, musste das am Arbeitscomputer tun. Einfach am Sonntagabend vom Handy aus ein paar Urlaubstage eintragen, während man an die Ferien dachte – das ging nicht. Noch unangenehmer war ein Fehler bei der Berechnung: Das Modul berücksichtigte Feiertage nicht richtig. Beantragte jemand Urlaub von Montag bis Freitag und war der Mittwoch ein Feiertag, konnte das System fünf statt vier Urlaubstage abziehen.

Der Support kannte das Problem, der Vertrieb ebenso. Für manche Kunden war das Modul zu einem ständigen Ärgernis geworden. Man behielt es, weil es da war, nicht weil es wirklich den Erwartungen entsprach.

— Wir müssten das endlich richtig überarbeiten, fasste jemand zusammen.

Niemand widersprach. Aber „richtig überarbeiten“ bedeutete, die Oberfläche neu zu gestalten, die Berechnungen zu korrigieren, die Bedienung auf dem Smartphone zu ermöglichen und einen Teil der veralteten Technik zu ersetzen. Man rechnete mit ungefähr zwei Monaten Arbeit. Andere Aufgaben waren dringender. Also landete das Thema wieder einmal auf der Liste der Dinge, die man irgendwann angehen müsste.

Am Freitagnachmittag bat Lucas am Ende einer Videokonferenz um ein paar Minuten.

— Ich habe euch etwas zu zeigen.

Er hatte nicht gesagt, worum es ging. Niemand wusste, dass er seit dem Mittwochsgespräch am Urlaubsmodul gearbeitet hatte.

Er teilte seinen Bildschirm. Eine übersichtliche, moderne Oberfläche öffnete sich. Die wichtigen Informationen standen dort, wo man sie erwartete.

Lucas zeigte, wie man Urlaub beantragen konnte. Dann verkleinerte er das Fenster auf die Breite eines Smartphones. Die Oberfläche passte sich an. Man konnte fortfahren, die Tage auswählen und den Antrag abschicken, ohne zum Desktop zurückkehren zu müssen.

Danach zeigte er eine Woche von Montag bis Freitag, in der der Mittwoch ein Feiertag war.

Das System berechnete vier Urlaubstage.

Er änderte die Daten. Die Berechnung aktualisierte sich. Er wechselte zurück zur mobilen Ansicht, scrollte durch mehrere Ansichten und probierte weitere Funktionen aus. Das waren keine starren Entwürfe: Die Formulare reagierten, die Berechnungen liefen, die Abläufe funktionierten.

Zunächst stellten die Kollegen gewöhnliche Fragen: Wo findet man diese Einstellung? Was geschieht, wenn man ein Datum verändert? Dann verstummten sie.

In den kleinen Videofenstern veränderten sich die Gesichter. Am Mittwoch hatten sie noch über zwei Monate Arbeit gesprochen. Am Freitag zeigte Lucas eine erste

funktionsfähige Version. Er hatte niemanden um Hilfe gebeten und niemanden informiert. Noch erklärte er auch nicht, wie ihm das gelungen war.

— Moment mal, sagte schließlich jemand. Ist das das Modul, über das wir vorgestern gesprochen haben?

Lucas bestätigte.

Es wurde still.

Alle blickten auf dieselbe Sache: eine Arbeit, für die sie Wochen eingeplant hätten, und die plötzlich innerhalb von zwei Tagen vor ihnen lag.

Niemand verstand, wie das möglich gewesen war.

2 — KI hat die Entwicklungskapazität von Softwareteams vervielfacht

Über Jahrzehnte beruhte ein großer Teil der Organisation von Softwareentwicklung auf einer einfachen Einschränkung: **Software zu entwickeln ist teuer.**

Eine Idee in ein Produkt zu verwandeln kostete Zeit, erforderte seltene Fachkenntnisse und viel Abstimmung. Eine Funktion musste eingegrenzt, in Aufgaben aufgeteilt, priorisiert, entwickelt, getestet, geprüft und bereitgestellt werden. Die Zahl verfügbarer Entwickler, ihre Erfahrung und die Zeit, die sie investieren konnten, begrenzten zwangsläufig, was ein Unternehmen umsetzen konnte.

Dieser Engpass prägte die Unternehmen. Sie lernten, die Zeit der Entwickler zu schützen, Aufgaben im Voraus vorzubereiten, Unterbrechungen zu reduzieren, Backlogs zu organisieren und Prioritäten gründlich abzuwägen, bevor etwas gebaut wurde. Ein großer Teil des modernen Product Managements entstand unter diesen Bedingungen: Man wollte vermeiden, teure Entwicklungskapazität für die falsche Idee zu verschwenden. Code-Assistenten und später Agenten, die umfangreichere Aufgaben selbstständig erledigen können, beginnen diese Rechnung zu verändern.

Es geht längst nicht mehr nur um bessere automatische Codevervollständigung

Die ersten KI-Werkzeuge für Entwickler ließen sich als neue Generation von Produktivitätswerkzeugen verstehen: eine Funktion vervollständigen, Code erklären, einen Test schreiben oder ein Refactoring vorschlagen.

Die nächste Generation verändert die Arbeit grundlegend. Ein Agent kann einen Auftrag entgegennehmen, eine Codebasis untersuchen, mehrere Dateien bearbeiten, Tests ausführen, bestimmte Fehler beheben und eine Änderung zur Prüfung vorlegen. Entwickler können damit einen beträchtlichen Teil der Ausführung delegieren und vor allem deutlich mehr Arbeit anstoßen und voranbringen.

Die Folge ist nicht einfach, dass „Entwickler schneller programmieren“.

Eine einzelne Person kann wesentlich mehr Softwarearbeit anstoßen und voranbringen als zuvor.

Einige Unternehmen berichten bereits von beeindruckenden Ergebnissen. AMPECO beschreibt den eigenen Wechsel zu einem AI-nativen Entwicklungssystem und gibt eine viermal höhere Entwicklungsgeschwindigkeit sowie 50 Prozent weniger Fehler je User Story an. Diese Zahlen sind kein Marktdurchschnitt. Es handelt sich um Angaben eines einzelnen Unternehmens unter besonderen Bedingungen. Sie zeigen jedoch die Größenordnung des angestrebten Wandels: Es geht nicht darum, beim Schreiben einer Funktion ein paar

Minuten zu gewinnen, sondern die Umsetzungskapazität ganzer Teams deutlich zu erhöhen. [1]

Solche unternehmenseigenen Angaben sind nicht die einzigen Hinweise. Ich selbst habe mit Claude Code etwas Ähnliches erlebt. In vier Tagen und mit mehr als 180 Commits entwickelte ich eine erste Produktversion mit Oberflächen, APIs und Tests. Mit meinen früheren Methoden hätte ich nach meiner Schätzung ein bis zwei Monate gebraucht. Auf Malorean.net habe ich im Juli 2026 darüber berichtet. Daraus lässt sich selbstverständlich kein allgemeiner Produktivitätsgewinn ableiten. Das Beispiel zeigt, was ein PM mit Entwicklungserfahrung heute erreichen kann – aber ebenso die Schwierigkeiten bei Qualität, Wartbarkeit und Kontrolle KI-generierten Codes. [7]

Damit verändert sich die Größenordnung der Frage.

Würde KI das Programmieren nur um einige Prozent beschleunigen, könnte die Organisation unverändert bleiben und den Gewinn aufnehmen. Wenn aber die Produktionskapazität deutlich schneller steigt, gewinnen die anderen Einschränkungen an Bedeutung.

Schneller Code bedeutet nicht automatisch ein schnelleres Unternehmen

In einem Softwareteam besteht die Arbeit nicht nur aus dem Schreiben von Code.

Bevor etwas gebaut wird, muss man ein Problem verstehen, entscheiden, ob eine Investition sinnvoll ist, eine Richtung

wählen und Einschränkungen klären. Danach folgen Prüfung, Tests, Absicherung, Bereitstellung, Beobachtung und Wartung. Und dazwischen sind ständig Entscheidungen erforderlich. Wird nur einer dieser Schritte stark beschleunigt, entsteht bei den anderen umso mehr Arbeit.

Code kann schneller entstehen, ohne dass das Kundenverständnis im gleichen Tempo wächst. Prototypen können sich vervielfachen, ohne dass jemand besser entscheiden kann, welchen man weiterverfolgen sollte. Mehr Änderungen können zur Code-Review gelangen, ohne dass sich die menschliche Aufmerksamkeit für ihre Prüfung entsprechend vermehrt.

Das Problem ist also nicht, dass KI plötzlich alle anderen Tätigkeiten überflüssig macht.

Sie **verschiebt die Engpässe**.

Was bisher durch die langsame Umsetzung verdeckt wurde, tritt deutlicher hervor: die Wahl des richtigen Problems, eine klare Absicht, fundierte Entscheidungen, Validierung, Kohärenz und die Fähigkeit, aus den Ergebnissen nach der Auslieferung zu lernen.

Mehr produzieren heißt nicht, mehr Wert schaffen

Diese Unterscheidung ist entscheidend.

Als die Umsetzung schwierig war, konnte die Menge ausgelieferter Software wie ein Maß für die Wertschöpfung eines Unternehmens erscheinen. Das war schon damals eine ungenaue Abkürzung. Mit KI wird sie gefährlich.

Ein Team kann mehr Code, mehr Pull Requests, mehr Prototypen und mehr Funktionen erzeugen, ohne das Produkt zu verbessern. Die Gesamtleistung kann sogar sinken, wenn dieser zusätzliche Umfang zu mehr Fehlern, Nacharbeiten, Kosten oder Wartungskomplexität führt.

Anders gesagt:

Etwas herstellen zu können ist nicht dasselbe wie entscheiden zu können, was überhaupt hergestellt werden sollte.

Deshalb reichen Aktivitätskennzahlen heute noch weniger aus. Die Zahl der Änderungen, die Menge an Code oder allein die Verwendung eines KI-Werkzeugs sagt nichts darüber aus, ob das Team mehr Wert geschaffen hat.

Canva bietet ein interessantes Beispiel für die Orientierung am Ergebnis: Das Unternehmen hat ein System entwickelt, mit dem Produkt-, Engineering- und Datenteams die geschäftlichen Auswirkungen ihrer Experimente einschätzen und lokale Entscheidungen unmittelbarer mit ihren Folgen für das Unternehmen verbinden können. [2]

Damit verändert sich die Frage.

Früher:

Wie viel können wir bauen?

Heute:

Was von all dem, was wir inzwischen bauen können, ist es wirklich wert?

> ***Die Frage der Softwareproduktivität ist über vierzig Jahre alt***
>
> *1986 stellte Fred Brooks* No Silver Bullet *vor, einen 1987 in* IEEE Computer *veröffentlichten Aufsatz. Er unterschied* ***zufällige beziehungsweise technische Erschwernisse*** *der Softwareentwicklung – Sprachen, Werkzeuge und die Umsetzung eines Entwurfs in Code – von der* ***wesentlichen Komplexität****: ein System verstehen, entwerfen, strukturieren und weiterentwickeln. Seiner Ansicht nach konnte keine einzelne technische oder organisatorische Neuerung versprechen, Produktivität, Zuverlässigkeit und Einfachheit innerhalb eines Jahrzehnts um den Faktor zehn zu steigern. [3]*
>
> *Der Aufsatz wurde 1995 in die Jubiläumsausgabe von* The Mythical Man-Month *aufgenommen, das ursprünglich 1975 erschienen war. [4] Generative KI kann das Ausmaß der Umsetzung verändern, aber nicht automatisch dafür sorgen, dass wir das richtige Problem wählen. Das ist eine heutige Einordnung von Brooks' Gedanken –* ***keine Vorhersage von ihm über KI-Agenten****.*

Der neue Engpass ist das Urteilsvermögen

Wenn die Umsetzung weniger kostet, werden andere Ressourcen im Verhältnis wichtiger: Kundenkenntnis, Verständnis des Problems, strategischer Kontext, die Fähigkeit, eine klare Absicht auszudrücken, Risiken

einzuschätzen, Gespür für das Produkt, die Bereitschaft, Nein zu sagen, und Verantwortung für Entscheidungen.

In einem Artikel auf Malorean.net vom März 2026 habe ich diese These vertreten: Je zugänglicher technische Werkzeuge werden, desto mehr verschiebt sich der Wettbewerbsvorteil zum Verständnis des Kontexts – Branchen und Arbeitsweisen, Nutzer, Kunden, technische Einschränkungen, Konkurrenz und Vorschriften. Das ist eine strategische Analyse und kein experimentelles Ergebnis. Sie macht deutlich, was Urteilsvermögen bedeutet: eine Situation gut genug verstehen, um die richtige Sache auszuwählen, statt sie nur umsetzen zu können. [8]

KI macht das „Wie“ leichter zugänglich.

Das „Warum“ wird dadurch nicht automatisch klar.

Sie ermöglicht, eine Lösung zu entwickeln, bevor ein Unternehmen das Problem vollständig verstanden hat. Sie erlaubt, einen Kundenwunsch schnell zu erfüllen, den das Unternehmen eigentlich hätte ablehnen sollen, weil er nicht zu seinem Markt gehört. Sie kann zahllose Varianten einer Funktion erzeugen, ohne dass klar ist, welche ein besseres Ergebnis bewirkt.

Je einfacher das Bauen wird, desto wichtiger wird die Disziplin, bewusst zu entscheiden, was wir **nicht** bauen.

Ramp veranschaulicht das: Der CPO betont, dass Unternehmen ihre wichtigsten idealen Kundenprofile (ICP) und die Probleme definieren müssen, die sie besonders gut

lösen wollen. Sie müssen auch lukrative Wünsche ablehnen können, wenn diese das Produkt in eine andere Richtung ziehen. [5]

Im Juli 2026 bezeichnete Marty Cagan diese Diskrepanz als *KI-Produktivitätsparadox*. Seiner Einschätzung nach löst eine schnellere Umsetzung nicht das Problem von Ideen, die niemals hätten gebaut werden sollen. Er unterscheidet daher zwischen Lösungen, die man entwickelt, um etwas zu lernen, und Produkten, die man ausliefert, um ein Ergebnis zu erzielen. Das ist die Analyse eines erfahrenen Produktpraktikers, keine Messung, die eine einheitliche Wirkung in allen Unternehmen belegt. [6]

Das KI-Paradox lautet also:

Sie erhöht unsere Handlungsfähigkeit erheblich und macht dadurch die Auswahl unserer Handlungen noch wichtiger.

Ein Wandel des Systems, nicht nur der Werkzeuge

Man könnte auf diese Entwicklung reagieren, indem man jedem Entwickler einen Code-Assistenten gibt und die Teams ansonsten unverändert weiterarbeiten lässt.

Das kann örtlich begrenzte Verbesserungen bringen.

Doch je größer diese Vorteile werden, desto deutlicher stößt die Organisation auf andere Grenzen: ihre Fähigkeit zu verstehen, zu entscheiden, zu koordinieren, zu überprüfen und die Ergebnisse der höheren Produktion aufzunehmen.

Dann geht es nicht mehr nur um die Einführung eines Werkzeugs.
Es wird zu einer organisatorischen Frage.
Eine Organisation, die entstand, als Entwicklungskapazität die knappste Ressource war, muss nicht länger optimal sein, sobald diese Knappheit nachlässt.
Damit drängt sich eine Frage auf:
Was geschieht mit einer Organisation, die darauf ausgelegt ist, eine knappe Ressource – die Entwicklungskapazität – zu schützen, wenn diese plötzlich viel reichlicher vorhanden ist?

Literaturverzeichnis

[1] How we built an AI-native engineering system — AMPECO

- URL: https://www.ampeco.com/blog/how-we-built-an-ai-native-engineering-system/
- Erfahrungsbericht über eine um Agenten herum aufgebaute Entwicklungsorganisation, die einen starken Kapazitätszuwachs für sich beansprucht.

[2] Measuring Commercial Impact at Scale — Canva

- URL: https://www.canva.dev/blog/engineering/measuring-commerical-impact-at-scale/
- Beschreibt, wie Canva Produktexperimente mit einer zugänglichen Schätzung ihrer geschäftlichen Wirkung verbindet.

[3] No Silver Bullet: Essence and Accidents of Software Engineering — Fred Brooks

- URL: https://www.cgl.ucsf.edu/Outreach/pc204/NoSilverBullet
- 1986 vorgestellter und 1987 veröffentlichter Aufsatz über wesentliche und technisch bedingte Schwierigkeiten der Softwareentwicklung.

[4] The Mythical Man-Month, Jubiläumsausgabe, Kapitel 16 — Fred Brooks

- URL: https://www.oreilly.com/library/view/mythical-man-month-the/0201835959/ch16.xhtml
- Verlagsübersicht der Ausgabe von 1995, die die spätere Aufnahme von *No Silver Bullet* bestätigt; nur eine Vorschau auf das Kapitel.

[5] 5 insights from Ramp's CPO on building for hypergrowth — Ramp

- URL: https://ramp.com/blog/5-insights-from-ramps-cpo-reddit-ama
- Der CPO von Ramp betont die Konzentration auf priorisierte Kundenprofile und das Ablehnen ablenkender Anforderungen.

[6] The AI Productivity Paradox — Marty Cagan, Silicon Valley Product Group (2026)

- URL: https://www.svpg.com/the-ai-productivity-paradox/

- Analyse vom Juli 2026, die eine höhere Umsetzungsgeschwindigkeit von wirklichen Ergebnissen unterscheidet; eine fachliche Einschätzung, kein Kausalnachweis.

[7] Quatre jours de vibe coding dans la peau d'un PM rouillé — Laurent Deséchalliers, Malorean.net (2026)

- URL: https://malorean.net/articles/2026-07-28-quatre-jours-de-vibe-coding-dans-la-peau-d-un-pm-rouille.de.html
- Persönlicher Erfahrungsbericht mit Claude Code über eine schnelle Produktentwicklung, technische Schwierigkeiten, Tests und Qualitätssicherung; keine kontrollierte Studie.

[8] Dans le logiciel, l'avantage ne sera plus la technologie. Ce sera la compréhension du contexte — Laurent Deséchalliers, Malorean.net (2026)

- URL: https://malorean.net/articles/2026-03-23-dans-le-logiciel-lavantage-ne-sera-plus-la-technologie-ce-sera-la-comprehension-du-contexte.de.html
- Strategischer Essay des Autors über die zunehmende Bedeutung des geschäftlichen, technischen und regulatorischen Kontexts; keine quantitative Untersuchung.

3 — Malorean Corp. — Der Tag danach

— Ich habe KI benutzt.

Lucas sagte es beinahe beiläufig, als wäre damit alles erklärt, was die anderen gerade gesehen hatten.

Er teilte noch immer seinen Bildschirm. Das neue Urlaubsmodul nahm den größten Teil des Fensters ein. Einige Minuten zuvor hatte er gezeigt, wie man vom Smartphone aus Urlaub beantragen konnte und wie Feiertage korrekt in die Berechnung einflossen.

— Moment mal, sagte Alice. Wenn du sagst, du hast KI benutzt, meinst du beim Schreiben des Codes?

— Ja. Na ja, nicht nur. Auch um das Modul zu entwerfen, den vorhandenen Code zu verstehen, die Ansichten zu entwickeln, die Berechnungen und die Tests zu schreiben … Eigentlich für fast alles.

— Und das hast du seit Mittwoch gemacht?

— Seit Mittwochnachmittag. Heute Morgen habe ich die letzten Sachen fertiggestellt.

Alice blickte auf die Gesichter in der Videokonferenz. Manche lächelten. Andere schienen zu überlegen, was sie übersehen hatten.

— Aber du hattest doch gar keine Spezifikation, bemerkte sie.

— Nein. Wir hatten am Mittwoch darüber gesprochen. Ich kannte das Modul und seine Probleme. Zuerst habe ich die KI

gebeten, das vorhandene System zu analysieren. Danach sind wir Schritt für Schritt vorgegangen.

— Und funktioniert alles?

Lucas lachte kurz.

— Alles weiß ich nicht. Das, was ich euch gezeigt habe, funktioniert. Sicher müssen wir noch einiges prüfen.

Thomas, der in demselben Team arbeitete, meldete sich zum ersten Mal.

— Und wie sieht der Code darunter aus?

— Kannst du dir gern ansehen. Ich habe ziemlich viele Tests geschrieben.

— Werde ich tun, antwortete Thomas.

Besonders beeindruckt wirkte er nicht. Vielleicht war er es auch, wollte es aber nicht zeigen.

Hubert Félix Duca hatte bis dahin geschwiegen.

Er hatte genug Vorführungen erlebt, um zu wissen, dass eine Software, die zehn Minuten lang funktioniert, nicht unbedingt bereit ist, von Hunderten Unternehmen genutzt zu werden. Er kannte die Ausnahmen, die besonderen Anforderungen der Großkunden und die Probleme, die erst im echten Einsatz auftreten.

Doch er erinnerte sich auch daran, was am Mittwoch gesagt worden war.

Zwei Monate.

Und jetzt sah er eine Lösung, die bereits erstaunlich dem entsprach, was sie sich vorgestellt hatten.

— Lucas, kannst du mir noch einmal sagen, wie lange du daran gearbeitet hast?
— Ungefähr zwei Tage.
Hubert beugte sich ein wenig zum Bildschirm.
— Gut.
Mehr fiel ihm nicht ein.
Wenige Minuten später endete die Besprechung. Alle kehrten zu ihren Aufgaben zurück, aber niemand sprach wirklich über die Arbeit, die für den Nachmittag geplant gewesen war.

Am Samstagmorgen dachte Alice beim Kaffeekochen erneut an die Vorführung.
Nicht nur Lucas' Geschwindigkeit beschäftigte sie. Vor allem hatte er niemanden gebraucht.
Sie kannte ihren Beruf als Product Manager gut. Bevor sie eine Entwicklung anstieß, versuchte sie das Problem zu verstehen, verglich Anforderungen, sprach mit Kunden und suchte gemeinsam mit den Entwicklern eine vernünftige Lösung.
Beim Urlaubsmodul hätte sie wahrscheinlich mehrere Gespräche organisiert. Sie hätte geholfen, die Regeln zu klären, Sonderfälle abzuwägen und festzulegen, was zuerst entwickelt werden sollte.
Lucas hatte nichts davon getan.
Er hatte ein Gespräch gehört, die Initiative ergriffen und eine Lösung gebaut.

Einerseits bewunderte sie ihn. Andererseits fragte sie sich, was geschehen würde, wenn alle Entwickler so arbeiteten.
Dabei hatte sich ihr eigener Alltag in den vergangenen Monaten bereits stark verändert.
Anfangs nutzte Alice KI für Zusammenfassungen von Protokollen und die Überarbeitung von Dokumenten. Dann übertrug sie ihr wichtigere Aufgaben. Sie automatisierte einen Teil der Auswertung von Kundenfeedback, die Erstellung von Übersichten, die Gruppierung ähnlicher Wünsche und mehrere Verwaltungsaufgaben, für die sie früher Stunden gebraucht hatte.
Heute konnte sie manchmal in wenigen Minuten abschließen, was zuvor einen großen Teil des Vormittags beansprucht hatte.
Keine dieser Aufgaben vermisste sie.
Ganz im Gegenteil.
Endlich hatte sie mehr Zeit, mit Kunden zu sprechen, die Probleme hinter ihren Wünschen zu verstehen und darüber nachzudenken, was sich wirklich zu bauen lohnte.
KI hatte ihren Beruf nicht weniger interessant gemacht.
Sie hatte ihn von einem Teil dessen befreit, was sie daran hinderte, ihn gut auszuüben.
Warum sollten also ausgerechnet die Entwickler weiterarbeiten wie zuvor?
Alice stellte ihre Tasse ab.
Die Frage schien offensichtlich.
Die Antwort viel weniger.

Auch Hubert hatte einen großen Teil des Wochenendes nachgedacht.
Sein Unternehmen hatte immer mehr Ideen gehabt, als es umsetzen konnte. Manche Funktionen warteten seit Monaten. Von Kunden gewünschte Verbesserungen wurden wieder und wieder verschoben. Dann war da noch die geplante Expansion nach Europa, für die an der Software vieles angepasst werden musste.
Das Hindernis kannte er nur zu gut: Es fehlte die Zeit, alles zu erledigen.
Vielleicht hatte Lucas am Freitag gezeigt, dass diese Grenze nicht mehr unüberwindbar war.
Wenn aus einer auf zwei Monate geschätzten Aufgabe in zwei Tagen eine erste Version entstehen konnte, wie viele andere Vorhaben waren plötzlich denkbar?
Malorean Corp. beschäftigte zehn Entwickler. Hubert stellte sich vor, was geschehen würde, wenn sie alle so schnell vorankämen.
Dann stellte er sich zehn Entwickler vor, die ohne Abstimmung jeweils eigene Funktionen auf den Weg brachten.
Dieses zweite Bild gefiel ihm sehr viel weniger.
Schon am Samstag hatte er Alice anrufen wollen, sich dann aber zurückgehalten.
Am Sonntagabend schrieb er ihr nur:

Hast du morgen früh etwas Zeit? Ich würde gern noch einmal über Freitag sprechen.
Ihre Antwort kam wenige Minuten später.
Ja. Ich auch.

Am Montagmorgen um neun trafen sich Alice und Hubert per Video.
— Und? fragte Hubert.
— Und was?
— Hast du verstanden, was am Freitag passiert ist?
Alice lächelte.
— Technisch gesehen ungefähr. Lucas hat mithilfe von KI sehr viel schneller entwickelt, als wir es für möglich gehalten hätten.
— Und der Rest?
— Genau das ist das Problem.
Hubert nickte.
— Ich habe das ganze Wochenende darüber nachgedacht. Seit Jahren fehlt uns Kapazität. Wir lehnen Wünsche ab, verschieben Projekte und überlegen ständig, was wir nicht machen. Und dann zeigt Lucas in zwei Tagen etwas, für das wir zwei Monate veranschlagt hatten.
— Eine erste Version, betonte Alice. Ob sie bereit für den Produktivbetrieb ist, wissen wir noch nicht.
— Ja, schon. Aber selbst wenn er noch zwei Wochen braucht, ändert das vieles.

Alice konnte ihm nicht widersprechen.

— Was mich stört, fuhr Hubert fort, ist, dass er das alles ohne dich gemacht hat.

Alice antwortete nicht sofort.

— Mich hat es auch gestört, gab sie zu.

— Sollen wir ihn daran erinnern, wie wir arbeiten?

— Daran habe ich gedacht.

— Und?

— Ich glaube, das wäre ein Fehler.

Hubert runzelte die Stirn.

— Wenn er allein entscheidet, was er entwickelt, kann uns das unangenehme Überraschungen bringen.

— Natürlich. Aber schau dir an, was geschehen ist. Am Mittwoch sagen wir, das Urlaubsmodul müsse neu gemacht werden. Normalerweise hätte ich begonnen, Anforderungen zu sammeln. Wir hätten Besprechungen organisiert, das Projekt vorbereitet, Aufgaben aufgeteilt … Wahrscheinlich wären Wochen vergangen, bevor wir irgendetwas gezeigt hätten.

— Das ist nicht unbedingt verschwendete Zeit.

— Nein. Und ich sage nicht, dass wir aufhören sollen, die Bedürfnisse zu verstehen. Aber Lucas hat gezeigt, dass er vorankommen kann, ohne all das abzuwarten.

— Du willst also die Entwickler selbst entscheiden lassen?

— Nein, gerade nicht.

Alice überlegte einen Augenblick.

— Ich glaube, wir verwechseln zwei Dinge: zu entscheiden, **was** wir tun sollten, und zu entscheiden, **wie** wir es tun.

— Das hängt doch zusammen.

— Ja. Aber das bedeutet nicht, dass ich bei jeder Entscheidung dabei sein muss.

Sie setzte sich etwas gerader hin.

— Mein Beruf hat sich in den letzten Monaten unglaublich verändert. Ich nutze KI täglich und habe vieles automatisiert, was ich früher von Hand erledigt habe. Für Dokumente und das Sortieren von Informationen brauche ich viel weniger Zeit.

— Ich weiß. Du hast es mir erzählt.

— Aber vielleicht ist dir nicht klar, wie tiefgreifend das ist. Heute habe ich viel mehr Zeit für das, was wirklich zählt: Kunden verstehen, Probleme erkennen und entscheiden, wo wir unsere Energie einsetzen.

Hubert hörte zu.

— Früher bestand ein Teil meiner Arbeit darin, Projekte durch die Organisation zu bringen. Tickets vorbereiten, Anforderungen präzisieren, Fragen beantworten, Gespräche koordinieren. Was wäre, wenn die Entwickler mich dafür bald viel seltener brauchten?

— Dann hättest du mehr Zeit.

— Genau. Und sie auch.

Sie machte eine Pause.

— Wir akzeptieren, dass KI meinen Beruf verändert. Warum sollten wir wollen, dass sich der Beruf der Entwickler nicht verändert?

Hubert lehnte sich zurück.

— In Ordnung. Aber irgendjemand muss weiterhin entscheiden, was wir bauen.

— Selbstverständlich. Und vor allem, warum wir es bauen.

— Wie stellst du dir das also vor?

Alice zögerte.

Sie hatte keine Präsentation vorbereitet, kein neues Organisationsmodell und keinen Prozess, den sie einführen wollte.

Sie hatte nur eine Ahnung.

— Vielleicht sollten wir etwas ausprobieren.

Hubert wartete.

— Wir könnten Lucas ein Projekt geben. Ein echtes Kundenproblem. Und ihm viel mehr Freiheit lassen, es zu lösen.

— Ohne Spezifikation?

— Ohne ihm die fertige Lösung vorzuschreiben. Wir erklären ihm das Problem, bringen ihn mit den Menschen zusammen, die es kennen, und lassen ihn suchen.

— Und du?

— Ich bleibe ansprechbar. Ich helfe ihm, die Bedürfnisse zu verstehen, stelle seine Ideen infrage und passe auf, dass er

nichts nur deshalb baut, weil er dazu in der Lage ist. Aber ich schreibe ihm nicht jeden Schritt vor.

Hubert dachte nach.

— Dann soll er also einen Teil der Arbeit eines Product Managers übernehmen?

— Vielleicht. Oder wir hören einfach auf, unsere Arbeit immer automatisch untereinander aufzuteilen.

— Und wenn es nicht klappt?

— Dann finden wir das heraus.

Hubert lächelte.

— Als Methode ist das etwas dürftig.

— Weiß ich.

Auch Alice lächelte.

— Aber ich würde es lieber an einem begrenzten Projekt ausprobieren, als drei Monate lang über eine neue Organisation zu diskutieren.

Hubert schwieg einen Moment.

Dann schien ihm etwas einzufallen.

— Wir haben seit Wochen ein Problem bei einigen Kunden aus der Lebensmittelindustrie.

Alice nickte. Sie wusste genau, welches.

Das ERP von Malorean Corp. verwaltete Lagerbestände und Fertigungsprozesse, konnte aber die Chargen der Rohstoffe, die zur Herstellung fertiger Produkte verwendet wurden, nicht zuverlässig nachverfolgen.

Wenn ein Lieferant bei einem Lebensmittelhersteller eine fehlerhafte Rohstoffcharge meldete, musste das Unternehmen schnell herausfinden können, welche Endprodukte mit dieser Charge hergestellt worden waren.

Derzeit mussten die Kunden von Malorean Corp. diese Informationen selbst zusammensuchen.

— Letzte Woche kam wieder eine Meldung, erklärte Hubert. Ein Kunde hat mir erzählt, dass er und sein Qualitätsverantwortlicher einen ganzen Tag gebraucht haben, um die Produkte zu finden, die von einer Lieferantenwarnung betroffen waren.

— Ja. Dem Support sind mehrere solche Fälle bekannt.

— Meinst du, Lucas könnte daran arbeiten?

Alice dachte nach.

Das war heikler als das Urlaubsmodul. Es reichte nicht, ein paar Ansichten zu erstellen. Man musste verstehen, wie in den Produktionsstätten tatsächlich gearbeitet wurde, welche Rohstoffe eingesetzt und wie die fertigen Produkte identifiziert wurden.

Aber gerade das reizte sie.

— Ich glaube schon, dass wir ihm das Problem übertragen könnten.

— Kennt er sich in der Branche genug aus?

— Vermutlich nicht.

Hubert hob die Augenbrauen.

— Sehr beruhigend.

— Ich kenne auch nicht alle Einzelheiten. Aber ich weiß, wen man fragen muss. Wir haben Kunden, die ihre Abläufe genau kennen. Im Support und bei der Einführung arbeiten Menschen, die diese Schwierigkeiten jeden Tag sehen. Lucas kann mit ihnen sprechen.

— Du willst ihn direkt zu den Kunden schicken?

— Warum nicht?

— Normalerweise ist das doch deine Aufgabe.

— Eben.

Hubert war noch nicht überzeugt.

— Und wenn er etwas baut, das am Bedarf vorbeigeht?

— Ich muss ihm helfen, das zu vermeiden. Aber nicht, indem ich ihm fünfzig Seiten Spezifikation schreibe.

— Und wenn er seine Lösung produktiv einsetzen will?

— Dann muss sie wie jede andere Änderung geprüft und freigegeben werden. Mehr Freiheit bedeutet nicht, dass er beliebige Risiken eingehen darf.

Hubert schwieg kurz.

— Weißt du, was mir Sorgen macht? fragte er schließlich. Nicht nur, dass er sich irren könnte. Sondern dass irgendwann keiner mehr weiß, wer wofür zuständig ist.

— Das macht mir auch Sorgen.

— Und trotzdem willst du es versuchen?

— Ja.

Sie überlegte, bevor sie hinzufügte:

— Denn die Alternative wäre, genauso weiterzumachen wie bisher und die Entwickler einfach zu bitten, schneller zu programmieren. Ich glaube nicht, dass Lucas uns am Freitag nur das gezeigt hat.

Sie diskutierten noch etwa zwanzig Minuten.
Hubert wollte die laufenden Projekte nicht stören. Alice wollte aus dem Experiment keine große unternehmensweite Initiative machen, bevor es überhaupt begonnen hatte.
Also beschlossen sie, es einfach zu halten.
Lucas würde das Problem der Chargenrückverfolgbarkeit übernehmen. Er dürfte Lösungen erkunden und direkt mit den Menschen sprechen, die ihm helfen konnten, den Bedarf zu verstehen.
Alice bliebe seine Ansprechpartnerin für Produktfragen. Sie würde keinen detaillierten Backlog schreiben, wäre aber weiterhin an wichtigen Entscheidungen beteiligt.
Sie würden mit einem begrenzten Umfang beginnen und keinem Kunden einen Liefertermin versprechen.
Vor allem aber wollten sie genau beobachten, was geschah.
— Wir sollten trotzdem festlegen, woran wir erkennen, ob der Versuch gelungen ist, sagte Hubert.
— Ja. Und ich will nicht einfach zählen, wie viele Funktionen er entwickelt hat.
— Was dann?

— Erst einmal, ob er das Problem versteht und eine nützliche Lösung vorschlagen kann. Danach, wie weit er kommt, ohne dass wir seine ganze Arbeit für ihn organisieren müssen.

— Und wie viel Zeit das uns kostet.

— Auch das.

Hubert sah auf die Uhr.

— Gut. Versuchen wir es.

Alice lächelte, ohne sich ganz sicher zu fühlen.

— Versuchen wir es.

— Sprichst du heute mit ihm?

— Ja.

Die Videokonferenz endete.

Alice blieb einen Augenblick vor dem Bildschirm sitzen. Sie öffnete den Messenger und suchte Lucas.

Dann schrieb sie:

Hallo Lucas. Hast du heute etwas Zeit? Ich würde dir gern ein Projekt vorschlagen, das etwas anders ist als unsere üblichen Aufgaben.

Sie las die Nachricht noch einmal, bevor sie sie abschickte.

Zum ersten Mal seit Langem würde sie einem Entwickler ein Problem anvertrauen, ohne zuvor selbst schon die Lösung im Einzelnen entworfen zu haben.

Sie wusste noch nicht, ob das eine gute Idee war.

Und sie war sich nicht mehr sicher, dass es zu ihrer Aufgabe gehörte, es vor ihm zu wissen.

4 — Neue Organisationsmodelle entstehen

KI ermöglicht es, sehr viel mehr umzusetzen. Aber wie organisieren Unternehmen daraufhin Teams und Entscheidungen? Ihre Antworten unterscheiden sich.
Manche stärken das Product Management im Umfeld der Entwickler. Andere erweitern die Rolle der Entwickler, geben nichttechnischen Fachbereichen neue Möglichkeiten oder verteilen die Arbeit auf kleine, von Agenten unterstützte Teams.

Die herkömmliche Arbeitsteilung ist nicht länger selbstverständlich.

Dabei hatte das traditionelle Modell seinen Sinn: Im Product Management lag ein großer Teil des Wissens über Kunden und Geschäft. Das Design gestaltete das Nutzererlebnis, und die Softwareentwicklung setzte die Absichten in Software um. Tickets, Spezifikationen, Übergaben und Besprechungen hielten diese Berufe bei begrenzter Entwicklungskapazität aufeinander abgestimmt.
Das macht die bisherige Organisation nicht plötzlich falsch. Doch mit steigender Umsetzungskapazität beginnen einige Unternehmen, Menschen, Kontext und Arbeit anders zu verteilen.
Es geht noch nicht darum, das eine „richtige Modell“ zu finden.

Zunächst müssen wir uns anschauen, welche unterschiedlichen Antworten entstehen.

Das Product Management um die Entwickler herum stärken

Die naheliegendste Reaktion besteht darin, die klassische Trennung zwischen Produkt und Engineering weitgehend zu bewahren, den Entwicklern aber mehr Produktkompetenz zur Seite zu stellen.

Gamma liefert dafür ein interessantes Beispiel. Das Unternehmen berichtet von ungefähr einem Product Manager pro vier Engineers, während als historischer Richtwert häufig ein Verhältnis von etwa eins zu zehn genannt wird. [1]

In diesem Modell wird die wachsende Entwicklungskapazität also durch mehr Product Management begleitet.

Die Idee ist einfach: Wenn mehr gebaut werden kann, sollten auch mehr Kapazitäten zur Verfügung stehen, um diese Arbeit zu verstehen, zu priorisieren und in die richtige Richtung zu lenken.

Entwicklern mehr Verantwortung für das Gesamtprojekt geben

Eine andere Strategie erweitert den Aufgabenbereich der Engineers.

Entwickler müssen nicht mehr unbedingt warten, bis ein Problem vollständig in eine Lösung und anschließend in Tickets übersetzt wurde. Sie können stärker daran beteiligt

sein, Bedürfnisse zu verstehen, Prototypen zu erstellen, lokale Entscheidungen zu treffen und Ergebnisse zu beobachten.
In einer Stellenausschreibung für Product Engineers beschreibt Linear die erwartete Verantwortung: eine Funktion von der Idee bis zum ausgelieferten Produkt begleiten und schon an der Definition des Problems mitarbeiten, ohne ständig vom Product Management abhängig zu sein. [2]
Dieses Modell soll einige Übergaben zwischen Berufen vermeiden und mehr Kontext und Handlungsfähigkeit bei einer Person bündeln.
Es bedeutet nicht, dass jeder Entwickler zum PM wird. Vielmehr übernimmt Engineering bei bestimmten Aufgaben einen größeren Teil des Gesamtprojekts.

Die alten Grenzen zwischen Berufen überschreiten

Diese Entwicklung betrifft nicht nur Entwickler.
In einer Untersuchung über sich verändernde Rollen berichtet Figma, dass sich 64 Prozent der Befragten mindestens zwei Rollen zuordnen und mehr als ein Drittel Verantwortung über mehrere Berufsbereiche hinweg übernehmen. Diese Ergebnisse lassen sich nicht allein durch KI erklären. Sie zeigen jedoch, dass die Grenzen zwischen Rollen durchlässiger werden. [3]
Atlassian beschreibt Arbeitsabläufe, in denen Designer direkt am echten Produktcode arbeiten und vor der technischen Review selbst einen Pull Request erstellen. [4]

Auch Cursor berichtet von einer Organisation, in der Produkt, Design und Engineering näher an der tatsächlichen Umsetzung zusammenarbeiten. [5]

Die Erkenntnis lautet also weniger „alle machen alles“, sondern: **Immer mehr Menschen arbeiten auch außerhalb ihres traditionellen Zuständigkeitsbereichs.**

Das wirft eine weitere Frage auf: Wenn mehrere Berufe im selben Bereich handeln können, wie bleiben die Verantwortlichkeiten klar?

Darauf kommen wir zurück.

Kleinen verantwortlichen Teams mehr Arbeit übertragen

Eine weitere Antwort besteht darin, mehr Verantwortung bei weniger Menschen zu bündeln.

AMPECO beschreibt kurze Abstimmungen, sogenannte *Alignment Syncs*, bei denen ein Engineer und ein Product Owner die Absicht klären, bevor Agenten einen großen Teil von Analyse, Entwicklung und Tests übernehmen. [6]

Taito beschreibt ein Vorgehen, bei dem das Team einen Plan bestätigt und anschließend eine oder zwei Personen dessen Umsetzung mithilfe von Agenten steuern. [7]

In solchen Modellen kann eine kleine menschliche Einheit weit mehr Arbeit in Bewegung setzen als zuvor.

Beobachtbar sind damit möglicherweise weniger Übergaben zwischen den Berufen und eine Bündelung des Projekts bei einer Person oder einer kleinen Gruppe.

Offen bleibt, welche Entscheidungen sie allein treffen dürfen und wann sie einen fachlich Verantwortlichen hinzuziehen müssen.

Einen Teil der Prüfungen automatisieren

Wenn schneller produziert wird, geraten auch die bisherigen Freigabeverfahren unter Druck.

Manche Unternehmen reagieren, indem sie mehr Prüfungen automatisieren oder Änderungen abhängig vom Risiko unterschiedlich behandeln.

Meta nutzt einen *Diff Risk Score*, um abzuschätzen, wie wahrscheinlich eine Störung durch eine Änderung ist. [8]

Cloudflare legt Engineering-Standards fest, die vor dem Merge automatisch überprüft werden können. [9]

Taito kombiniert darüber hinaus eindeutige Regeln, Review-Agenten und risikobasierte Auslöser. [7]

Wichtig ist an dieser Stelle lediglich: **Nicht jede Änderung muss auf dieselbe Weise geprüft werden.**

Wie sich Automatisierung, menschliche Verantwortung und Eskalationsschwellen verbinden lassen, behandeln wir später.

Kontext bereitstellen, statt jeden Schritt vorzuschreiben

Andere Unternehmen gehen noch weiter und stellen das ausführliche Ticket als wichtigsten Mechanismus zur Koordination infrage.

Linear kritisiert ausdrücklich ein Modell, in dem der PM den Umfang eines Problems festlegt, es im Ticketsystem dokumentiert und dann darauf wartet, dass Engineering die

Aufgabe erledigt. Das Unternehmen betont, dass Rückmeldungen von Kunden, Ideen, Strategie, Entscheidungen und Code sowohl für Menschen als auch für Agenten zugänglich sein müssen. [10]

Cursor unterstreicht ebenfalls, wie wichtig ausreichend klare Ziele sind, damit Menschen vor Ort mehr eigene Initiative entwickeln können. [5]

Beobachtet wird daher nicht das Ende jeder Abstimmung. Ein Teil der Koordination verlagert sich von detaillierten Anweisungen hin zu **einem reichhaltigeren Kontext, der unmittelbar zugänglich ist**.

Was dieser Kontext enthalten sollte, wie er gepflegt wird und wo die Grenzen der Autonomie liegen, betrachten wir in einem eigenen Kapitel.

Die Menschen bauen lassen, die den Bedarf kennen

Manche Unternehmen ermöglichen Fachbereichen sogar, selbst kleinere Werkzeuge zu erstellen, ohne jede Idee durch die Warteschlange der Entwickler schicken zu müssen.

Shopify hat mit Quick eine interne Plattform geschaffen, auf der mehr Mitarbeiter schnell Websites und kleine Werkzeuge veröffentlichen können. [11]

Vercel beschreibt, wie Mitarbeiter ohne technischen Hintergrund interne Werkzeuge in einer abgesicherten Umgebung bauen können, ohne Entwickler von den Prioritäten des Kernprodukts abzuziehen. [12]

Auch das Beispiel der Designer, die bei Atlassian direkt Code bearbeiten und ihn anschließend technisch prüfen lassen, gehört dazu. [4]

In diesen Fällen beschleunigt KI nicht nur die Entwickler. Sie ermöglicht es bisweilen, die Umsetzung **der Person zu überlassen, die den Kontext des Bedarfs bereits kennt**.

Ein Grundgedanke, der älter ist als KI

Bereits 2018 vertrat Marty Cagan das Konzept der *Empowered Product Teams*: interdisziplinäre Teams, denen die Unternehmensführung Probleme zur Lösung anvertraut, statt eine Liste abzuarbeitender Funktionen vorzugeben. Diese Teams sollen mithilfe der Kompetenzen aus Produkt, Design und Engineering selbst Lösungen finden und für Ergebnisse statt für die produzierte Menge verantwortlich sein. [13]

Gemeinsam mit Chris Jones vertiefte er diesen Ansatz 2020 im Buch *EMPOWERED*. Das Modell kündigte weder austauschbare Berufe noch das Ende der Manager an. Im Gegenteil: Es setzte eine gemeinsame Vision, hohe Fachkompetenz und Führungskräfte voraus, die ihre Teams unterstützen. [14]

Die Idee gab es also schon vor den Agenten. KI könnte nun ermöglichen, dass autonome Teams bestimmte Lösungsansätze wesentlich schneller und direkter erproben. **Die Autonomie von Produktteams ist ein älteres Ziel; KI verändert die Mittel und die Grenzen ihrer Umsetzung.**

Entscheidungen, die noch nicht in eine Richtung führen

Diese Ansätze sind keine aufeinanderfolgenden Schritte derselben Transformation. Das Product Management zu stärken und Entwicklern mehr Produktverantwortung zu geben, kann sogar widersprüchlich erscheinen. In unterschiedlichen Unternehmen können beide Strategien funktionieren – oder innerhalb desselben Unternehmens nebeneinander bestehen. Ihr gemeinsamer Kern liegt woanders: **Wenn die Umsetzung leichter zugänglich wird, müssen wir neu entscheiden, wie Kontext, Entscheidungen und Verantwortlichkeiten verteilt werden.**

Bleibt zu sehen, was das im Alltag der Menschen bewirkt, jenseits von Ankündigungen und Organigrammen.

Wir kehren bald zu Malorean Corp. zurück und beobachten, was diese Modelle im Leben eines kleinen französischen Unternehmens verändern. Danach befassen wir uns mit den Verantwortlichkeiten und Regeln, die eine solche Autonomie braucht.

Literaturverzeichnis

[1] The bottleneck keeps shifting: What AI is changing about how we build — Atlassian / Gamma

- URL: https://www.atlassian.com/blog/how-we-build/the-bottleneck-keeps-shifting-what-ai-is-changing-about-how-we-build

- Beschreibt den höheren Anteil an Product Management bei Gamma, während die Umsetzungskapazität von Engineering steigt.

[2] Product Engineer — Linear Careers

- URL: https://linear.app/careers/0c7c2e26-0a98-42cf-a47c-9a3999fb513b
- Stellenausschreibung über die Verantwortung eines Product Engineers von der Definition eines Problems bis zur Auslieferung.

[3] Are Roles and Responsibilities a Thing of the Past? — Figma

- URL: https://www.figma.com/blog/2025-shifting-roles-report/
- Dokumentiert durchlässigere Berufsrollen und mehr Verantwortung über klassische Fachgrenzen hinweg.

[4] Designers' workflow for shipping code — Atlassian

- URL: https://www.atlassian.com/blog/ai-at-work/designers-workflow-for-shipping-code
- Beschreibt, wie Designer direkt am Produktcode arbeiten, bevor Engineering die Änderung prüft.

[5] How Cursor Is Navigating the New Shape of Product Craft — Atlassian / Cursor

- URL: https://www.atlassian.com/blog/how-we-build/how-cursor-is-navigating-the-new-shape-of-product-craft
- Zeigt eine stärkere Verbindung von Produkt, Design und Engineering mit der Umsetzung.

[6] How we built an AI-native engineering system — AMPECO

- URL: https://www.ampeco.com/blog/how-we-built-an-ai-native-engineering-system/
- Beschreibt eine Organisation, in der wenige Menschen die Absicht klären, bevor Agenten viel Ausführungsarbeit übernehmen.

[7] Inside an AI-native engineering organization — Swarmia / Taito.ai

- URL: https://www.swarmia.com/blog/inside-an-ai-native-engineering-organization/
- Stellt kleine Teams vor, in denen eine oder zwei Personen die agentengestützte Umsetzung leiten.

[8] Diff Risk Score: AI-driven risk-aware software development — Meta

- URL: https://engineering.fb.com/2025/08/06/developer-tools/diff-risk-score-drs-ai-risk-aware-software-development-meta/
- Ein Risikowert, mit dem Prüfprozesse an die wahrscheinliche Kritikalität einer Änderung angepasst werden.

[9] How Cloudflare enforces engineering standards using AI — Cloudflare

- URL: https://blog.cloudflare.com/engineering-standards-enforcement/

- Zeigt, wie ausdrücklich festgelegte Entwicklungsstandards automatisch überprüft werden.

[10] Issue tracking is dead — Linear

- URL: https://linear.app/next
- Befürwortet gemeinsame Kontexte aus Kundenfeedback, Strategie, Entscheidungen und Code über einzelne Tickets hinaus.

[11] Quick: An internal hosting platform for the AI era — Shopify

- URL: https://shopify.engineering/quick
- Stellt eine Plattform vor, auf der Mitarbeiter selbst Werkzeuge entwickeln und veröffentlichen können.

[12] Stopping the slow death of internal tools — Vercel

- URL: https://vercel.com/blog/stopping-the-slow-death-of-internal-tools
- Zeigt, wie Fachbereiche interne Werkzeuge bauen können, ohne die Entwickler vom Kernprodukt abzuziehen.

[13] Empowered Product Teams — Marty Cagan, Silicon Valley Product Group (2018)

- URL: https://www.svpg.com/empowered-product-teams/
- Grundlegender Text über Teams, die Probleme lösen, statt bloß vorgegebene Funktionen abzuliefern.

[14] EMPOWERED: Ordinary People, Extraordinary Products — Marty Cagan und Chris Jones (2020)

- URL: https://www.svpg.com/books/empowered-ordinary-people-extraordinary-products/
- Beschreibung des Buches über autonome Produktteams, gemeinsame Ziele, strategischen Kontext und die Aufgabe ihrer Führungskräfte.

5 — Malorean Corp. — Was wir tun können und was wir tun sollten

Am Dienstagmorgen begann Lucas mit einer Frage, die Alice überraschte.

— Kann ich mit dem Kunden sprechen?

Am Vortag hatte er die wenigen Supportberichte über das Problem der Chargenrückverfolgbarkeit erhalten. Darin war von Produktionsnummern, Rohstoffen, Excel-Dateien und Warnmeldungen der Lieferanten die Rede. Er hatte den Abend damit verbracht, sie mithilfe von KI zu sortieren.

— Natürlich, antwortete Alice. Warum?

— Weil ich nicht genau verstehe, was ihnen fehlt.

Alice lächelte. Eine Woche zuvor hätte sie diesen Termin wahrscheinlich selbst übernommen und ihre Erkenntnisse anschließend an die Entwicklung weitergegeben. Jetzt hatte sie Lucas versprochen, das Projekt selbst zu führen.

— Ich stelle dir die zuständige Person aus dem Support vor. Sie kann das Gespräch organisieren.

Am Mittwoch stellte Lucas fest, dass hinter der Formulierung „Chargen verwalten“ eine deutlich kompliziertere Wirklichkeit steckte.

Der Qualitätsverantwortliche des Kunden zeigte ihnen einen Brief eines Lieferanten. Ein Rohstoff musste ausgesondert werden. Der Lieferant kannte die Chargennummer. Der

Hersteller wusste, welche Lieferungen eingegangen waren. Doch um genau herauszufinden, welche Endprodukte betroffen waren, mussten seine Mitarbeiter mehrere Dokumente miteinander abgleichen.

— Beim letzten Mal, erklärte der Qualitätsverantwortliche, haben wir zu viert fast den ganzen Tag Dateien überprüft. Und selbst danach mussten wir noch einmal in die Produktion, um sicherzugehen.

Lucas wollte wissen, wie die Informationen während der Herstellung erfasst wurden. Der Kunde zeigte ihm eine ERP-Ansicht, danach eine Excel-Tabelle.

— Aber warum tragen Sie die Nummer zusätzlich in Excel ein?

— Weil wir mit Ihrer Software die Informationen nicht finden, die wir brauchen.

Die Antwort war nicht aggressiv. Gerade deshalb war sie schwer zu hören.

Am Ende des Gesprächs hatte Lucas drei Seiten Notizen und weniger Gewissheiten als zu Beginn.

— Und? fragte Alice, als er sie zurückrief.

— Eigentlich wollte ich eine Ansicht bauen, in der man Chargennummern eingibt.

— Und jetzt?

— Jetzt glaube ich, dass wir erst verstehen müssen, wie sie erfassen, welche Rohstoffe tatsächlich verbraucht wurden. Sonst haben wir eine schöne Oberfläche, können ihre Frage aber immer noch nicht beantworten.

Alice nickte. Das hätte sie ihm bereits am Montag sagen können. Doch es gefiel ihr, dass er selbst darauf gekommen war.

Am folgenden Freitag zeigte Lucas ihr eine erste Version mit Testdaten.
Man konnte eine Rohstoffcharge einem Fertigungsvorgang zuordnen und anschließend die daraus entstandenen Endprodukte finden. Er hatte ein kleines Beispiel vorbereitet, in dem sich eine Charge bis zu den von einer Warnmeldung betroffenen Produkten verfolgen ließ.
Alice klickte sich durch die Ansichten.
— Hast du das diese Woche gebaut?
— Ja. Es ist noch vieles falsch oder fehlt. Aber so können wir dem Kunden schon etwas zeigen.
— Und wenn wir auch die Suche in umgekehrter Richtung ermöglichen?
— Ist schon drin.
Er öffnete eine weitere Ansicht.
Alice brach in Gelächter aus.
— Natürlich.
Seit der Vorführung des Urlaubsmoduls hatte sich ihr Maßstab für Projekte verändert. Was sie früher ins nächste Quartal verschoben hätte, schien nun unmittelbar erreichbar.

— Wir könnten auch nach Lieferanten suchen. Und betroffene Aufträge direkt kennzeichnen. Vielleicht sogar Rückrufunterlagen vorbereiten.

Lucas antwortete nicht sofort.

— Wir können diese Ideen aufschreiben, sagte er schließlich. Aber zuerst würde ich dem Kunden gern zeigen, was wir bisher haben.

— Wir warten doch nicht zwei Wochen, bis wir ihm eine vollständigere Version präsentieren?

— Die Vorführung ist nicht mein Problem.

— Was dann?

Lucas zeigte jetzt einen viel weniger ansehnlichen Bildschirm: alte Tabellen, unverständliche Feldnamen und mehrere Verknüpfungen zwischen Fertigung und Lager.

— Wir können so viele Oberflächen bauen, wie wir wollen. Das Problem ist, dass unser ERP nie dafür ausgelegt war, die komplette Geschichte der Chargen zu speichern. Es gibt jahrealte Lagerbewegungen, die teilweise von Hand korrigiert wurden. Manche Informationen existieren einfach nicht.

— KI kann uns doch helfen, das in Ordnung zu bringen?

— Sie kann mir helfen, ein Migrationsskript zu schreiben. Aber sie kann keine Informationen erfinden, die wir nie erfasst haben.

Alice schwieg.

— Wenn wir es richtig machen wollen, fuhr Lucas fort, müssen wir einen Teil des Datenmodells ändern und prüfen, welche

Folgen das für andere Module hat. Wir können nicht die Lagerbestände eines Kunden verfälschen, nur um zwei Wochen zu sparen.

— Ich verlange nicht, dass du die Software kaputt machst.

— Weiß ich. Aber seit Freitag, jedes Mal, wenn ich dir etwas zeige, möchtest du drei weitere Funktionen.

Die Bemerkung war nicht vorwurfsvoll. Trotzdem traf sie Alice.

— Stimmt, gab sie zu.

— Ich will auch schnell vorankommen. Das weißt du. Aber nur weil ich in einer Stunde Code generieren kann, werden fünfzehn Jahre ERP-Geschichte nicht plötzlich einfach.

Nach kurzem Zögern fügte er hinzu:

— Thomas hat sich einen Teil des Codes vom Urlaubsmodul angesehen. Er fand zwei Fälle, die wir falsch behandelt hatten. Er hatte recht. Wir haben sie korrigiert, aber es hat mich an etwas erinnert: Wir können sehr viel schneller bauen und trotzdem jemanden brauchen, der die Fallstricke kennt.

Alice erinnerte sich an Thomas, der am Tag der Vorführung einfach nur den Code hatte sehen wollen.

— Was schlägst du vor?

— Wir arbeiten mit fiktiven Daten weiter und prüfen, ob wir den Bedarf richtig verstanden haben. Aber bevor wir echte Lagerbestände verändern, möchte ich das bestehende System gründlich verstehen und die nötigen Prüfungen durchführen.

Alice sah die erste Version an, die noch auf ihrem zweiten Bildschirm geöffnet war.

In der Vorwoche hatte sie Lucas' Autonomie gegenüber Hubert verteidigt. Plötzlich gefiel ihr weniger, dass zu dieser Autonomie auch gehörte, ihr Nein sagen zu dürfen.

— Einverstanden, sagte sie schließlich. Wir rühren die echten Daten vorerst nicht an.

Am nächsten Montag begann Hubert die wöchentliche Besprechung mit einer inzwischen vertrauten Frage.

— Und? Wie läuft unser Experiment?

— Es geht voran, antwortete Alice. In manchen Punkten schneller als gedacht, in anderen langsamer.

— Zum Beispiel?

Alice zeigte Lucas' Vorführung und erzählte von ihrem Gespräch am Freitag.

Hubert blieb an den technischen Bedenken hängen.

— Es ist gut, dass er vorsichtig ist. Ich möchte kein Problem der Rückverfolgbarkeit lösen und dabei bei allen Kunden neue Lagerprobleme erzeugen.

— Ja. Ich glaube, ich habe etwas zu viel Druck gemacht.

— Du?

— Seit zwei Wochen habe ich das Gefühl, alles sei plötzlich möglich. Sobald jemand eine Idee hat, denke ich, wir könnten sie sofort ausprobieren.

Hubert lächelte.

— So ging es mir nach der ersten Vorführung.

— Man verwechselt leicht die Geschwindigkeit, mit der etwas auf dem Bildschirm funktioniert, mit der Zeit, die eine zuverlässige Lösung braucht.
— Immerhin weiß Lucas, wann er langsamer machen muss.
— Ja. Und ich muss seinem Urteil vertrauen.

Zwei Tage später fragte Julie Morel, ob Alice fünf Minuten Zeit habe.
Alice kannte diese Frage. Bei Julie wurden aus fünf Minuten leicht dreißig.
— Ich habe letzte Woche einen ziemlich großen Interessenten getroffen, begann Julie. Sie haben mehrere Werke und suchen ein ERP, das sich an ihre Arbeitsweise anpassen lässt. Schau mal, was ich vorbereitet habe.
Sie teilte eine Reihe sorgfältig ausgearbeiteter Entwürfe. Darin gab es unterschiedliche Freigabeschritte je nach Art der Bestellung, eigene Regeln für jedes Werk und Ansichten, die sich je nach Nutzer veränderten.
— Hast du das selbst gemacht? fragte Alice.
— Mit ChatGPT. Ich habe meine Gesprächsnotizen genommen, einen Ablauf entwerfen lassen und anschließend alles gemeinsam mit der KI korrigiert. Einen Abend hat mich das gekostet.
Alice sah sofort, wie nützlich diese Bilder waren. In wenigen Stunden hatte Julie einem Kundenwunsch eine konkrete Form

gegeben, der bisher nur in einem Vertriebsbericht beschrieben war.

— Hat der Interessent die Entwürfe schon gesehen?

— Noch nicht. Ich wollte sie dir erst zeigen. Aber er hat ziemlich klar gesagt, dass es für die Verhandlungen viel ändern könnte, wenn wir so etwas anbieten.

Alice blätterte durch die Ansichten.

— Interessant. Du solltest direkt mit Lucas sprechen.

Julie schaute sie überrascht an.

— Direkt? Ich dachte, du bereitest das Projekt vor.

— Warum? Du kennst den Kunden, hast mit ihm gesprochen und sogar die Entwürfe erstellt. Du kannst Lucas viel besser als ich erklären, wonach er sucht.

— Und Lucas kann sagen, ob es machbar ist?

— Ja. Aber Julie, bitte denke daran: Machbarkeit zu prüfen bedeutet nicht, dass wir die Entwicklung zusagen.

— Natürlich.

Alice kannte dieses „natürlich“. Es war keine Lüge, sondern die Begeisterung einer Vertriebsmitarbeiterin angesichts eines möglichen Abschlusses.

Am folgenden Tag trafen sie sich mit Lucas. Julie zeigte ihre Entwürfe, erklärte die Anforderungen und beantwortete die Fragen.

Lucas interessierte sich sofort dafür, wie stark sich die Regeln zwischen den Werken unterschieden.

— Etwas Ähnliches haben wir bereits im ERP, sagte er. Nicht genau dafür, aber der Mechanismus existiert. Für einen Versuch könnten wir wahrscheinlich etwas vorbereiten.

— Wie lange würde das dauern? fragte Julie.

— Für eine Vorführung? Vermutlich nicht besonders lange.

Julie sah zu Alice. Ihr Lächeln wirkte fast, als hätte sie den Auftrag bereits gewonnen.

— Wenn wir das nächste Woche zeigen, könnte das wirklich den Unterschied machen.

Alice spürte selbst den Drang, sofort loszulegen. Dann dachte sie an die Rückverfolgbarkeit, die offenen Anforderungen, die Sonderregeln dieses Interessenten und die vielen anderen Kunden, für die diese Regeln womöglich nutzlos waren.

— Nein, sagte sie.

Die beiden sahen sie an.

— Nein wozu? fragte Julie.

— Nein dazu, diese Funktion jetzt fest in unser Produkt einzubauen, nur um den Auftrag zu gewinnen.

— Aber du hast eben noch gesagt, dass die Idee interessant ist.

— Das ist sie. Nur wissen wir noch nicht, ob der Kunde eine Funktion braucht, die für alle unsere Kunden sinnvoll wäre, oder einen ganz besonderen Ablauf seiner Werke.

— Wir könnten den Auftrag verlieren.

— Vielleicht.

Julie hatte mit dieser Antwort offenkundig nicht gerechnet.

— Endlich haben wir die Mittel, solche Wünsche schnell zu erfüllen, beharrte sie. Es wäre schade, diese Chance nicht zu nutzen.

— Ich will sie nutzen. Aber wir müssen verstehen, was wir anbieten. Wenn wir jede Sonderanforderung in den Kern des ERP übernehmen, wird die Software irgendwann unwartbar.

— Lucas sagt doch, dass es geht.

— Lucas sagt, dass er eine Vorführung vorbereiten kann. Das ist nicht dasselbe.

Es wurde still. Julie betrachtete noch einmal die Entwürfe.

— Was soll ich ihm dann sagen?

— Dass wir sein Problem verstanden haben und ihm einen möglichen Weg zeigen wollen. Nicht, dass wir bereits entschieden haben, die Funktion in unser Produkt aufzunehmen.

Julie nickte. Sie wirkte enttäuscht, aber nicht entmutigt.

— Gut. Ich kann einen Termin vereinbaren, bei dem er uns genauer erklärt, worin sich die Abläufe der Werke unterscheiden.

— Perfekt. Und behalte die Entwürfe. Sie helfen uns, die richtigen Fragen zu stellen.

Als Julie die Videokonferenz verließ, blieben Alice und Lucas noch kurz verbunden.

— Hättest du das wirklich schnell bauen können?

— Eine erste Version, ja.

— Und hättest du Lust dazu gehabt?

— Ehrlich? Ja. Es hätte Spaß gemacht.

Alice lachte.

— Ich hätte auch gern Ja gesagt.

— Was hat dich abgehalten?

— Dass wir beide nur noch darüber gesprochen haben, was wir tun **können**, und fast gar nicht mehr darüber, was wir tun **sollten**.

In der folgenden Woche schickte Lucas Alice einen Link mit einer Nachricht.

Ich habe gestern etwas ausprobiert. Kannst du es dir ansehen?

Alice öffnete die Seite. Sie erkannte die Oberfläche ihres ERP, doch daneben gab es ein kleines Bedienfeld, über das sich Regeln verändern ließen. Sie konnte ein Werk auswählen, einen Freigabeschwellenwert ändern, einen Schritt ein- oder ausblenden und die Vorführung neu starten.

Sie rief Lucas an.

— Was ist das?

— Ein kleines Werkzeug, um Vorführungen mit unterschiedlichen Regeln vorzubereiten. Ich habe unseren alten Konfigurationsmechanismus wiederverwendet. Es läuft nur in einer Testumgebung.

— Wie lange hast du gebraucht?

— Vielleicht zwei Stunden. Die Idee kam mir, als ich noch einmal an Julies Entwürfe dachte.

— Aber das gehört doch überhaupt nicht zu deinem aktuellen Projekt!

— Nein. Ich wollte sehen, ob wir nicht darauf verzichten können, bei jeder neuen Variante den Code zu ändern.

Alice begann, Parameter anzupassen. Der Bildschirm reagierte. Sie probierte eine andere Einstellung, dann eine dritte.

— Moment. Du meinst, ich kann jetzt selbst verschiedene Varianten einer Vorführung zusammenstellen?

— Innerhalb der freigegebenen Regeln, ja. Nicht alles. Und auf keinen Fall in den Kundensystemen.

— Aber ich kann Julie zwei unterschiedliche Abläufe zeigen und sie fragen, welcher wirklich zu den Bedürfnissen ihres Interessenten passt.

— Genau.

Alice betrachtete die Seite eine Weile. Das erste Urlaubsmodul hatte Lucas schneller gemacht. Dieses Werkzeug verschaffte **ihr** eine Fähigkeit, die sie gestern noch nicht gehabt hatte.

Sie konnte eine Idee ausprobieren, ohne eigens einen Entwickler bitten zu müssen, sie umzusetzen.

— Kannst du dir vorstellen, was wir damit alles machen könnten?

Lucas lächelte.

— Ich glaube schon. Aber bevor du mich bittest, ein vollständiges Produkt daraus zu machen: Das ist ein Provisorium, an dem ich zwei Stunden gesessen habe.

— Versprochen.

Nach dem Gespräch spielte sie weiter mit den Einstellungen.

Am nächsten Montag bemerkte Hubert, dass Alice das übliche Projektstatusdokument nicht geöffnet hatte.

— Hast du heute nichts vorbereitet?

— Doch. Aber ich will dir etwas zeigen.

Sie stellte ihm Lucas' Werkzeug vor und baute vor seinen Augen zwei Varianten derselben Vorführung.

Hubert begriff ziemlich schnell.

— Du hast die Entwickler nicht um diese Änderungen bitten müssen?

— Nein. Lucas hat ein kleines Werkzeug geschaffen. Damit kann ich bestimmte Regeln selbst ausprobieren.

— Können wir das dem Vertrieb geben?

— Vielleicht später. Noch ist es nicht robust genug. Aber Julie kann damit schon gemeinsam mit mir die Fälle ihres Interessenten testen.

Hubert schaute interessiert auf den Bildschirm.

— Und das Chargenprojekt?

— Lucas macht Fortschritte. Er hat seine erste Version wieder mit dem Kunden besprochen. Wir verstehen besser, was nötig

ist. Aber er will die historischen Daten nicht anfassen, bevor wir ihren Umgang damit abgesichert haben.

— Klingt vernünftig.

— Ja. Am Freitag hat es mich geärgert. Heute bin ich froh, dass er darauf bestanden hat.

Alice drehte ihren Stift zwischen den Fingern.

— Etwas Merkwürdiges passiert, Hubert. Lucas spricht mit Kunden und trifft Entscheidungen über die Lösung. Julie kommt mit fast benutzbaren Entwürfen. Und ich kann Varianten unseres ERP testen, ohne eine Entwicklung in Auftrag zu geben.

— Macht dir das Angst?

— Manchmal. Ich frage mich, was aus unseren Berufen wird, wenn jeder anfängt, einen Teil der Arbeit anderer zu übernehmen.

— Hast du eine Antwort gefunden?

— Nicht wirklich.

Sie dachte nach.

— Ich weiß nur, dass ich nicht die Person sein will, durch die jede Idee hindurchmuss. Julie musste nicht darauf warten, dass ich aus ihren Entwürfen Tickets mache. Lucas muss von mir nicht erfahren, wie er jede einzelne Ansicht bauen soll.

— Aber du hast abgelehnt, den Wunsch des Interessenten als Funktion zu entwickeln.

— Ja. Weil das eine Entscheidung über unser Produkt war. Ich könnte mich irren und meine Meinung noch ändern. Aber ich musste diese Entscheidung verantworten.

— Und Lucas hat die Abkürzung bei den Lagerbeständen abgelehnt.

— Weil er weiß, was wir damit kaputt machen könnten. Da hat er ein Urteil, auf das ich hören muss, selbst wenn ich ungeduldig bin.

Hubert schaute eine Weile auf ihr kleines Videofenster.

— Wir können die Menschen also mehr Dinge tun lassen, aber wir dürfen nicht zulassen, dass Entscheidungen ohne Verantwortlichen zwischen ihnen hängen bleiben.

— Genau das beginne ich zu glauben.

Sie zögerte und fügte hinzu:

— Früher hatte ich manchmal das Gefühl, meine Arbeit sei wichtig, weil so viele Projekte über meinen Schreibtisch gingen. Jetzt frage ich mich, ob nicht das Ergebnis viel wichtiger ist. Lösen wir das richtige Problem? Bauen wir etwas Zuverlässiges? Lohnt es sich überhaupt?

Hubert nickte.

— Und wenn niemand weiß, wer diese Fragen beantworten muss, haben wir ein Problem.

— Ja.

Sie schwiegen einige Sekunden. Keiner wollte aus dem Gespräch die Ankündigung einer neuen Organisation machen.

Ihr Experiment hatte gerade erst begonnen. Es eröffnete neue Möglichkeiten, brachte aber auch zusätzliche Entscheidungen mit sich.

Alice erhielt eine Benachrichtigung. Julie schrieb, der Interessent sei bereit, die unterschiedlichen Vorführungen anzusehen. Er wollte auch wissen, ab wann seine Teams sie nutzen könnten.

Alice antwortete nicht sofort.

Die Software machte es inzwischen möglich, Antworten viel schneller zu **bauen**.

Doch jemand musste weiterhin entscheiden, welche Antworten Malorean Corp. wirklich **versprechen** und **verantworten** konnte.

6 — Die Rollen verschwimmen, die Verantwortung bleibt

Lange Zeit boten die Grenzen zwischen Product Management, Design und Engineering eine einfache Orientierung.
Das Product Management verstand das Problem und entschied, was gebaut werden sollte. Das Design gestaltete das Nutzererlebnis. Engineering setzte die Absicht in Software um. In Wirklichkeit war die Trennung nie so eindeutig. Trotzdem prägte sie Teams, Einstellungen, Prozesse und Verantwortlichkeiten.
Schon vor KI wurde diese Aufteilung infrage gestellt: Die autonomen Produktteams aus Kapitel 1.4 brachten Produkt, Design und Engineering zusammen, um ein gemeinsames Problem zu lösen. Nun beschleunigt KI das Überschreiten einer weiteren Grenze, nämlich der praktischen Umsetzung. Jeder kann mehr Aufgaben übernehmen, die früher einem bestimmten Beruf vorbehalten waren.
Der Überblick über verschiedene Organisationen hat diese Entwicklung gezeigt, besonders anhand der Figma-Studie. [1] Auf Malorean habe ich außerdem beschrieben, wie sich Entwickler stärker der Problemdefinition und Product Manager stärker der Umsetzung zuwenden. [2]
Man könnte daraus schließen, dass Berufsrollen verschwinden werden. Ich halte eine andere Schlussfolgerung für wichtiger:

Je durchlässiger die Rollen werden, desto ausdrücklicher müssen die Verantwortlichkeiten festgelegt werden.

Handlungsfähigkeit gehört nicht mehr allein einem Beruf

Was KI zunächst neu verteilt, ist nicht unbedingt die Entscheidungsbefugnis, sondern die Fähigkeit, etwas selbst zu tun.

Nehmen wir Designer, die direkt Code ändern – eine von Atlassian beschriebene Praxis. Ein technisches Team prüft weiterhin die Qualität und die Bedingungen für den Produktivbetrieb. [3] Das ist kein Rückfall in die alte Arbeitsteilung. Es zeigt vielmehr, dass Code ändern zu können und für seine Zuverlässigkeit einzustehen zwei verschiedene Dinge sind.

Dieselbe Unterscheidung findet sich in Linears Stellenausschreibung für Product Engineers. Das Unternehmen erwartet, dass diese Personen ein Projekt vom anfänglichen Problem bis zum ausgelieferten Ergebnis begleiten. [4] Daraus ergibt sich nicht automatisch die Befugnis, sämtliche strategischen Produktentscheidungen zu treffen.

Wir können also Übergaben reduzieren, ohne die Fachkompetenz abzuschaffen, die manche Übergaben notwendig machte. Die entscheidende Frage lautet nicht mehr, wie viele Berufe beteiligt sind, sondern **wer wofür einsteht, wenn mehrere Berufsgruppen im selben Bereich tätig werden**.

Mitwirken zu können bedeutet nicht, für den Fachbereich verantwortlich zu sein

Wir müssen zwei Dinge unterscheiden, die Unternehmen oft verwechseln: das Recht zu handeln und die Verantwortung für einen Bereich.

Ein Vertriebsmitarbeiter kann ein internes Werkzeug entwickeln. Damit wird er nicht für die Sicherheit der Plattform verantwortlich.

Ein PM kann eine Benutzeroberfläche ändern. Damit ist er nicht plötzlich für die Einheitlichkeit des Design Systems zuständig.

Ein Entwickler kann über ein lokales Produktverhalten entscheiden. Dadurch erhält er nicht automatisch die Verantwortung für Strategie, Positionierung oder die gesamte Roadmap.

Diese Trennung gibt es in vielen Teams bereits. Doch KI zwingt uns, sie deutlicher festzuhalten.

In stark spezialisierten Organisationen deckten sich Beruf und Verantwortung oft weitgehend. Wer handeln konnte, gehörte auch zu dem Bereich, für den er zuständig war. Sobald die Handlungsmöglichkeiten auf mehr Menschen verteilt werden, fällt diese Deckung weg.

Dann besteht die Gefahr, Autonomie mit einer Verwässerung der Verantwortung zu verwechseln.

Ähnliches habe ich bereits zum Thema Qualität vertreten: Entwickler können nicht einfach ausliefern und anschließend

behaupten, Fehler seien Sache des QA-Teams, eines Bugdienstes oder des nächsten verfügbaren Entwicklers. Für die technische Qualität der ausgelieferten Arbeit muss weiterhin klar jemand verantwortlich sein. [5]
Durchlässige Rollen ändern diesen Grundsatz nicht. Sie machen ihn wichtiger.

Das Konzept des Garanten

Um dieses Problem zu lösen, verwende ich das Konzept des **Garanten**.
Für jeden wichtigen Bereich bleibt eine Person oder ein Team für dessen Kohärenz und Qualität verantwortlich.
Der Garant erledigt nicht notwendigerweise sämtliche Aufgaben dieses Bereichs selbst.
Er muss auch nicht jede kleinste Entscheidung genehmigen.
An ihn wendet sich die Organisation, wenn eine wichtige Entscheidung seinen Fachbereich berührt, eine Grenze überschritten wird oder etwas schiefläuft.
In einem einfachen Unternehmen lassen sich mindestens drei eindeutige Verantwortungsbereiche benennen:

- **Garant für das Produkt:** Kohärenz des Produkts, grundlegende Entscheidungen, Zielkunden, Wertversprechen, wesentliche Abwägungen und Verbindung zur Strategie.
- **Garant für das Design:** Design System, einheitliches Nutzererlebnis, Barrierefreiheit, neue Gestaltungsmuster und Qualitätsanspruch an die UX.

- **Garant für die Entwicklung:** Architektur, Sicherheit, Performance, Wartbarkeit, technische Qualität und Voraussetzungen für den Produktivbetrieb.

Der Grundsatz ist einfach:

Alle können in mehreren Bereichen mitwirken, aber in jedem Bereich bleibt jemand verantwortlich.

Dieses Modell erlaubt eine durchlässigere Zusammenarbeit, ohne die Verantwortung aufzulösen.

Vertrieb darf bauen – ohne Garant für die Technik zu werden

Nehmen wir ein bewusst einfaches Beispiel.

Ein Vertriebsmitarbeiter erkennt einen Bedarf und erstellt mit einem KI-Werkzeug eine kleine Anwendung oder eine nützliche Änderung. Wenn das Unternehmen dafür eine ausreichend abgesicherte Umgebung bietet, gibt es keinen sachlichen Grund, warum ausschließlich Engineering etwas bauen dürfen sollte.

Das bedeutet aber nicht, dass damit die technische Qualität zur Verantwortung des Vertriebs wird.

Wenn die Änderung Sicherheit, Berechtigungen, sensible Daten oder die Architektur betrifft, muss der Garant für die Entwicklung eine Überprüfung verlangen, Korrekturen fordern oder die Änderung stoppen können.

Der Vertriebsmitarbeiter bleibt für den Bedarf verantwortlich, den er lösen will.

Engineering bleibt Garant für das, was das Unternehmen technisch betreiben und warten möchte.
Dasselbe gilt für Design.
Ein PM kann eine Ansicht aus bekannten Komponenten zusammenstellen. Solange er innerhalb des Design Systems bleibt und bewährte Muster verwendet, würde eine verpflichtende Designfreigabe bei jedem Schritt seine Autonomie unnötig einschränken.
Wenn er dagegen das Design System verlässt, ein neues Muster erfindet oder eine grundlegende und komplexe Interaktion gestaltet, muss der Design-Garant eingreifen können.
Die Grenze lautet dann nicht mehr: „Wer darf was anfassen?“
Sondern: **„Ab welchem Risiko oder welchem Grad an Neuheit muss der Garant des betreffenden Bereichs eingreifen?“**

Projektverantwortliche und Garanten sind nicht dasselbe

Das ist wohl eine der wichtigsten Unterscheidungen in den entstehenden Organisationsmodellen.
Der **Owner** trägt die Verantwortung für ein bestimmtes Projekt. Er muss das Problem verstehen, die Arbeit voranbringen, lokale Entscheidungen treffen, notwendige Abstimmungen organisieren, Ergebnisse prüfen und dafür einstehen, dass das Vorhaben gelingt.
Der **Garant** übernimmt dagegen eine übergreifende Verantwortung.

Er muss nicht das Projekt leiten. Er verantwortet ein bestimmtes Niveau an Kohärenz, Qualität oder Risikobegrenzung innerhalb seines Fachgebiets.

Ein Product Engineer kann also Owner einer Funktion sein, ohne Garant der gesamten Unternehmensarchitektur zu werden.

Ein PM kann Owner einer Markteinführung sein, ohne für sämtliche Designfragen zu garantieren.

Ein Vertriebsmitarbeiter kann Owner eines internen Werkzeugs sein, ohne für die Sicherheit aller internen Anwendungen verantwortlich zu werden.

Diese Unterscheidung klärt die Verantwortung für das Ergebnis, ohne dass ein einziger Mensch in allen Bereichen Experte sein muss.

AMPECO beschreibt eine Variante dieser Entwicklung: Ein Product Owner und ein Engineer stimmen das „Was“ und „Warum“ ab, während Agenten einen großen Teil der Analyse, Entwicklung und Tests übernehmen. Die Durchführung ist verteilt; ein Mensch bleibt für die abschließende Freigabe zuständig. [6]

KI kann also die Zahl der Menschen reduzieren, die eine Aufgabe ausführen müssen, und zugleich den Bedarf erhöhen, eindeutig zu wissen, **wer dafür einsteht**.

Vielleicht sind Berufsbezeichnungen nicht mehr der beste Maßstab

Diese Entwicklung wirft eine größere Frage auf: Sollten wir Unternehmen weiterhin hauptsächlich nach Jobtiteln organisieren?

Die Debatte um den Product Owner ist bezeichnend. Auf Malorean habe ich dafür argumentiert, dass PO in erster Linie eine Funktion ist – ein Mandat für Entscheidungen und Verantwortung – und kein eigenständiger Beruf, der allein durch die Pflege eines Backlogs gerechtfertigt wäre. [7]

Durch KI wird diese Unterscheidung besonders nützlich.

Wenn eine Rolle über die Liste ihrer Aufgaben definiert wird, macht KI diese Definition instabil. Immer mehr Tätigkeiten werden für ganz unterschiedliche Menschen zugänglich.

Wenn wir Verantwortung hingegen darüber definieren, wofür eine Person oder ein Team einstehen muss, ist sie wesentlich beständiger.

Ein PM kann weniger Spezifikationen schreiben und dennoch für die Qualität von Produktentscheidungen verantwortlich bleiben.

Ein Entwickler kann weniger Code von Hand schreiben und dennoch für die technische Qualität einstehen.

Ein Designer muss nicht mehr als Einziger Pixel verschieben, um weiterhin die Einheitlichkeit des Nutzererlebnisses zu verantworten.

Auch bei Cursor, wo die Grenzen zwischen Produkt, Design und Entwicklung weniger starr sind, betont die Produktführung, was zentral bleibt: auswählen, was gebaut werden sollte, und eine Richtung vorgeben, die unnötige Abweichungen verhindert. [8]

Ein Beruf verändert sich also nicht in erster Linie dadurch, dass alle Verantwortlichkeiten verschwinden. Er verliert vielmehr das Monopol auf bestimmte ausführende Tätigkeiten.

Der Garant darf nicht zur neuen Bürokratie werden

Allerdings lauert eine offensichtliche Falle.

Wenn jede Codeänderung eines PM von Engineering genehmigt, jede lokale Produktentscheidung eines Entwicklers vom PM freigegeben und jede Ansicht, die kein Designer entworfen hat, von einem Designgremium geprüft werden muss, haben wir nichts gewonnen.

Wir haben die alten verpflichtenden Übergaben nur anders benannt.

Der Garant darf kein universeller Genehmigungsschritt werden.

Gerade seine Existenz soll mehr Autonomie ermöglichen: Jeder kann innerhalb bekannter Regeln handeln, ohne dauernd eine Erlaubnis einzuholen. Die Organisation weiß, wo die Verantwortung liegt und in welchen Fällen die Garanten erneut einbezogen werden müssen.

Im echten Leben funktioniert das nie perfekt.

Manche Änderungen gelangen in Produktion, die ein Garant so nicht genehmigt hätte. Einige lokale Entscheidungen fallen nicht auf. Teams erlauben sich mitunter mehr Freiheit als vorgesehen. Andere Menschen eskalieren aus Vorsicht zu früh. Das muss kein Scheitern des Modells sein.

Eine wirklich autonome Organisation akzeptiert, dass ein Teil der Entscheidungen ohne vorherige Kontrolle fällt. Sonst ist sie nicht autonom.

Wir müssen also nicht garantieren, dass niemals Abweichungen vorkommen.

Wir müssen festlegen, **welche Abweichungen akzeptabel sind, welche das System erkennen muss und welche das Eingreifen eines Garanten erfordern, bevor sie teuer werden**.

Hier verändert sich die Frage.

Wir haben zwei Verantwortungen unterschieden: die der Person, die ein Projekt leitet, und die des Garanten eines Fachbereichs.

Eine Frage bleibt offen:

Wann sollte der Owner allein entscheiden dürfen und wann muss der Garant eingreifen?

Bei Malorean Corp. beginnt diese Grenze zwischen Initiative und Verantwortung Reibung zu erzeugen. Davon erzählt die nächste Folge, bevor wir die Schutzmechanismen und Eskalationsregeln genauer betrachten.

Literaturverzeichnis

[1] Are Roles and Responsibilities a Thing of the Past? — Figma

- URL: https://www.figma.com/blog/2025-shifting-roles-report/
- Untersucht erweiterte Verantwortungsbereiche und durchlässigere Berufsbilder in modernen Produktteams.

[2] PM, développeurs et IA : les rôles se brouillent, les responsabilités restent — Malorean

- URL: https://malorean.net/articles/2026-08-01-pm-developpeurs-et-ia-les-roles-se-brouillent-les-responsabilites-restent.de.html
- Beschreibt die Position des Autors: KI erweitert die Möglichkeiten, ohne Verantwortung für Produkt, Design und Engineering abzuschaffen.

[3] Designers' workflow for shipping code — Atlassian

- URL: https://www.atlassian.com/blog/ai-at-work/designers-workflow-for-shipping-code
- Zeigt Designer, die den Produktcode direkt ändern, während Engineering die technische Prüfung übernimmt.

[4] Product Engineer — Linear Careers

- URL: https://linear.app/careers/0c7c2e26-0a98-42cf-a47c-9a3999fb513b
- Stellenausschreibung zu den Aufgaben eines Product Engineers von der Problemdefinition bis zum ausgelieferten Ergebnis.

[5] La qualité appartient à ceux qui livrent — Malorean

- URL: https://malorean.net/articles/2026-06-03-la-qualite-appartient-a-ceux-qui-livrent.de.html
- Argumentiert für klare technische Verantwortung auch nach Auslieferung und gegen das Abwälzen der Qualitätssicherung.

[6] How we built an AI-native engineering system — AMPECO

- URL: https://www.ampeco.com/blog/how-we-built-an-ai-native-engineering-system/
- Dokumentiert die breite Delegation der Durchführung an Agenten bei fortbestehender menschlicher Verantwortung.

[7] Le PO n’est pas un métier, c’est une fonction — et l’IA n’y change rien, elle accélère — Malorean

- URL: https://malorean.net/articles/2026-08-03-le-po-nest-pas-un-metier-cest-une-fonction.de.html
- Unterscheidet ein Entscheidungs- und Verantwortungsmandat von einem Berufsbild, das vorwiegend Backlogs verwaltet.

[8] How Cursor Is Navigating the New Shape of Product Craft — Atlassian / Cursor

- URL: https://www.atlassian.com/blog/how-we-build/how-cursor-is-navigating-the-new-shape-of-product-craft
- Beschreibt durchlässigere Rollen mit weiterhin klaren Zielen, Produkturteil und Schutzmechanismen.

7 — Malorean Corp. — Was Alice wusste, ohne es je aufgeschrieben zu haben

Am Donnerstagmorgen rief Lucas Alice zum dritten Mal in dieser Woche an.

— Dauert nur zwei Minuten, versprach er.

Alice hatte gelernt, bei diesem Satz vorsichtig zu sein. Lucas bat nie um lange Besprechungen. Er kam mit einer konkreten Frage, sie klärten sie, und er machte weiter. Gerade deshalb fiel es ihr schwer, solche Unterbrechungen abzulehnen.

— Schieß los.

— Bei der Chargenrückverfolgbarkeit möchte der Kunde auch nach Lieferanten suchen können. Das kann ich in die Demo einbauen. Aber wenn ein Lieferant mehrere Chargen mit derselben internen Nummer geliefert hat, zeigen wir die alle zusammen oder muss er eine auswählen?

Alice dachte kurz nach.

— Wir dürfen auf keinen Fall den Eindruck vermitteln, eine Charge sei eindeutig identifiziert, wenn mehrere passen. Zeig die Ergebnisse getrennt. Mit dem Kunden können wir dann klären, wie er sie unterscheidet.

— Das dachte ich auch. Aber ich war mir nicht sicher.

— Warum rufst du mich dann an?

Lucas lachte verlegen.

— Seit der Geschichte mit Julies Entwürfen frage ich lieber nach, bevor ich etwas entscheide, das nach einer Produktregel aussieht.

In der Vorwoche hatte Alice ihn gebeten, den Wunsch eines Interessenten nicht einfach in eine Standardfunktion zu verwandeln. Jetzt ärgerte sie, dass er daraus gelernt hatte, sie im Zweifelsfall immer zu fragen.

— Schon gut. Aber für diese Entscheidung brauchtest du mich nicht.

— Es beruhigt mich, das von dir zu hören.

Sie beendeten das Gespräch. Sieben Minuten waren vergangen. Alice kehrte zu ihrem Dokument zurück, las den letzten Satz und konnte sich nicht mehr erinnern, was sie hatte hinzufügen wollen.

Zehn Minuten später schrieb Lucas:

Noch eine Frage: Wenn bei älteren Fertigungen keine Chargeninformation vorhanden ist, zeigen wir „unbekannt“ oder blenden wir die Spalte aus?

Alice begann zu tippen, hielt dann aber inne.

Die Frage war anders, doch der Grundgedanke war derselbe: Eine Information, über die das ERP nicht verfügte, durfte niemals als gesichert erscheinen. Sie hätte Lucas diesen Grundsatz erklären können, statt die beiden Fälle getrennt zu entscheiden.

Trotzdem antwortete sie:

Wir zeigen deutlich, dass die Information fehlt. Das Problem nicht verstecken.

Lucas reagierte mit einem Daumen nach oben.

Alice schaute in ihren Kalender. Ein Gespräch mit dem Support war verschoben worden. Am Nachmittag stand ein Kundentermin an. Dazwischen wollte sie Julies Vorschläge für den Interessenten mit mehreren Standorten durchsehen.

Sie schätzte es, dass Lucas direkt mit ihr sprach. Genau das hatte sie sich gewünscht, als sie ihm die Aufgabe übertragen hatte. Aber er kam so schnell voran, dass jede neue Möglichkeit eine weitere Frage aufwarf. Je öfter er sie fragte, desto weniger Zeit blieb ihr für die Dinge, die sie am besten konnte: Kunden verstehen und über die Ausrichtung des Produkts nachdenken.

Auch Lucas verlor Zeit. Manchmal wartete er auf ihre Antwort, bevor er weiterarbeitete.

Ein unangenehmer Gedanke kam ihr: Wenn das so weiterging, würden sie am Ende wieder alle Entscheidungen in Tickets festhalten müssen. Das wäre einfacher zu organisieren. Sie würde spezifizieren, Lucas ausführen, und niemand müsste mehr rätseln, wie weit seine Autonomie reichte.

Genau das hatten sie eigentlich überwinden wollen.

Mittags traf sich Alice per Video mit Bob. Sie hatten sich angewöhnt, ohne Tagesordnung miteinander zu sprechen und Erfahrungen zu vergleichen.

— Wie läuft es mit Lucas? fragte Bob.

— Fast zu gut.

— Das klingt nicht wie ein Kompliment.

— Er kommt sehr schnell voran, sieht vieles, spricht mit Kunden. Aber er ruft mich dauernd wegen Entscheidungen an, die er selbst treffen könnte.

— Und du antwortest?

— Natürlich.

— Dann hat er recht, dich anzurufen.

Alice verdrehte die Augen.

— Danke, Bob.

— Nur halb im Scherz. Wenn ich deine Antwort in fünf Minuten haben kann, statt das Risiko eines Fehlers einzugehen, warum sollte ich darauf verzichten?

— Weil wir so niemals vernünftig arbeiten können.

Bob bat um ein Beispiel. Alice erzählte von den zwei Fragen des Morgens, dann vom Gespräch mit Julie und schließlich von einem weiteren Zweifel Lucas': Sollte man eine Einstellung hinzufügen, die ein Kunde verlangt hatte, obwohl wahrscheinlich nur er sie brauchen würde?

— Merkst du es? sagte Alice. Das sind immer Varianten derselben Frage.

— Und was antwortest du?

— Kommt darauf an.

Bob lächelte.

— Siehst du? Genau so eine Antwort bringt die Leute dazu, dich wieder anzurufen.

Alice wollte widersprechen. Aber er hatte recht.

— Das „kommt darauf an“ ist nicht willkürlich, sagte sie. Ich prüfe, ob es um ein wiederkehrendes Problem geht, ob es unseren wichtigsten Kunden hilft, ob wir das Produkt dadurch für alle komplizierter machen und ob wir die Lösung später warten können …

Sie brach ab.

— Da hast du’s, sagte Bob. Das war schon ein Teil der Regel.

— Es ist keine Regel. Eher eine Denkweise.

— Umso besser. Vielleicht solltest du Lucas genau diese Denkweise vermitteln.

Alice schwieg.

Seit Jahren schrieb sie Tickets. Darin beschrieb sie das erwartete Verhalten, Sonderfälle und Abnahmekriterien. Aber sie schrieb fast nie auf, was ihr half, zwischen mehreren möglichen Verhaltensweisen zu entscheiden. Diese Überlegungen blieben in ihrem Kopf oder verschwanden nach einer Besprechung, sobald die Funktion ausgeliefert war.

— Ich habe wahrscheinlich Hunderte solcher Dinge im Kopf, sagte sie.

— Bestimmt. Ich auch.

— Und wir hatten nie einen Grund, sie aufzuschreiben.

— Gründe schon. Nur keine Zeit. Außerdem hat noch nie jemand ein Ticket mit dem Titel „Erklären, warum wir manchmal eine gute Idee ablehnen“ eröffnet.
Beide lachten.
Nach dem Gespräch öffnete Alice ein neues Dokument und schrieb: *Grundsätze für Produktentscheidungen*.
Sie fügte einen Satz hinzu, löschte ihn und probierte einen anderen. Nach fünfzehn Minuten hatte sie einen Absatz geschrieben, der wie eine interne Richtlinie klang. Er war nicht falsch. Aber er klang überhaupt nicht nach dem, was sie Bob gerade erklärt hatte.
Sie schloss das Dokument.
Dann öffnete sie ChatGPT.

— Hilf mir, das aufzuschreiben, was ich über unser Produkt weiß, diktierte sie. Ich will keine theoretische Vorschrift. Stell mir Fragen, hilf mir, mich an Beispiele zu erinnern, und trenne echte Regeln von meinen persönlichen Gewohnheiten.
Die erste Antwort schlug vor, bei den jüngsten Entscheidungen anzufangen.
Alice begann zu erzählen.
Sie berichtete von Julies Interessenten, den überzeugenden Entwürfen und der Gefahr, eine besondere Anforderung zu einem Versprechen an alle Kunden zu machen. Sie erklärte, dass ein Großkunde viel Umsatz bringen konnte, ohne damit automatisch die richtige Richtung für das Produkt vorzugeben.

— Moment, ich muss dir erklären, warum, sagte sie laut. Unser ERP wird von ganz unterschiedlichen Unternehmen genutzt. Wenn wir jede Sonderregel in den Kern der Software einbauen, machen wir das Leben für alle komplizierter. Aber manchmal deutet eine besondere Anfrage auf ein viel allgemeineres Problem hin. Wir können auch nicht alles ablehnen.
ChatGPT fragte sie, wie sie die Fälle unterschied.
Alice schilderte die Gespräche mit dem Support, Fälle bei mehreren Kunden, Anfragen aus dem Vertrieb und Einschränkungen bei der Einführung. Sie erinnerte sich an alte Diskussionen, von denen kein Protokoll mehr existierte. Sie erläuterte, warum bestimmte Möglichkeiten verworfen worden waren, weshalb sie eine Konfiguration statt einer neuen Funktion bevorzugt hatten und warum eine Ausnahme bei der Lohnabrechnung anders zu behandeln war als eine persönliche Vorliebe bei der Bedienoberfläche.
Manchmal überraschte sie sich selbst mit der Antwort auf eine Frage, die sie noch nie ausdrücklich gestellt hatte.
Die KI formulierte um, gruppierte und schlug einen Grundsatz vor. Alice korrigierte ihn.
— Nein, nicht „wir entwickeln niemals kundenspezifische Funktionen“. Unter bestimmten Bedingungen können wir das tun. Aber wir müssen uns bewusst entscheiden, was uns das kostet. Vor allem darf eine Annahme aus dem Vertrieb nicht als getroffene Produktentscheidung erscheinen.

Später schrieb ChatGPT, eine Rückverfolgbarkeitsfunktion könne garantieren, sämtliche historischen Chargen nachträglich zu rekonstruieren.

— Ganz bestimmt nicht, sagte Alice zum Bildschirm. Daten, die nie gespeichert wurden, können wir nicht herbeizaubern. Das ist eine Grenze, die wir zeigen müssen, keine Leistung, die wir versprechen dürfen.

Sie ließ den Absatz ändern. Das Dokument sollte unterscheiden, was bereits beschlossen war, was sie für wahrscheinlich hielt und was der Support, Lucas oder Kunden erst noch bestätigen mussten. Vertrauliche Informationen aus echten Kundenkonten nahm sie nicht auf und formulierte die Beispiele um.

Eine Stunde verging. Dann zwei.

Das Dokument füllte sich mit Dingen, die Alice seit Langem wusste: welche Kunden Malorean bevorzugt bedienen wollte, wiederkehrende Probleme, akzeptierte Kompromisse, Zusagen, die das Unternehmen nicht machen sollte, Grenzen historischer Daten und die Gründe, warum ein Prototyp noch keine verkaufsfertige Funktion war.

Sie fügte Verweise auf bereits verfügbare Informationen hinzu: Supportmeldungen, auffindbare ältere Entscheidungen, Gesprächsnotizen und Produktdokumentation. Wo ein Nachweis fehlte, musste der Text das ausdrücklich sagen.

Gegen Ende der dritten Stunde schlug ChatGPT eine Zusammenfassung vor.

Alice las sie durch. Manche Seiten waren zu lang. Andere vermischten einen allgemeinen Grundsatz mit einem Beispiel. Sie korrigierte besonders kategorische Formulierungen, ergänzte Datum und Geltungsbereich des Dokuments. Anschließend exportierte sie alles als PDF.
Überrascht blätterte sie durch die zahlreichen Seiten.
Drei Stunden zuvor hatte sie Lucas nur erklären wollen, wann er selbst entscheiden konnte. Nun hielt sie ein unvollkommenes, aber erstaunlich reichhaltiges Bild dessen in Händen, was das Team über sein Produkt wusste, ohne es jemals gemeinsam schriftlich festgehalten zu haben.
Es war keine endgültige Wahrheit. Manche Passagen mussten bestätigt oder später aktualisiert werden. Das PDF ersetzte weder Kunden noch Daten noch das Urteil der verantwortlichen Menschen.
Aber die nächste Person müsste wenigstens nicht von vorn beginnen.

Am nächsten Morgen öffnete Lucas die Datei.
— Hast du das alles für mich geschrieben?
— Nicht allein, antwortete Alice. Ich habe drei Stunden mit ChatGPT gesprochen. Es hat mir geholfen, die richtigen Fragen zu stellen, meine Antworten zu ordnen und Lücken zu erkennen. Danach habe ich alles gelesen und korrigiert. Aber das meiste wusste ich schon.
— Warum habe ich es dann noch nie gesehen?

Alice lächelte etwas verlegen.

— Weil es in kein Ticket passte.

Lucas überflog das Inhaltsverzeichnis. Er fand Grundsätze, Beispiele für Entscheidungen, fachliche Einschränkungen, Informationen über Kunden, Links zu Quellen und offene Fragen.

— Kann ich das meiner KI geben, wenn ich an der Rückverfolgbarkeit arbeite?

— Ja, in unserer dafür zugelassenen Umgebung. Aber sie soll es als Kontext verwenden, nicht als Beweis, dass alles darin stimmt. Bei wichtigen Regeln gehen wir zur Quelle oder zu der Person zurück, die sie verantwortet.

— Das würde mir unglaublich viel Zeit sparen. Manchmal bitte ich die KI, mir zwischen zwei Lösungen zu helfen. Aber sie weiß nur, was ich ihr vorher erzählt habe. Dann schlägt sie etwas vor, das sehr überzeugend klingt, aber eine Entscheidung von vor drei Jahren ignoriert.

Alice kannte das Gefühl. Eine Antwort konnte für sich genommen vernünftig erscheinen und für ihr ERP trotzdem vollkommen unsinnig sein.

Lucas blieb an einer Passage hängen.

— Hier steht, dass wir fehlende Geschäftsinformationen niemals verbergen dürfen, wenn dadurch ein Ergebnis vollständig aussehen könnte, obwohl es das nicht ist.

— Genau das habe ich dir gestern gesagt.

— Ja. Hätte ich den Satz gestern schon gehabt, hätte ich dich nicht angerufen.

Er las weiter und sah dann auf.

— Gut, aber ich habe immer noch ein Problem.

Alice seufzte innerlich.

— Welches?

— Jetzt verstehe ich besser, wie du denkst. Aber ich weiß immer noch nicht, ab wann ich allein entscheiden darf.

Sie stellte den Kaffee ab.

Er hatte gerade die wichtigste Lücke des Dokuments gefunden.

Sie griffen das Thema gemeinsam auf. Alice teilte eine leere Seite.

— Versuchen wir es einfach. Bevor du mich anrufst, stell dir drei Fragen: Habe ich verstanden, welches Ergebnis wir erreichen wollen? Bleibt meine Entscheidung innerhalb der bekannten Regeln? Und was könnte schiefgehen, wenn ich mich irre?

— Dann müssen wir definieren, was „innerhalb der Regeln“ bedeutet.

— Genau.

Sie legten drei Stufen fest.

Grün: Lucas entscheidet selbst, wenn eine Entscheidung lokal, rückgängig zu machen und mit bekannten Regeln vereinbar ist und kein neues Versprechen an Kunden beinhaltet. Er muss

seine Überlegung erklären und das Ergebnis überprüfen können.

Orange: Lucas spricht mit Alice, wenn eine neue Situation entsteht, Ziele miteinander in Konflikt stehen oder erhebliche Zweifel am Produktverhalten bestehen. Geht es um technische Risiken, zieht er auch die zuständigen Fachleute hinzu – etwa Thomas, wenn die Geschichte des ERP betroffen ist.

Rot: Niemand entscheidet allein, wenn eine grundlegende Regel verändert, eine geschäftliche Zusage für Malorean gemacht, mit sensiblen Daten gearbeitet oder ein großes, kaum umkehrbares Risiko eingegangen wird. Dann ist eine ausdrückliche Entscheidung mit den betroffenen Verantwortlichen und manchmal auch Hubert nötig.

Lucas las die Übersicht noch einmal.

— Wenn ich also die Darstellung von Ergebnissen in der Demo ändere, damit sie nicht missverstanden werden, ist das grün.

— Ja.

— Wenn ich ein neues Verhalten einführen will, das alle Kunden betreffen könnte, spreche ich mit dir.

— Genau, besonders wenn wir den Bedarf noch nicht richtig verstehen.

— Und wenn ich historische Daten ändern will, um Chargen zu erfassen?

— Das darf ich nicht allein freigeben. Wir brauchen eine technische Analyse und die passenden Prüfungen. Du hast mich oft genug daran erinnert.

Lucas grinste.

— Ich wollte nur sichergehen, dass du dich noch daran erinnerst.

Sie ergänzten eine Regel: Wenn Alice nicht antwortete, wurde aus einer orangen oder roten Entscheidung niemals automatisch eine grüne. Umgekehrt musste Lucas nicht für jede grüne Entscheidung um Erlaubnis bitten.

— Und wenn ich die falsche Farbe wähle? fragte Lucas.

— Dann sagst du es mir. Wir verbessern die Regeln. Mir ist lieber, wir machen unser System besser, als dir vorzuwerfen, du hättest meine Gedanken nicht erraten.

Lucas nickte.

— Ich könnte sogar meine KI fragen, ob eine Entscheidung zur Übersicht passt, und sie die Begründungen und fehlenden Informationen auflisten lassen.

— Ja. Aber die KI darf nicht an unserer Stelle entscheiden, ob ein Risiko akzeptabel ist.

Alice nahm die Übersicht in das Referenzdokument auf. Sie begann zu verstehen, dass es nicht dasselbe ist, Informationen weiterzugeben und Menschen wirklich zum Entscheiden zu befähigen.

Damit hätten sie aufhören können. Doch Lucas kam auf die Rückverfolgbarkeit zurück.

— Wir sprechen seit einer Stunde über meine Autonomie. Jetzt habe ich eine Frage zum Projekt selbst: Woran erkennen wir, dass wir erfolgreich waren?

Alice antwortete fast automatisch:

— Wenn der Kunde die Produkte finden kann, die mit einer bestimmten Charge hergestellt wurden.

— Ja. Aber wie prüfen wir das? Und wie zuverlässig muss es sein?

Dieses Mal antwortete Alice nicht sofort.

Sie hatten tagelang Oberflächen, Regeln und Datenstrukturen ausprobiert. Die Vorführung sah vielversprechend aus. Aber niemand hatte genau festgehalten, woran das Ergebnis gemessen werden sollte.

Alice erinnerte sich an den Qualitätsverantwortlichen, der nach einer Lieferantenwarnung fast einen ganzen Tag lang Dateien abgeglichen hatte.

— Dort sollten wir anfangen, sagte sie. Wenn er eine Warnung zu einer Chargennummer bekommt, kann er die betroffenen Produktionsvorgänge finden, ohne wieder alles von Hand durchsuchen zu müssen? Wie lange dauert es? Ist das Ergebnis vollständig? Und falls Informationen fehlen, wird das klar angezeigt?

— Wir können es mit ihm testen. Nicht einfach fragen, ob ihm unsere Demo gefällt.

— Genau.

Alice öffnete eine neue Seite.

— Eigentlich hätten wir damit beginnen sollen.

Lucas lachte laut.

— Wenn du mir das vor zwei Wochen vorgeschlagen hättest, hätte ich gesagt, du willst wieder alles in ein Ticket verwandeln.

— Vielleicht. Aber schau, was wir gerade aufschreiben.

Sie hielten das beobachtete Problem fest, die betroffenen Nutzer, das erwartete Ergebnis, die Art der Messung, Grenzen der Daten, den ausgeschlossenen Umfang und Entscheidungen, die einer Freigabe bedurften. Eine Seite, nicht mehr.

Lucas schlug vor, das Dokument *Mini PRD* zu nennen.

— Du sagst mir nicht, welche Oberfläche ich bauen soll, stellte er fest. Wir einigen uns darauf, was die Arbeit nützlich macht.

— Und auf die Grenzen, die wir nicht überschreiten dürfen.

Alice dachte laut nach:

— So sollten wir künftig Projekte beginnen. Bevor wir entscheiden, was wir entwickeln, fragen wir, welches Ergebnis wir erreichen wollen und wie wir es überprüfen. Erst danach grenzen wir das Projekt ein und wählen aus, was sich zu bauen lohnt.

— Und wenn ich dieses Dokument mit den richtigen Informationen ausfüllen kann, muss ich nicht auf einen Backlog von dir warten.

— Du kannst einen Projektzuschnitt vorschlagen. Das heißt nicht, dass wir jede Zusage ungeprüft genehmigen. Aber ja, genau das ist das Ziel.

Sie vervollständigten das Mini PRD für die Rückverfolgbarkeit und ließen die Fragen offen, auf die sie noch keine Antwort hatten. Einige Fälle mussten mit dem Kunden geprüft und bestimmte technische Einschränkungen bestätigt werden.

Das Dokument vermittelte keine falsche Gewissheit. Es zeigte, was sie noch lernen wollten.

Alice fand es nützlicher als eine Liste von Funktionen zum Abhaken.

Am folgenden Montag verlief die wöchentliche Besprechung mit Hubert anders.

Alice zeigte weder ihre übliche Fortschrittstabelle noch die Liste der von Lucas entwickelten Funktionen.

Sie öffnete die Entscheidungsübersicht und anschließend das Mini PRD.

Hubert las die ersten Zeilen.

— Ihr habt so viel Zeit für Dokumente aufgewendet, obwohl Lucas jetzt wesentlich schneller arbeitet?

Alice kannte den Einwand. Wenige Tage zuvor hatte sie selbst so gedacht.

— Ich glaube, genau solche Dokumente haben uns gefehlt. Wir schrieben auf, was gebaut werden sollte, aber selten die Informationen, die uns bei der Auswahl halfen. Also musste

Lucas mich dauernd fragen. Damit kann er einige Entscheidungen ohne mich treffen.

— Und wie weißt du, ob er innerhalb der Grenzen bleibt?

— Wir haben festgelegt, wann er selbst entscheiden darf und wann er uns fragen muss. Ansonsten werden wir nicht alles kontrollieren. Wir prüfen die Ergebnisse und ändern die Regeln, wenn es nötig ist.

Hubert hielt bei einem Satz über historische Daten inne.

— Ihr könnt keine Rückverfolgbarkeit für etwas garantieren, das nie erfasst wurde.

— Nein, sagte Alice. Und das steht auch da. Wir müssen es dem Kunden klar zeigen.

Hubert nickte.

— Das erfahre ich lieber jetzt, bevor jemand aus dem Vertrieb das Gegenteil verspricht.

— Eben. Der Kontext soll nicht nur Lucas helfen. Auch Julie kann darin nachlesen, was wir wissen und was noch nicht entschieden ist. Aber wir müssen die Informationen aktuell halten und kennzeichnen, welche Aussagen verbindlich sind.

Sie beschlossen, eine bearbeitbare Arbeitsfassung zu behalten, von der das PDF nur eine datierte Momentaufnahme darstellte. Wichtige Entscheidungen sollten mit ihrer Begründung erfasst werden. Wiederkehrende Grundsätze könnten bei Bedarf zu Agentenanweisungen oder automatisierten Prüfungen werden. Technische Tests und

notwendige Freigaben vor dem Produktivbetrieb blieben davon unberührt.

— Ich will nicht in sechs Monaten feststellen, dass eine KI eine alte Regel anwendet, hinter der niemand mehr steht, sagte Hubert.

— Ich auch nicht, erwiderte Alice.

Zum ersten Mal schien Hubert vor allem das Mini PRD zu interessieren.

— Und ihr werdet prüfen, ob der Kunde seine Chargen tatsächlich findet?

— Ja, sagte Alice. Wir testen mit ihm konkrete Fälle. Wenn das Ergebnis nicht zuverlässig ist, spielt die Zahl der fertigen Ansichten keine Rolle.

Hubert schloss das Dokument.

— Sehr gut. Ich will wissen, was ihr gelernt habt, nicht nur, was ihr ausgeliefert habt.

Am Mittwoch bemerkte Alice etwas Merkwürdiges in ihrem Messenger.

Seit dem Morgen keine Nachricht von Lucas.

Sie öffnete den Projektkanal. Dort war durchaus etwas passiert: Lucas hatte in der Testumgebung mehrere Arten der Ergebnisdarstellung ausprobiert. Bei einer hatte er erklärt, dass er Chargen mit identischen Referenzen voneinander getrennt darstellte, damit keine Verbindung vorgetäuscht wurde. Er verwies auf einen Grundsatz aus dem

Produktkontext, nannte die verwendeten Daten und schlug vor, die Verständlichkeit der Ansicht mit dem Kunden zu prüfen.
Er hatte nicht um Erlaubnis gefragt.
Alice las die Erklärung und arbeitete weiter. Sie hatte nichts hinzuzufügen.
Am späten Nachmittag rief Lucas an.
— Ich habe etwas Oranges, das vielleicht rot wird, sagte er.
— Ich höre.
— Der Kunde möchte auch in älteren Produktionsvorgängen suchen. Für einen Teil der historischen Daten haben wir aber keine verlässliche Verknüpfung zwischen Rohstoffen und Endprodukten. Ich kann anzeigen, was wir wissen. Doch wenn wir das als vollständige Suche präsentieren, könnte der Kunde glauben, dass keine Produkte betroffen sind, sobald die Suche nichts findet.
Alice richtete sich auf.
— Du hast recht. Diese Mehrdeutigkeit dürfen wir nicht zulassen.
— Ich könnte einen Warnhinweis hinzufügen und Zeiträume unterscheiden, für die die Daten verwendbar sind. Aber ich möchte gemeinsam entscheiden, was wir dem Kunden versprechen können.
— Hier ist das richtig. Wir sollten auch den Support fragen, wie der Kunde diese Grenze verstehen würde.
Lucas teilte seinen Bildschirm. Sie begannen, das Problem zu bearbeiten. Es ging nicht um die Freigabe eines Buttons oder

eine Antwort, die Alice ohnehin schon wusste. Die Entscheidung konnte das Vertrauen eines Kunden in ihre Software beeinflussen.

Während des Gesprächs fiel Alice auf, dass sie kein einziges Mal daran gedacht hatte, wie viel Zeit Lucas sie kostete.

Bevor sie auflegten, zögerte Lucas.

— Weißt du, dein PDF hat nicht alle meine Fragen beantwortet.

— Zum Glück, sagte Alice.

— Zum Glück?

— Wenn ich dort alle Antworten hätte unterbringen können, gäbe es für dich nichts mehr zu entdecken.

Lucas lächelte.

— Gut. Aber ich glaube, ich weiß jetzt besser, wann ich zu dir kommen muss.

Nach dem Gespräch öffnete Alice wieder ihren Kalender. Vor dem nächsten Kundentermin hatte sie noch eine Stunde frei. Eine ganze Stunde.

Sie wusste noch nicht, ob das Experiment in größerem Maßstab funktionieren würde und wie viele Fehler sie machen würden. Aber Lucas hatte mehrere Entscheidungen ohne sie getroffen und sich genau in dem Moment wieder gemeldet, als ihr Urteil wirklich nötig war.

Sie ergänzte im gemeinsamen Dokument eine offene Frage zur Darstellung historischer Daten. Kein Ticket, das Lucas vorschrieb, welchen Code er zu schreiben hatte. Eine

Entscheidung, die sie zusammen mit den Betroffenen klären mussten.

Ihre Arbeitsweise war nicht weniger anspruchsvoll geworden. Sie lernten lediglich, ihre Sorgfalt dort einzusetzen, wo sie wirklich gebraucht wurde.

8 — Mehr Autonomie braucht klarere Regeln, nicht weniger

Ein Unternehmen kann beschließen, dass mehr Menschen handeln dürfen, ohne auf einen Spezialisten zu warten.
Die durchlässigeren Berufsrollen und die zuvor beschriebenen Verantwortlichkeiten schaffen ein neues Koordinationsproblem: Mehrere Menschen können am selben Projekt arbeiten, ohne sich unbedingt auf dieselben Maßstäbe für ihre Entscheidungen zu stützen.
Wenn aber jede neue Handlungsmöglichkeit eine systematische menschliche Genehmigung erfordert, bleibt die Autonomie nur auf dem Papier bestehen.
Die alten verpflichtenden Übergaben werden lediglich von einem Beruf zum nächsten verschoben.
Die eigentliche Herausforderung ist daher, ein System zu schaffen, in dem mehr Entscheidungen vor Ort getroffen werden können, ohne die Kohärenz, Sicherheit, Qualität oder Produktausrichtung zu verlieren.
Hier zeigt sich ein wichtiges Paradox:
Je mehr Autonomie wir verteilen wollen, desto ausdrücklicher müssen die Regeln sein.
Autonomie entsteht nicht durch das Fehlen von Regeln.

Sie entsteht, wenn Menschen handeln können, ohne ständig um Erlaubnis zu bitten, weil Ziele, Grenzen, Risiken und Verantwortlichkeiten ausreichend klar sind.

Wenn alles genehmigt werden muss, gibt es keine Autonomie

Stellen wir uns einen Product Engineer vor, der für ein Projekt verantwortlich ist.

Er spricht mit Nutzern, untersucht Lösungen, entwickelt Prototypen, programmiert und bereitet die Einführung vor.

Auf dem Papier arbeitet er autonom.

Muss er jedoch jede kleine Produktentscheidung vom PM, jedes Detail der Oberfläche vom Designer und jede technische Entscheidung vom Tech Lead genehmigen lassen, hat sich die Verantwortung nicht wirklich verlagert. Stattdessen muss eine einzelne Person noch mehr Freigabestationen durchlaufen.

Dasselbe Problem betrifft andere Berufe: Menschen bauen zu lassen hilft wenig, wenn jede Handlung noch die alten Prozesse und Freigaben durchlaufen muss.

Die Frage ist also nicht nur, **wer handeln darf**.

Entscheidend ist, **innerhalb welcher Grenzen diese Person selbst entscheiden kann**.

Bei Cursor wird dieser Gedanke durch ausreichend klare Ziele und Schutzmechanismen umgesetzt, damit die Menschen, die etwas bauen, handeln können, ohne von der Vision oder Roadmap abzuweichen. [1]

Ein Teil der detaillierten Anweisungen wird durch einen Entscheidungsrahmen ersetzt.

Einen Teil der Tickets durch Entscheidungsregeln ersetzen

An Linear haben wir bereits gesehen, warum ein Ticket, das alles vorschreibt, die Eigeninitiative begrenzen kann. [2] Jetzt wird die Frage praktisch: Welche Informationen braucht ein Team, um selbst entscheiden zu können, ohne auf eine neue Spezifikation zu warten?

Das heißt nicht, Tickets abzuschaffen. Auf Malorean habe ich den Backlog als Werkzeug für die Umsetzung und Abstimmung beschrieben, nicht als vollständiges Gedächtnis des Produkts. [3]

Ein Rahmen für Autonomie kann die Informationen sichtbar machen, die lokale Entscheidungen leiten, ohne jede Lösung im Voraus festzulegen.

Ein solcher Rahmen könnte enthalten:

- den wichtigsten Kunden oder das priorisierte Ideal Customer Profile (ICP);
- das zu lösende Problem;
- das erwartete Ergebnis;
- die Grenzen des Projektumfangs;
- wichtige Einschränkungen;
- das erforderliche Qualitätsniveau;
- Dinge, auf die das Unternehmen ausdrücklich verzichtet;

- Schwellen, ab denen eine Entscheidung eskaliert werden muss.

Ramp veranschaulicht das: Wenn zentrale Kundenprofile, die wichtigsten Probleme und die eigenen Stärken klar sind, kann ein Unternehmen sogar lukrative Möglichkeiten ablehnen, die sein Produkt in eine andere Richtung ziehen würden. [4]
Figma erinnert daran, dass ein schneller Weg von null zum ersten Prototyp nicht genügt. Reale Anwendungsfälle, Sonderfälle und Qualität bleiben entscheidend, damit aus einem schnellen Prototyp ein gutes Produkt wird. [5]
Diese Übersicht beschreibt eine mögliche Organisationsentscheidung: nicht nur **festlegen, was zu tun ist**, sondern auch die Grenzen sichtbar machen, die eingehalten werden müssen. Sie ist kein allgemeingültiges Verfahren.

Kontext wird zur Infrastruktur

Wer vor Ort gute Entscheidungen treffen soll, braucht Zugang zu Informationen, mit denen das Problem verstanden werden kann.
Das ist vermutlich eine der wichtigsten organisatorischen Veränderungen im Zeitalter der Agenten.
Ein Unternehmen kann nicht mehr Menschen Entscheidungen übertragen und gleichzeitig fast den gesamten Kontext in den Köpfen weniger Spezialisten belassen. Linear betont, dass Kundenfeedback, strategische Ausrichtung, Entscheidungen

und Code sowohl für Menschen als auch für Agenten zugänglich sein müssen. [2]

Auf Malorean habe ich die Rolle des PM als **Architekt des Kontexts** beschrieben: verstreutes Wissen aus Interviews, Vertriebseinwänden, Support, Entscheidungen, Daten, Quellen und schwachen Signalen in eine wiederverwendbare Ressource des Unternehmens verwandeln. [6]

Damit verändert sich der Zweck der Dokumentation grundlegend.

Es geht nicht darum, mehr Seiten zu produzieren.

Es geht darum, die Elemente bereitzustellen, die jemand für eine Entscheidung braucht, ohne die gesamte Überlegung jedes Mal neu rekonstruieren zu müssen.

Ein guter Kontext ermöglicht es, zu verstehen:

- was Kunden tatsächlich gesagt haben;
- welche Hypothesen bereits getestet wurden;
- welche Einschränkungen relevant sind;
- welche grundlegenden Entscheidungen getroffen wurden;
- welche Kompromisse schon akzeptiert wurden;
- mit welchen Kennzahlen sich das Ergebnis beurteilen lässt.

Je größer die Autonomie, desto verlässlicher und gemeinsamer muss dieser Kontext werden.

Sonst verteilt das Unternehmen Handlungsmöglichkeiten, ohne zugleich die Fähigkeit zum Verstehen zu verteilen.

Die Gründe hinter Entscheidungen bewahren

Diese Neuverteilung schafft ein weiteres Problem.

Mehr Entscheidungen werden vor Ort getroffen, manchmal von unterschiedlichen Menschen, manchmal mithilfe von Agenten, die bei früheren Abwägungen nicht dabei waren.

Doch ohne gespeicherte Begründung wird eine Entscheidung schnell wieder verhandelbar.

Man erinnert sich noch an das Ergebnis, aber nicht mehr an die Umstände.

Man weiß, dass eine Regel existiert, aber nicht mehr, weshalb sie gewählt wurde.

Ein neuer Kunde kommt, eine neue Führungskraft übernimmt, ein Team verändert sich – und schon beginnt eine längst entschiedene Diskussion erneut, mit weniger Informationen als beim ersten Mal.

Genau dafür verwende ich **Product Decision Records**. [7]

Ein PDR dokumentiert nicht jede kleine Abwägung.

Er hält Produktentscheidungen fest, die wichtig genug sind, um mehrere Teams, Produkte oder spätere Situationen zu beeinflussen.

Insbesondere bewahrt er:

- den Kontext;
- die betrachteten Alternativen;
- die getroffene Entscheidung;
- die Gründe dafür;
- positive und negative Folgen;

- akzeptierte Risiken;
- Bedingungen, unter denen die Entscheidung erneut geprüft werden sollte;
- die Personen, die entschieden haben oder hinzugezogen wurden.

Es geht nicht darum, zusätzliche Verwaltung zu schaffen. Im Gegenteil: Ein guter PDR sollte verhindern, dass dieselbe Diskussion fünfzigmal von vorn geführt wird.

Hier hilft eine klare Unterscheidung:

Der Code zeigt, was gebaut wurde; die Entscheidungsdokumentation muss bewahren, warum bestimmte grundlegende Regeln gewählt wurden.

Dauerhafte Autonomie braucht beides.

Schutzmechanismen in das System verlagern

Selbst mit gutem Kontext wäre es ineffizient, von jedem Mitarbeiter zu verlangen, sich sämtliche Unternehmensregeln zu merken.

Einige Unternehmen verlagern deshalb wiederkehrende Prüfungen von menschlichen Freigaben in die Werkzeuge selbst. Diese Entwicklung ist beobachtbar, beweist aber nicht, dass Automatisierung allein Qualität garantiert.

Meta verwendet einen *Diff Risk Score*, um die Wahrscheinlichkeit eines Zwischenfalls durch eine Änderung abzuschätzen und risikogerechte Abläufe zu unterstützen. [8] Cloudflare übersetzt ausdrücklich festgelegte Engineering-Standards in automatisierte Prüfungen. Manche verbindlichen

Regeln können die Genehmigung einer Änderung verhindern, solange sie nicht korrigiert oder ausdrücklich eine Ausnahme zugelassen wurde. [9]

Dieser Wandel verändert die Arbeit der Experten.

Ein technischer Garant muss nicht in fünfzig Pull Requests dieselbe Regel von Hand wiederholen, wenn sie automatisch geprüft werden kann.

Ein Design-Garant muss nicht jeden Button freigeben, wenn das Design System die richtige Wahl leicht und eine Abweichung sichtbar macht.

Ein Produkt-Garant muss nicht jede Woche dieselben Einschränkungen erklären, wenn die grundlegenden Entscheidungen, Zielkunden und Ziele zugänglich sind.

Der Rahmen wird zur Infrastruktur.

Tests, Continuous Integration, Design System, Standards, Sicherheitsregeln, Vorlagen, Anweisungen für Agenten, Kennzahlen und geteilter Kontext übernehmen einen Teil der Kontrollen, die früher Menschen erledigten.

Menschliche Fachkenntnis kann sich dadurch auf die Ausnahmen konzentrieren.

Nicht jede Entscheidung braucht dieselbe Kontrolle

Hier wird das Risiko zum entscheidenden Maßstab.

Eine Textübersetzung und eine Änderung des Berechtigungssystems sollten nicht denselben Genehmigungsweg durchlaufen.

Eine lokale, umkehrbare Änderung benötigt nicht dieselbe Kontrolle wie eine Datenmigration oder eine Entscheidung, die die Positionierung des Unternehmens verändert.

Eine Möglichkeit, solche Organisationen zu verstehen, besteht darin, die Ja-nein-Frage „Brauchen wir eine Freigabe?“ durch eine abgestufte Frage zu ersetzen: **Welche Kontrolle rechtfertigt dieses Risiko?**

Die folgenden drei Stufen sind **ein in diesem Buch vorgeschlagener Denkrahmen**, keine bei allen genannten Unternehmen nachgewiesene Praxis.

Lokale Autonomie

Die Entscheidung bleibt innerhalb der bekannten Ziele und Standards.

Sie ist rückgängig zu machen, hat begrenzte mögliche Folgen und verändert keine grundlegende Regel.

Der Owner entscheidet.

Gezielte Freigabe

Die Entscheidung bringt eine Neuerung, eine Ausnahme oder ein bedeutendes Risiko in einem Bereich mit sich.

Der betreffende Garant wird einbezogen – aus Entwicklung, Design, Produkt oder einem anderen Fachgebiet.

Er übernimmt nicht unbedingt das Projekt. Er bringt das spezifische Fachurteil ein, das sonst fehlen würde.

Eskalation

Die Entscheidung verändert die Grenzen des Systems selbst.

In der Technik betrifft das beispielsweise grundlegende Architekturentscheidungen, Sicherheit, sensible Daten, kritische Migrationen oder regulatorische Risiken.
Im Design geht es um wichtige neue Interaktionen, Brüche mit dem Design System oder erhebliche Auswirkungen auf andere Nutzerwege.
Im Produktbereich kann es um einen anderen Zielkundenkreis, das Wertversprechen, Preise, große Investitionen, die Roadmap oder strategische Grundsatzentscheidungen gehen.
In solchen Fällen entscheidet der Owner nicht mehr einfach **innerhalb** des vorgegebenen Rahmens.
Er schlägt vor, den Rahmen selbst zu verändern.
Nach dem hier vorgestellten Modell ist das eine Situation, in der ein Garant oder eine höhere Entscheidungsebene hinzugezogen werden sollte.

Ohne klare Grenzen drohen zwei gegensätzliche Risiken

Fehlen deutliche Entscheidungsgrenzen, ist die Organisation zwei gegensätzlichen Gefahren ausgesetzt.
Die erste ist Bürokratie.
Alles muss nach oben weitergegeben werden. Experten werden zu Engpässen. Führungskräfte müssen belanglose Entscheidungen genehmigen, und die Autonomie verschwindet.
Die zweite ist verwässerte Verantwortung.

Jeder entscheidet vor Ort, ohne zu erkennen, wann eine Entscheidung nicht mehr lokal ist. Widersprüchliche Entscheidungen häufen sich, bis für Kunden oder Unternehmen ein sichtbares Problem entsteht.
Auf Malorean habe ich verschiedene Werkzeuge für organisatorische Kohärenz beschrieben, die solche Grenzen verdeutlichen: gemeinsame Begriffe, einseitige Dokumente, eine Hierarchie von Kennzahlen, ein Entscheidungsprotokoll und Eskalationsregeln. [10]
Sie sind nicht wertvoll, weil sie mehr Dokumentation erzeugen. Sie zwingen uns, bestimmte Fragen zu beantworten:

- Wer entscheidet?
- Über welche Themen?
- Zu welchem Zeitpunkt?
- Was darf lokal bleiben?
- Was muss nach oben weitergegeben werden?
- Unter welchen Bedingungen darf eine Entscheidung neu aufgerollt werden?

Die beiden Gefahren führen zu einer organisatorischen Hypothese: **Entscheidungen an der richtigen Stelle zu treffen** könnte wichtiger sein, als zusätzliche Freigaben einzuführen oder bestehende abzuschaffen.

Entscheiden heißt auch, einen Verlust zu akzeptieren

Schutzmechanismen dürfen nicht nur technischer Art sein.

Ein wesentlicher Teil des Rahmens besteht in der Fähigkeit, echte Abwägungen zu treffen.

Unternehmen treffen oft so vage Entscheidungen, dass alle scheinbar einverstanden sind. Jeder liest seine eigene Priorität hinein, und erst bei der Umsetzung stellt sich heraus, dass die unterschiedlichen Anforderungen unvereinbar sind.

Ich nenne das **weiche Entscheidungen**. [11]

In einer sehr autonomen Organisation sind sie besonders gefährlich.

Ist das Ziel unklar, kann jedes Team für sich genommen vernünftig entscheiden, während die Gesamtheit der Entscheidungen widersprüchlich wird.

Mehr Autonomie verlangt daher, dass wir die akzeptierten Verluste sichtbar machen.

Welcher Kunde hat Vorrang?

Welches Kundensegment werden wir vorerst nicht bedienen?

Welches Qualitätsniveau ist unverhandelbar?

Auf welche Einschränkung können wir vorübergehend verzichten?

Welches Ziel hat Vorrang, wenn zwei Ziele miteinander in Konflikt geraten?

Autonomie braucht keine Regeln, die alles vorschreiben.

Sie braucht einen Rahmen, der die wirklichen Abwägungen deutlich genug macht, damit lokale Entscheidungen in dieselbe Richtung führen.

Ergebnisse zu messen wird zum letzten Schutzmechanismus

Autonomie endet nicht, sobald etwas ausgeliefert ist.

Wer wirklich die Verantwortung für ein Projekt trägt, muss auch beobachten, was danach geschieht.

Im ersten Kapitel haben wir zwischen Produktionsmenge und geschaffenem Wert unterschieden. Hier lautet die Frage anders: **Woher weiß jemand im Nachhinein, ob die eigene Entscheidung sinnvoll war?**

Canva bietet dafür einen konkreten Mechanismus: Die Teams können Experimente mit einer Schätzung ihrer geschäftlichen Auswirkungen verbinden. [12]

Für jedes Projekt bedeutet das, zu klären:

- welche Veränderung erwartet wurde;
- welche Messgröße sie sichtbar macht;
- welches Qualitätsniveau erhalten bleiben sollte;
- was die Nutzer tatsächlich getan haben;
- ob man fortfahren, korrigieren, aufgeben oder genauer untersuchen sollte.

Die Messung schließt den Kreis der Verantwortung.

Ohne sie droht Autonomie zu einer bloßen Erlaubnis zu werden, mehr zu produzieren.

Mit ihr wird daraus Verantwortung für ein Ergebnis.

Autonomie bedeutet nicht den Rückzug des Managements

Das in Kapitel 1.4 beschriebene Modell autonomer Produktteams von Marty Cagan verdeutlicht die Rolle von

Führungskräften. Wer einem Team ein Problem anvertraut, ohne die Lösung vorzugeben, muss Ziele und strategischen Kontext klären. [13] Cagan betont außerdem die Förderung von Kompetenzen, die Begleitung der Teams und die Bedeutung guter Führung: Menschen entscheiden zu lassen heißt nicht, sie allein zu lassen. [14]

Der Vergleich hat allerdings eine wichtige Grenze. Cagan beschreibt die Verantwortung von Produktteams mit ergänzenden Kompetenzen. Das Konzept des Garanten in diesem Buch beantwortet eine andere Frage: Wie bleibt die Verantwortung eines Fachbereichs erhalten, wenn die Umsetzung zwischen verschiedenen Berufen wechselt?

Der Garant schützt den Rahmen, statt jede Handlung abzusegnen

Kapitel 1.6 hat den Garanten als Verantwortlichen für die Kohärenz eines Fachgebiets eingeführt, auch wenn er nicht jede Aufgabe darin selbst erledigt. Hier wird seine Aufgabe konkret: verständliche Regeln festlegen, wiederkehrende Prüfungen automatisieren und dann eingreifen, wenn Risiko oder Neuheit die selbstständige Entscheidungskompetenz des Teams übersteigen.

Muss er alles persönlich prüfen, wird er erneut zum Engpass. Ist niemand mehr verantwortlich, verwässert die Verantwortung.

Autonomie bedeutet also weder, jede Handlung genehmigen zu lassen, noch sämtliche Kontrollen

abzuschaffen. Sie bedeutet, mehr lokale Entscheidungen zu ermöglichen und dabei klare Verantwortlichkeiten zu bewahren.

Doch diese Veränderung führt zu einer Frage, die über Produkt und Engineering hinausgeht.

Wenn der Kontext direkt zugänglich ist, Entscheidungen näher am eigentlichen Arbeitsgeschehen fallen und Agenten einen Teil der Abstimmung übernehmen, **was geschieht dann mit den Managementebenen, die bislang diese Funktionen erfüllt haben?**

Malorean Corp. zeigt uns zunächst, wie diese Frage konkrete Menschen betrifft. Anschließend betrachten wir ihre Folgen für das gesamte Unternehmen.

Literaturverzeichnis

[1] How Cursor Is Navigating the New Shape of Product Craft — Atlassian / Cursor

- URL: https://www.atlassian.com/blog/how-we-build/how-cursor-is-navigating-the-new-shape-of-product-craft
- Veranschaulicht, wie Ziele, Schutzmechanismen und Produkturteil durchlässigere Rollen steuern können.

[2] Issue tracking is dead — Linear

- URL: https://linear.app/next
- Befürwortet gemeinsame Kontexte aus Kundenfeedback, Strategie, Entscheidungen und Code statt isolierter Tickets.

[3] Le backlog n’est pas un dépotoir : c’est un outil d’action — Malorean

- URL: https://malorean.net/articles/2026-06-03-le-backlog-nest-pas-un-depotoir.de.html
- Unterscheidet den Backlog als zeitweiliges Koordinationswerkzeug vom dauerhaften Produktgedächtnis.

[4] 5 insights from Ramp’s CPO on building for hypergrowth — Ramp

- URL: https://ramp.com/blog/5-insights-from-ramps-cpo-reddit-ama
- Beschreibt die Konzentration auf wichtigste Kundengruppen und das bewusste Ablehnen ablenkender Chancen.

[5] What It Takes to Build Great Products Now — Figma

- URL: https://www.figma.com/blog/what-it-takes-to-build-great-products-now/
- Erinnert daran, dass schnelle Prototypen reale Sonderfälle und Qualitätsanforderungen nicht ersetzen.

[6] Le PM comme architecte du Contexte — Malorean

- URL: https://malorean.net/articles/2026-06-08-le-pm-comme-architecte-du-contexte.de.html
- Beschreibt, wie verstreute Kundeninformationen und Entscheidungen zu wiederverwendbarem Kontext werden.

[7] Product Decision Record : tracer les choix produit qui structurent l'entreprise — Malorean

- URL: https://malorean.net/articles/2026-06-03-product-decision-record-tracer-les-choix-produit.de.html
- Erläutert knappe Aufzeichnungen grundlegender Entscheidungen mit Gründen, Alternativen und Bedingungen einer erneuten Prüfung.

[8] Diff Risk Score: AI-driven risk-aware software development — Meta

- URL: https://engineering.fb.com/2025/08/06/developer-tools/diff-risk-score-drs-ai-risk-aware-software-development-meta/
- Stellt einen Risikowert vor, mit dem Prüfprozesse an wahrscheinliche Störungen angepasst werden.

[9] How Cloudflare enforces engineering standards using AI — Cloudflare

- URL: https://blog.cloudflare.com/engineering-standards-enforcement/
- Zeigt, wie verbindliche Entwicklungsstandards als automatisierte Kontrollen umgesetzt werden.

[10] Les outils de cohérence organisationnelle — Malorean

- URL: https://malorean.net/articles/2026-06-02-les-outils-de-coherence-organisationnelle.de.html

- Beschreibt gemeinsame Begriffe, Ziele, Kennzahlen, Entscheidungen und Eskalationsregeln als Koordinationswerkzeuge.

[11] Pourquoi les organisations préfèrent les décisions molles — Malorean

- URL: https://malorean.net/articles/2026-06-02-pourquoi-les-organisations-preferent-les-decisions-molles.de.html
- Analysiert scheinbaren Konsens, der Verluste verschleiert und Widersprüche in die Umsetzung verschiebt.

[12] Measuring Commercial Impact at Scale — Canva

- URL: https://www.canva.dev/blog/engineering/measuring-commerical-impact-at-scale/
- Beschreibt selbstständige Messungen der geschäftlichen Wirkung von Experimenten.

[13] Team Objectives – Empowerment — Marty Cagan, Silicon Valley Product Group (2020)

- URL: https://www.svpg.com/team-objectives-empowerment/
- Befürwortet die Übergabe von Problemen an Teams mit ausreichend strategischem Kontext.

[14] Product Leadership Is Hard — Marty Cagan, Silicon Valley Product Group (2020)

- URL: https://www.svpg.com/product-leadership-is-hard/
- Erklärt, warum autonome Teams anspruchsvolle Führung und aktive Unterstützung benötigen.

9 — Malorean Corp. — Zwei vernünftige Entscheidungen

Am Montagmorgen begann Vivien, der die beiden Entwicklungsteams koordinierte, die Besprechung mit derselben Frage wie seit Jahren.

— Wer sagt mir, wie weit wir mit den Projekten dieser Woche sind?

Er hatte die Übersicht bereits geöffnet. Tickets, Verantwortliche, Fristen und Hindernisse standen ordentlich in den Spalten, die er über Jahre hinweg verbessert hatte. Er kannte die beiden Teams mit je fünf Entwicklern, ihre Stärken, Gewohnheiten und die Aufgaben, bei denen man nachhaken oder lieber etwas abwarten sollte.

Zunächst antwortete niemand.

Dann meldete sich Lucas.

— Bei der Rückverfolgbarkeit habe ich am Freitag das Mini PRD aktualisiert. Wir haben die Suche nach Chargen mit dem Kunden getestet. Als Nächstes müssen wir die Fälle mit unvollständigen Daten prüfen. Die offenen Fragen stehen im Kontext.

— Und wann lieferst du? fragte Vivien.

— Das hängt von den Prüfungen der historischen Lagerdaten ab. Thomas muss sich einen Punkt mit mir ansehen.

Thomas nickte. Er arbeitete selbst an einer Verbesserung des Prozesses zur Versandvorbereitung und hatte den Fortschritt bereits im gemeinsamen Werkzeug dokumentiert.
Vivien ging die übrigen Entwickler durch. Die Antworten ähnelten einander. Jeder kannte sein Problem inzwischen besser, hatte bestimmte Entscheidungen getroffen und konnte den nächsten Schritt erklären. Einige wollten noch Hilfe beim Aufteilen ihrer Aufgaben. Die anderen hatten nicht bis zur Besprechung gewartet, um weiterzuarbeiten.
Am Ende betrachtete Vivien seine Übersicht.
Alles, was er gerade gehört hatte, stand dort schon, teilweise noch ausführlicher.
— Gut, sagte er. Damit wären wir durch.
Die Besprechung hatte zwölf Minuten gedauert.
Früher hätte er fünfundvierzig gebraucht.
Vivien wusste nicht, ob er sich darüber freuen sollte.

Er arbeitete schon lange genug bei Malorean, um die Zeit erlebt zu haben, als alle Entwickler noch im Büro in Le Havre saßen. Er hatte die Auslieferungen geplant, Verzögerungen bearbeitet, Prioritäten mit dem Product Management ausgehandelt und Hubert erklärt, warum etwas, das für Freitag versprochen war, erst am folgenden Dienstag fertig würde.
Als das Unternehmen nach der Pandemie auf Fernarbeit umgestiegen war, hatte er seine Arbeitsweise angepasst. Gespräche auf dem Flur reichten nicht mehr. Er verlangte,

dass Entscheidungen schriftlich festgehalten, Hindernisse sichtbar gemacht und Absprachen mit anderen Abteilungen dokumentiert wurden.

Das wurde seine tägliche Arbeit: Informationen weiterzugeben, die sonst niemand vollständig überblickte.

Er mochte das. Besonders schätzte er die Augenblicke, in denen ein festgefahrenes Projekt wieder in Gang kam, weil endlich die richtigen Menschen miteinander sprachen.

Doch seit KI-Werkzeuge Teil der Teamarbeit geworden waren, hatte sich etwas verändert. Die Entwickler produzierten schneller. Vor allem aber begannen manche, Projekte anders anzugehen. Lucas sprach direkt mit Kunden. Alice stellte einen Produktkontext bereit, der die Gründe für Entscheidungen erklärte, statt nur Funktionen aufzulisten. Bob begann, auf seinen Projekten ähnlich zu arbeiten.

Und inzwischen erstellte ein Agent einen Teil der Zusammenfassungen, die Vivien früher eigenhändig geschrieben hatte.

Die Arbeit war nicht verschwunden. Sie verlagerte sich Stück für Stück von ihm weg oder fand woanders statt.

Mit niemandem hatte er bisher darüber gesprochen.

8

Am folgenden Mittwoch zeigte Lucas Alice eine weiterentwickelte Vorführung der Chargenrückverfolgbarkeit.

— Wir haben getestet, was bei einer Meldung über eine möglicherweise fehlerhafte Charge passiert. Die bekannten

Produktionsvorgänge finden wir. Ich habe den Status „in Prüfung“ ergänzt, damit niemand glaubt, es gebe sicher nichts zu beanstanden.

Alice betrachtete den Bildschirm.

Ein Hinweis erklärte, dass die Situation noch geprüft werden musste. Für ältere Fertigungen erschien eine zweite Meldung: Bestimmte Informationen seien nicht vorhanden.

— Das ist ehrlicher als die erste Version, sagte sie.

— Ja. Der Qualitätsverantwortliche wollte ausdrücklich, dass wir „kein Problem erkannt“ nicht mit „die nötigen Informationen fehlen“ verwechseln.

Alice nickte. Über diesen Unterschied hatten sie bereits im Kontextdokument gesprochen.

— Steht das im Mini PRD?

— Ja. Und ich habe Testfälle ergänzt.

— Dann mach weiter.

Alice musste weder die Position des Hinweises noch seine genaue Formulierung festlegen. Lucas wusste, welches Ergebnis sie erreichen wollten, hatte sich für eine Darstellung entschieden und wollte sie mit dem Kunden prüfen.

Etwas später zeigte Thomas Vivien eine Verbesserung des Versandmoduls.

Hier lag das Problem anders. Bei einigen Kunden blieben Bestellungen aufgrund von Zwischenzuständen oder fehlerhaften Angaben hängen, obwohl kein ausdrückliches

Verbot vorlag. Die Logistikteams mussten Vorgänge korrigieren lassen, bei denen kein Risiko bekannt war.
Thomas hatte deshalb die Regeln für die Versandvorbereitung präzisiert: Eine als „gesperrt“ markierte Charge durfte nicht versandt werden. In anderen Fällen blieb der Zustand sichtbar, und der Ablauf ging mit den bestehenden Kontrollen weiter.
— Bist du sicher, dass du damit nicht das Verhalten anderer Module änderst? fragte Vivien.
— Bei den bisherigen Fällen ja. Ich habe Regressionstests hinzugefügt und die Sperrregeln nicht angefasst.
Vivien hatte keinen Grund, an Thomas’ Vorsicht zu zweifeln. Der kannte mitunter Ausnahmen, die selbst in der alten Dokumentation nicht mehr erwähnt wurden.
— Gut. Lass die Änderung prüfen, bevor es weitergeht.
Thomas nickte.
Die beiden Änderungen waren allerdings noch nie in einem gemeinsamen Szenario getestet worden.

Am Freitag tauchte das Problem in einer integrierten Testumgebung auf.
Eine Person aus dem Support hatte einen vollständigen Ablauf vorbereitet: Wareneingang eines Rohstoffs, Fertigung, Lieferantenwarnung, Identifizierung betroffener Produkte und Vorbereitung des Versands.
Lucas verfolgte die Vorführung per Video. Thomas war dabei. Alice hatte sich einladen lassen, weil sie wissen wollte, wie

jemand den neuen Ablauf verstand, der ihn nicht selbst entworfen hatte.

Die ersten Minuten liefen gut.

Die Chargennummer wurde gefunden. Die Fertigungen erschienen. Der Hinweis „in Prüfung“ wurde eingeblendet. Dann wechselte der Supportmitarbeiter zum Versandmodul. Der Lieferschein blieb in Vorbereitung. Kein Hinweis hinderte ihn daran, fortzufahren.

— Moment, sagte Alice. Geht es hier um dieselbe Charge?

Lucas beugte sich zum Bildschirm.

— Ja. Warum?

— Oben steht „in Prüfung“, aber hier könnten wir einfach weitermachen, als wäre alles in Ordnung.

Thomas öffnete die Bestelldetails.

— Sie ist nicht gesperrt. Sie ist in Prüfung. Das sind zwei verschiedene Zustände.

— Für mich, erwiderte Lucas, bedeutete „in Prüfung“ gerade, dass wir sie nicht als freigegeben betrachten dürfen, solange niemand entschieden hat.

Thomas wandte sich zur Kamera.

— Warum hast du sie dann nicht als „gesperrt“ markiert?

— Weil ich kein pauschales Verbot einführen wollte. Manche Kunden müssen einen Fall untersuchen können, ohne den gesamten Betrieb stillzulegen. Ihr Qualitätsverantwortlicher soll entscheiden, welche Produkte tatsächlich betroffen sind.

— Und in meinem Modul, entgegnete Thomas, gibt es alte Chargen mit unvollständigen Informationen. Wenn ich alles sperre, was nicht ausdrücklich genehmigt wurde, können manche Bestellungen nie mehr versandt werden.
Alice schaute zwischen den beiden Ansichten hin und her.
Sie verstand Lucas' Gründe. Und sie verstand die von Thomas.
Das Problem war, dass jemand im einen Teil des ERP „in Prüfung" lesen und im anderen ohne einen Hinweis fortfahren konnte.
— Wir brechen die Vorführung hier ab, sagte sie.
Niemand protestierte. Sie hatten keine echten Kundendaten verändert, und keine der Änderungen war produktiv eingesetzt worden. Genau dafür war der Test da.
Trotzdem fühlte sich das Schweigen danach anders an als nach einer erfolgreichen Vorführung.

Vivien kam in die Videokonferenz, als man ihm erklärte, was geschehen war.
Er ließ Lucas seine Entscheidung erläutern und Thomas die eigene. Dann fragte er nach den Tests und nach dem Umfang der beiden Änderungen.
— Habt ihr jeder ein Mini PRD? fragte er.
— Ja, sagte Lucas.
— Und eure Entscheidungsregeln?

— Ja, antwortete Thomas. Ich habe die bestehende Sperrlogik nicht geändert. Es ging nur um eine lokale Verbesserung der Versandvorbereitung.

— Ich auch nicht, sagte Lucas. Ich habe Informationen ergänzt, damit wir Unsicherheit nicht verstecken. Genau das hatten wir für wichtig erklärt.

Vivien öffnete die beiden Dokumente nebeneinander.

Sie waren nicht falsch. Sie behandelten nur unterschiedliche Fragen.

Im ersten stand, wie Unsicherheit in der Rückverfolgbarkeit sichtbar gemacht wird. Im zweiten stand, wann ein Versand gestoppt werden soll. Keines beantwortete die Frage, die beide miteinander verband: Was sollte die Software tun, wenn eine Charge verdächtig, aber noch nicht ausdrücklich gesperrt war?

— Zwischen den beiden Projekten fehlt eine Regel, sagte Vivien.

Thomas nickte.

— Technisch können wir mehrere Verhaltensweisen umsetzen. Aber wir müssen wissen, welche wir wollen.

— Und wir müssen den Kunden fragen, was sein Qualitätsverantwortlicher in so einem Fall tatsächlich macht, ergänzte Lucas. Ich habe Vermutungen, keine Regel, die für alle gilt.

Alice ergriff das Wort.

— Ich werde klären, welches Produktverhalten wir wollen. Aber nicht allein. Wir brauchen den Support, Lucas, Thomas und wahrscheinlich einen Kunden, um die Folgen abzuschätzen.
Vivien sah auf.
— Dann organisiere ich für morgen eine Besprechung.
Alice zögerte.
— Du kannst uns helfen, die richtigen Menschen zusammenzubringen. Aber du musst die Entscheidung nicht für uns treffen.
Sie wollte ihn damit nicht ausschließen. Vivien empfand den Satz trotzdem als Verschiebung seiner Rolle.
Jahrelang hatte er dafür gesorgt, dass Probleme den richtigen Weg durch die Organisation fanden. Diesmal wussten die Beteiligten bereits, dass sie miteinander reden mussten. Sie brauchten niemanden, der aus dem Konflikt eine weitere Ticketkette machte.
Sie brauchten eine Entscheidung, die keiner der beiden Entwickler allein treffen durfte.

Am folgenden Tag bat Alice zunächst alle zu erklären, was sie vermeiden wollten.
Lucas wollte nicht, dass ein Qualitätsverantwortlicher vollständige Rückverfolgbarkeit annahm, wenn wichtige Daten fehlten.

Thomas wollte nicht, dass eine ungenaue Regel den Betrieb von Kunden mit lückenhaften historischen Daten unnötig stoppte.

Der Support erinnerte daran, dass eine Lieferantenwarnung nicht zwangsläufig einen sofortigen Rückruf bedeutete. Manche Kunden mussten erst die betroffenen Chargen eingrenzen, ihre Qualitätsabteilung hinzuziehen und über die nächsten Maßnahmen entscheiden.

— Gut, sagte Alice. Vielleicht haben wir versucht, einem einzigen Status zwei verschiedene Bedeutungen zu geben.

Sie zeigte einen Vorschlag.

Eine Charge konnte als **„zu prüfen“** gekennzeichnet werden, ohne dass das ERP vorgab, anstelle des Kundenunternehmens eine regulatorische oder qualitätsbezogene Entscheidung zu treffen. In Abläufen, die diese Charge verwendeten oder versandten, musste die Warnung jedoch sichtbar sein und die vorgesehene Bestätigung eingefordert werden, statt einfach zu verschwinden.

Unsichere historische Angaben mussten weiterhin ausdrücklich als unsicher erscheinen. Fehlende Daten durften nicht als Zeichen dafür ausgegeben werden, dass kein Risiko bestand.

— Das lässt sich im Versandmodul überprüfen, sagte Thomas. Aber ich möchte auch ältere Bestellungen testen.

— Und ich will die Begriffe mit dem Kunden besprechen, fügte Lucas hinzu. „Zu prüfen“ bedeutet für seinen Qualitätsverantwortlichen vielleicht etwas anderes als für uns.

Alice nickte.

Sie hatten die Lösung noch nicht fertig entworfen. Aber jetzt wussten sie endlich, wer was entscheiden musste.

Sie hielt die vorläufige Entscheidung mit Gründen und offenen Fragen im gemeinsamen Kontext fest. Dazu nahm sie die zu überprüfenden Fälle auf. Außerdem ergänzte sie die Entscheidungsübersicht: Wenn eine lokale Entscheidung die Bedeutung einer Information verändert, die mehrere Module verwenden, ist sie nicht automatisch grün – selbst wenn die Codeänderung rückgängig zu machen wäre.

— Das hätten wir früher erkennen müssen, sagte Lucas.

— Ja, erwiderte Alice. Aber genau deswegen testen wir. Jetzt wissen wir, worauf wir beim nächsten Mal achten müssen.

Thomas betrachtete die beiden Dokumente.

— Wir brauchen auch einen Test, der mehrere Module verbindet. Wir haben viele Tests, die das Verhalten jedes Moduls einzeln prüfen. Aber kaum welche, die kontrollieren, wie sich die Module zusammen verhalten.

Vivien hörte seit einigen Minuten zu und machte eine Notiz. Zum ersten Mal in diesem Gespräch begegnete ihm ein Problem, das er nur zu gut kannte: Niemand hatte den vollständigen Überblick über alle Abhängigkeiten.

Gleichzeitig überraschte ihn, wie die Lösung entstand. Alice hatte keine Bildschirme vorgegeben. Lucas und Thomas warteten nicht auf eine Aufteilung der Aufgaben. Der Support brachte die Erfahrung der Kunden ein. Viviens eigener Beitrag bestand darin, die Lücke sichtbar zu machen und das Gespräch zu erleichtern.

Das war nützlich gewesen.

Aber es hatte fast nichts mit den Auslieferungsbesprechungen zu tun, die er jeden Montag organisierte.

Am Nachmittag erhielt Thomas eine Nachricht von einem weniger erfahrenen Entwickler aus dem anderen Team.

Dieser verstand nicht, warum ein alter Lagerprozess zwei scheinbar identische Prüfungen ausführte. Sein Code-Agent schlug vor, eine davon zu entfernen.

Thomas öffnete die Datei.

— Fass nichts an, bevor wir es gemeinsam angesehen haben, antwortete er.

Sie teilten eine halbe Stunde lang den Bildschirm. Die zweite Prüfung deckte einen Sonderfall ab, der auf eine ältere Produktänderung zurückging und im Code schlecht erklärt war. Thomas wusste nicht mehr genau, aus welcher Version sie stammte, aber er wusste, wo er suchen musste.

— Dein Werkzeug hat sich keine falsche Antwort ausgedacht, erklärte er. Es hat zwei ähnliche Kontrollen gesehen und

vorgeschlagen, das Ganze zu vereinfachen. Ohne die Geschichte der Software kann ich das nachvollziehen.

— Verwendest du wirklich nie KI beim Programmieren? fragte der Entwickler.

— Nicht, um mir bei solchen Dingen Änderungen auf gut Glück vorschlagen zu lassen.

— Aber du hast gerade dreißig Minuten mit etwas verbracht, das Lucas vielleicht in fünf Minuten überprüft hätte.

Thomas dachte nach.

— Vielleicht. Und vielleicht hätte Lucas dasselbe entdeckt. Entscheidend ist, dass du verstehst, warum diese Prüfung existiert. Beim nächsten Mal kannst du sowohl deinem Werkzeug als auch mir eine bessere Frage stellen.

Der Entwickler fragte, ob er die Erklärung dokumentieren dürfe.

— Unbedingt. Ich habe genug davon, der Einzige zu sein, der sich daran erinnert.

Thomas beendete das Gespräch und war selbst von seinen Worten überrascht.

Jahrelang hatte ihn seine Kenntnis besonderer Fälle unentbehrlich gemacht. Manchmal hatte er gefürchtet, KI könnte seine Erfahrung entwerten. Jetzt sah er eine andere Möglichkeit: Er konnte anderen helfen, das fehlende Urteilsvermögen zu entwickeln, statt jede Antwort für sich zu behalten.

Er öffnete das gemeinsame Dokument und schrieb ein paar Zeilen.

Bei ihrem Montagstermin fragte Hubert Alice, wie das Experiment vorankam.
Sie berichtete von den beiden unvereinbaren Entscheidungen, dem Test, der sie ans Licht gebracht hatte, und der Anpassung der Regeln.
— Ihr habt also zwei Entwickler selbst entscheiden lassen, und beinahe hätte das zu einem Widerspruch geführt? fasste Hubert zusammen.
— Ja.
— Nicht sehr beruhigend.
— Wir haben es erkannt, bevor etwas produktiv ging, entgegnete Alice. Das Problem liegt nicht darin, dass zwei Menschen Entscheidungen getroffen haben. Es liegt darin, dass keine unserer Regeln beschrieb, wie ihre Entscheidungen zusammenpassen sollten.
Hubert betrachtete die Notizen zum Vorfall.
— Früher hätten wir wohl eine zusätzliche Freigabebesprechung angesetzt.
— Vielleicht. Aber eine weitere Besprechung hätte das Problem nicht zwangsläufig aufgedeckt. Wir brauchten jemanden, der den vollständigen Ablauf testet, und klare Verantwortlichkeiten, wenn eine Entscheidung mehrere Bereiche berührt.

— Und wer entscheidet jetzt?

— Lucas verantwortet sein Projekt. Thomas verantwortet seins und hilft uns bei den technischen Risiken. Ich bin Garantin für die Kohärenz des Produkts. In heiklen Fällen holen wir die zuständigen Spezialisten dazu. Das alles können wir nicht einfach einem Werkzeug überlassen.

Hubert nickte langsam.

— Und Vivien?

Alice überlegte ein paar Sekunden.

— Er hat die beiden Teams zusammengebracht und uns geholfen, den Konflikt zu verstehen. Aber er musste weder die Aufgaben verteilen noch das Produktverhalten festlegen.

Hubert schien an etwas anderes zu denken.

— Ich habe ihn am Freitag gesehen, sagte er. Er hat mir die neue Projektübersicht gezeigt. Sein Bericht war ausgezeichnet. Aber ich glaube, eine Maschine könnte achtzig Prozent dessen vorbereiten, was er mir erzählt hat.

— Das weiß er selbst, sagte Alice.

— Ich weiß nur nicht, welche Konsequenzen wir daraus ziehen sollten.

Alice hatte keine fertige Antwort.

— Nicht, dass er nutzlos geworden ist, sagte sie schließlich. Wir haben gerade gesehen, wie gut er Menschen zusammenbringen kann. Er muss nur nicht mehr zwangsläufig die Station sein, durch die jedes Projekt hindurchläuft.

Hubert blickte weiter auf seinen Bildschirm.

— Wir müssen mit ihm reden.

Vivien erhielt die Einladung für den nächsten Tag.
Sein erster Gedanke war eine Umstrukturierung. Seit einigen Wochen hatte er mehr freie Zeit im Kalender. Die Entwickler verantworteten ihre eigenen Projekte. Werkzeuge sammelten automatisch einen Teil der Informationen, die er früher abends zusammentragen musste. Er kümmerte sich weiterhin um zwischenmenschliche Schwierigkeiten, half neuen Kollegen und löste komplizierte Situationen. Aber das füllte seine Tage nicht mehr wie zuvor.
Er hatte es gemocht, für die Auslieferungen von Malorean unentbehrlich zu sein.
Jetzt war er sich nicht sicher, ob er es mochte, weniger unentbehrlich zu werden.
Doch die Besprechung begann mit einer Frage, die ihn überraschte.
— Was würdest du gern weiterhin tun, Vivien? fragte Hubert.
— Meine Arbeit, antwortete er.
Hubert lächelte.
— Genau darum geht es.
Vivien sah zu Alice, die eingeladen war, um über die Arbeitsweise der Teams zu sprechen, und dann wieder zu Hubert.
— Wir sollten uns nichts vormachen. Die Teams brauchen mich nicht mehr, um sie jede Woche nach dem Fortschritt zu

fragen. Und ich will nicht den ganzen Tag Zusammenfassungen lesen, die ein Agent ohnehin bereits geschrieben hat.

— Was brauchen sie weiterhin? fragte Alice.

— Konflikte, die niemand aufschreibt. Menschen, die sich nicht trauen zuzugeben, dass sie nicht weiterwissen. Neue Kollegen, die nicht wissen, wen sie fragen sollen. Entscheidungen, die mehrere Teams betreffen. Und manchmal einfach jemanden, der sich Zeit zum Zuhören nimmt.

Er machte eine Pause.

— Aber das bedeutet nicht, dass all diese Dinge weiterhin über mich laufen müssen.

Hubert setzte sich aufrecht hin.

— Nein. Wir müssen dann aber klar bestimmen, wer dafür verantwortlich ist, wenn sich deine Rolle verändert. Ich will nicht einen Jobtitel streichen und erst später entdecken, dass niemand mehr auf die Menschen achtet.

Vivien nickte.

Sie sprachen darüber, wie Entwickler begleitet, teamübergreifende Entscheidungen getroffen und bestimmte Themen weiter von der Geschäftsführung betreut werden sollten. Alice erzählte davon, wie Thomas Kollegen half, den alten ERP-Code besser zu verstehen.

Sie fanden keine endgültige Lösung.

Nach der Videokonferenz blieb Vivien vor seinem Kalender sitzen.

Er fühlte sich weniger bedroht als am Vortag, aber noch nicht wirklich beruhigt. Sein Beruf war nicht verschwunden. Er zerfiel in mehrere Verantwortlichkeiten, die zum Teil anders ausgefüllt werden konnten.

An diesem Abend schaltete er den Computer früher aus als gewöhnlich.

Zum ersten Mal musste er keinen Rückstand bei der Auslieferung aufholen.

Noch wusste er nicht, was er mit der zurückgewonnenen Zeit anfangen sollte.

10 — Wird KI die mittleren Führungsebenen im Unternehmen verschwinden lassen?

Heute kann ein Team unmittelbar auf die für es geltenden Entscheidungen zugreifen, innerhalb bekannter Regeln handeln und einen Teil der Fortschrittskontrolle Agenten überlassen. Es braucht seltener eine Führungskraft, die Informationen weitergibt oder jeden Arbeitsschritt organisiert. Was geschieht also **mit den Führungsebenen, die bisher genau diese Aufgaben erfüllten?**

Die Antwort scheint naheliegend: Wenn KI die Koordination übernimmt, werden mittlere Führungskräfte überflüssig. Doch eine Führungskraft gibt nicht bloß Anweisungen weiter. Sie wägt ab, begleitet Mitarbeiter, verbindet Teams und kümmert sich um Probleme, die in keine Verfahrensanweisung passen. Die eigentliche Frage ist deshalb weniger, wie viele Stellen wegfallen, sondern **wohin die Aufgaben und Entscheidungsbefugnisse dieser Stellen wandern**.

Eine Hierarchie zur Koordination von Entscheidungen

Ein wichtiger Kunde meldet einen Fehler. Der Support gibt die Meldung an seine Führungskraft weiter, die das Produktteam kontaktiert. Das Produktteam bittet Engineering um eine Aufwandsschätzung. Verschiedene Verantwortliche sammeln Informationen, vergleichen Dringlichkeiten, und die

Entscheidung wird gegebenenfalls eine weitere Ebene nach oben gereicht.

Dieser Weg ist nicht unbedingt unsinnig. Jede Führungskraft besitzt einen Teil des Kontexts und kann für einen Teil des Unternehmens verbindlich handeln. Belastend wird die Kette, wenn es keine andere Möglichkeit gibt, ein Problem zu verstehen und zu entscheiden.

Historisch ermöglichte die Hierarchie Unternehmen das Wachstum, obwohl Informationen auf viele Stellen verteilt waren. Mittlere Ebenen übersetzen Unternehmensziele in Prioritäten, koordinieren Teams, beobachten Abhängigkeiten und lösen Konflikte um Ressourcen. Sie wirken auch an der Beurteilung und Entwicklung von Mitarbeitern sowie an der Lösung zwischenmenschlicher Schwierigkeiten mit.

Doch diese Tätigkeiten sind sehr unterschiedlich. Einen Bericht über einen Zwischenfall zu erstellen ist etwas anderes, als zu entscheiden, eine Geschäftsmöglichkeit aufzugeben. Eine Frist zu überwachen verlangt nicht dasselbe Urteil wie einen Konflikt beizulegen oder einen Menschen in seiner Entwicklung zu unterstützen.

Auf Malorean habe ich bereits beim Product Owner eine ähnliche Unterscheidung vertreten: Eine Stelle kann ihre Berechtigung als bloße Vermittlungsstation verlieren, ohne dass die Funktion, die sie erfüllen sollte, unnötig wird. Wir müssen die Arbeit betrachten, nicht allein den Titel der Person, die sie verrichtet. [1]

Wenn die mittleren Ebenen schrumpfen, wer übernimmt dann ihre Aufgaben?

Fernarbeit hatte die Spielregeln der Koordination bereits verändert

KI hat diese Frage nicht erfunden. Unternehmen, die verteilt über mehrere Länder und Zeitzonen arbeiten, mussten lernen, Entscheidungen ohne Flurgespräche und spontane Besprechungen zu treffen.

Bei GitLab bedeutet *Handbook First*, dass Regeln und Kontext schriftlich zugänglich sein sollen. Jeder darf Änderungen vorschlagen, die zuständigen Fachverantwortlichen entscheiden über die Aufnahme. So kann ein Mitarbeiter Informationen finden, ohne darauf warten zu müssen, dass eine Führungskraft sie weiterleitet. [2]

Alan beschreibt eine andere, ergänzende Praxis. Wichtige Entscheidungen werden in GitHub-Diskussionen besprochen: Eine Person entscheidet und setzt die Entscheidung um; eine andere bleibt für den Erfolg des Projekts verantwortlich. Die Betroffenen werden hinzugezogen oder informiert. Eine offene Diskussion verlangt nicht zwangsläufig Einstimmigkeit. [3]

Bei 37signals, dem Unternehmen hinter Basecamp, läuft ein großer Teil der Koordination über schriftliche Diskussionen, Zusammenfassungen und Informationen, die zusammen mit den Projekten aufbewahrt werden, statt in flüchtigen Gesprächen zu verschwinden. [4]

Das sind keine drei Unternehmen ohne Führungskräfte. Es sind drei Möglichkeiten, **die Abhängigkeit von einer Person zu verringern, die Informationen besitzt oder andere immer wieder an frühere Entscheidungen erinnern muss**. Dieser Vorläufer ist für KI-Agenten relevant. Ein System kann eine Regel nachschlagen, eine frühere Entscheidung finden und die verantwortliche Person identifizieren – vorausgesetzt, das Unternehmen hat diese Informationen ausdrücklich festgehalten. Was nur im Kopf einer Führungskraft existiert, bleibt für Kollegen und Agenten gleichermaßen unzugänglich.

Dabei sollten wir Fernarbeit nicht idealisieren. Eine Untersuchung von mehr als 61.000 Microsoft-Mitarbeitern zeigte unter anderem, dass die Netzwerke der Zusammenarbeit nach dem Wechsel ins Homeoffice stärker voneinander getrennt waren. Mehr schriftliche Kommunikation garantiert weder gemeinsamen Kontext noch Zusammenarbeit zwischen Teams. [5]

Das eigentliche Vermächtnis der Fernarbeit ist nicht das Verschwinden des Büros. Es ist die Möglichkeit, einen Teil der Arbeit um Informationen und Entscheidungen herum zu organisieren, die ohne dauerhafte Vermittler zugänglich sind. KI kann auf diesem Fortschritt aufbauen. Sie hat ihn nicht allein hervorgebracht.

KI übernimmt allmählich einen Teil dieser Arbeit

Ein Dokument speichert eine Regel. Ein Dashboard zeigt eine Verzögerung. Ein Agent kann weitergehen: Informationen aus

mehreren Quellen zusammenführen, einen Fortschrittsbericht vorbereiten, einen Widerspruch erkennen, eine Maßnahme vorschlagen oder eine genehmigte Aktion durchführen.

Genau solche Tätigkeiten beanspruchen einen Teil der Zeit vieler Führungskräfte: Konsolidierung, Berichterstattung, Überwachung und Vorbereitung von Entscheidungen. Ihre Kosten könnten sinken. Dass sie vollständig verschwinden, steht nicht fest.

Eine explorative Analyse der HEC Paris, die unter anderem auf 45 Interviews beruht, legt nahe, dass strukturierte und datenreiche Managementaufgaben leichter zu automatisieren sind als solche, die Vertrauen, menschliche Beziehungen und Urteilsvermögen voraussetzen. Die Untersuchung misst weder tatsächlich eingesparte Führungsstunden noch überflüssig gewordene Stellen. [6]

Ein Experiment bei Procter & Gamble zeigt zudem, dass eine Person mit KI bei Aufgaben der Produktinnovation Ergebnisse erreichen kann, die mit denen eines Zweierteams ohne KI vergleichbar sind. Das Werkzeug hilft insbesondere dabei, verschiedene berufliche Perspektiven zusammenzubringen. Doch gelegentliche Zusammenarbeit zu erleichtern bedeutet nicht, langfristig eine Führungskraft zu ersetzen. [7]

In einem weiteren Experiment mit 7.137 Mitarbeitern aus 66 Unternehmen sparten regelmäßige Nutzer eines KI-Assistenten während der zweiten Hälfte der Untersuchung ungefähr zwei Stunden E-Mail-Arbeit pro Woche. Die

Forschenden stellten in sechs Monaten trotzdem keine signifikante Veränderung der Zusammensetzung ihrer Aufgaben fest. Dabei ging es um einen Büroassistenten, nicht um Agenten, die ganze Prozesse eigenständig erledigen. [8] Dieser Unterschied ist entscheidend: **Ein individueller Produktivitätsgewinn ist noch kein organisatorischer Wandel.**

Damit eine Führungsebene weniger notwendig wird, müssen Informationen tatsächlich bei den Teams ankommen, Entscheidungsrechte verändert werden und außergewöhnliche Situationen einen Ort für ihre Klärung finden. Das ist keine reine Werkzeugfrage mehr, sondern eine Organisationsfrage.

Drei Unternehmen, drei sehr unterschiedliche Wege

Unternehmen reagieren keineswegs einheitlich. Drei Beispiele helfen, eine bereits laufende Umstrukturierung, eine auf KI ausgerichtete Vision und die Schwierigkeiten eines beschleunigten Wandels auseinanderzuhalten.

Bayer: eine tatsächlich verkleinerte Hierarchie

Mit seinem Modell *Dynamic Shared Ownership* (DSO) hat Bayer einen Teil der Entscheidungen an interdisziplinäre Teams übertragen. Im Aktionärsbrief für 2025 erklärte die Unternehmensführung, bis zu sechs Organisationsebenen abgeschafft und die Zahl der Managementpositionen um etwa zwei Drittel gesenkt zu haben. Bayer berichtet außerdem von schnelleren Entscheidungen und kürzeren Entwicklungszeiten. [9]

Das Ausmaß verdient Beachtung, auch wenn die Ergebnisse hier vom Unternehmen selbst stammen. Eine solche Transformation verlangt veränderte Zuständigkeiten im Alltag und nicht bloß neue Beschriftungen im Organigramm. Teams entscheiden mehr vor Ort, Führungskräfte konzentrieren sich stärker auf Begleitung, und bestimmte Koordinationsfunktionen werden neu organisiert.

Eine wichtige zeitliche Einordnung verändert die Bedeutung des Beispiels: **DSO begann, bevor generative KI-Agenten verbreitet waren.** Bayer zeigt, dass ein Großkonzern seine Hierarchie stark verkleinern kann. Es beweist nicht, dass KI dafür verantwortlich ist.

Block: Kontext ohne den Umweg über die gesamte Hierarchie verteilen

Block geht bei der Rolle der KI in seiner Vision weiter. In einem Text von Jack Dorsey und dem Investor Roelof Botha beschreiben die Autoren eine Organisation, die auf einem gemeinsamen Informationssystem basiert.

Das Ziel: Mitarbeiter sollen eine Situation verstehen und handeln können, ohne darauf zu warten, dass eine Führungskraft ihnen den Kontext weitergibt. Die Autoren unterscheiden zwischen direkt Mitwirkenden, Verantwortlichen für bereichsübergreifende Ergebnisse (DRI) und Führungskräften, die selbst produktiv mitarbeiten und zugleich andere begleiten, sogenannten *Player-Coaches*. [10]

Führungskräfte, die vor allem Informationen weiterreichen, würden dadurch weniger wichtig. Die Förderung von Menschen, die Verantwortung für Ergebnisse und die Entscheidungen der Unternehmensleitung blieben dagegen notwendig.

Das ist die von Unternehmen und Investor beschriebene Vision, keine unabhängige Bewertung, die einen dauerhaften Erfolg des Modells nachweist.

Meta: wenn die Umsetzung schneller wächst als die Kontrolle

Einen Gegenpol bietet eine Reuters-Recherche über Metas Umstrukturierungspläne von 2026: kleinere Teams, ein hoher Stellenwert der KI sowie geplante oder begonnene Kürzungen bestimmter Führungsebenen. Eine zweite Restrukturierungswelle wurde schließlich verworfen.

Nach internen Dokumenten, die Reuters zitiert, nahm die Zahl der Änderungen an interner Software und Infrastruktur viel schneller zu als die Zahl der neuen Funktionen für Endnutzer. Gleichzeitig stiegen technische Zwischenfälle und der Aufwand zu ihrer Behebung. Meta widersprach einigen Aspekten der Darstellung beziehungsweise relativierte sie. [11]

Aus diesen Zahlen lässt sich nicht schließen, dass KI oder der Abbau von Führungskräften die Zwischenfälle verursachten: Mehrere Veränderungen liefen gleichzeitig ab. Das Beispiel stellt aber eine entscheidende Frage: Was nützt höhere Produktionsleistung, wenn das Unternehmen die Ergebnisse

nicht mehr ausreichend prüfen, stabilisieren und bewältigen kann?

Bayer zeigt, dass Hierarchien abgeflacht werden können. Block entwirft eine KI-orientierte Variante davon. Und Meta erinnert an die Risiken, wenn die Umsetzung schneller wächst als die Fähigkeit zur Koordination. Keiner dieser Wege allein sagt die Zukunft des Managements voraus.

Die Managementaufgaben verschwinden nicht automatisch mit ihren Stelleninhabern

Nehmen wir ein Beispiel: Ein Kunde meldet einen Softwarefehler. Solange das Problem bekannt und seine Lösung geregelt ist, kann das Team handeln, ohne auf mehrere Genehmigungen zu warten. Mit verlässlichen Daten kann es den Fehler untersuchen, korrigieren und andere informieren.

Dann widersprechen sich zwei Diagnosen. Der Agent empfiehlt eine Maßnahme, ein Entwickler eine andere. Die Korrektur könnte einen benachbarten Dienst beeinträchtigen.

Nun muss jemand den Konflikt sichtbar machen, die nötigen Kompetenzen zusammenbringen und entscheiden, welches Risiko vertretbar ist. Autonomie hat die Koordination nicht abgeschafft. Sie hat nur verändert, wie sie geschieht.

Ein Experiment von Martin Krzywdzinski und Mitautoren liefert einen interessanten Hinweis. An einer simulierten Produktionslinie mussten 87 Zweierteams Störungen mithilfe eines KI-Systems beheben, das gelegentlich Fehler machte.

Selbstorganisierte Teams kommunizierten mehr und lösten die Probleme erfolgreicher als Teams mit einer ausdrücklich eingesetzten Führungsperson. Die Analyse betont die entscheidende Rolle der Kommunikation. [12]

Zwei Menschen im Labor sind kein Großunternehmen. Doch die Studie zeigt eine wichtige Unterscheidung: **Ein Team kann ohne formell eingesetzten Chef arbeiten, aber nicht auf die Aufgaben der Koordination verzichten.**

Diese Arbeit können Kollegen, Fachleute, eigens für Entscheidungen beauftragte Personen oder Führungskräfte übernehmen. Wer das tut, hängt von Kompetenzen, Risiken und Konflikten ab.

Daneben bleibt all das, was kein automatisch erstellter Bericht allein leisten kann: jemanden willkommen heißen, ihm bei der Entwicklung helfen, Spannungen lösen, eine schwere Entscheidung begleiten oder geleistete Arbeit anerkennen. Nicht alle Manager erledigen diese Aufgaben gut, und sie können auch anders verteilt werden. Aber jemand braucht weiterhin den Auftrag und die Zeit dafür.

Gallup beobachtet, dass die Nutzung von KI die Sorgen um die eigene Arbeit nicht beseitigt und dass wahrgenommene Unterstützung am Arbeitsplatz mit geringerer Besorgnis einhergeht. Das beweist nicht, dass traditionelle Manager nötig sind. Es erinnert jedoch daran, dass eine Organisation auch eine menschliche Gemeinschaft ist. [13]

Und wer bildet die Menschen aus, die morgen schwierige Entscheidungen treffen sollen?

Auf Malorean habe ich ein mögliches Risiko für junge Entwickler angesprochen: Wenn KI Code sofort schreibt und Fehler behebt, können Lerngelegenheiten wegfallen, bei denen Menschen das Debugging üben, Fehlversuche verstehen und eigenes Urteilsvermögen entwickeln. Das ist eine Sorge, die untersucht werden sollte, kein Nachweis eines allgemeinen Kompetenzverlusts. [14]

Dasselbe Problem betrifft künftige Führungskräfte. Man lernt auch, indem man schwierige Entscheidungen miterlebt, Rückmeldungen erhält und beobachtet, wie Teams Konflikte lösen. Wer Führungsebenen abbaut, muss andere Wege schaffen, diese Erfahrung weiterzugeben.

Im Ergebnis kann jede Aufgabe verschiedene Wege nehmen: Sie wird automatisiert, vom Team übernommen, an Spezialisten übertragen oder zur Unternehmensleitung verlagert. **Eine Aufgabe kann verschwinden, den Besitzer wechseln oder zusätzlich in den Arbeitsbereich eines anderen Menschen fallen.**

Um die Wirksamkeit einer flacheren Organisation zu beurteilen, reicht es daher nicht, gestrichene Positionen zu zählen. Der eingesparte Koordinationsaufwand muss mit neuen Belastungen verglichen werden: Agenten überwachen, Zwischenfälle beheben, Menschen begleiten und ausbilden.

Für diese Neuverteilung gibt es noch keine belastbare Nettobilanz.

Ist eine flachere Organisation wirklich autonomer?

Eine letzte Frage ist in Organigrammen weniger sichtbar: **Wer entscheidet, wenn die Vermittler verschwinden?**

Man kann sich besser informierte Teams vorstellen, die mehr Probleme selbst lösen und nicht jede Entscheidung weiterreichen müssen. Aber auch das Gegenteil ist denkbar: weniger direkte Vorgesetzte und mehr Entscheidungen, die die Unternehmensleitung unmittelbar über digitale Systeme kontrolliert.

Diese Spannung gab es schon vor generativer KI. In einer Untersuchung von Industrieunternehmen zeigen Nicholas Bloom und Mitautoren, dass Werkzeuge, die Mitarbeitern mehr Informationen bereitstellen, mit größerer lokaler Autonomie zusammenhängen. Werkzeuge, die die Kommunikation mit Vorgesetzten erleichtern, können dagegen Zentralisierung fördern: Es wird einfacher, Anweisungen von oben zu erfragen. [15]

Dieser ältere Befund beweist nichts über heutige Agenten. Er verdeutlicht aber einen Mechanismus: **Daten zu verbreiten ist nicht dasselbe wie Macht zu verteilen.**

Algorithmische Managementsysteme machen das Risiko greifbarer. Sie können Aufgaben zuteilen, Prioritäten vorschlagen, Leistung messen und Warnungen auslösen. Forschung zeigt, dass solche Werkzeuge bestimmte

Funktionen der Führung oder Bewertung übernehmen können, auch wenn menschliche Manager weiterhin die Umsetzung begleiten oder Ausnahmen behandeln. [16]

Ein System erscheint nicht als Vorgesetzter im Organigramm. Wenn es aber entscheidet, wer welche Aufgaben erhält, nach welchen Maßstäben Ergebnisse bewertet werden und unter welchen Voraussetzungen eine Handlung erlaubt ist, übt es einen Teil der Autorität aus.

Wir müssen deshalb hinter die Bildschirme blicken: Wer legt die Ziele der Agenten fest? Wer bestimmt die Regeln, und wer darf sie verändern? Wer kann ihnen widersprechen? Und wer besitzt tatsächlich die Macht, die Folgen einer falschen algorithmischen Entscheidung rückgängig zu machen?

Ein Team kann über enorme Umsetzungskapazität verfügen, ohne seine Prioritäten ändern zu dürfen. Ein anderes kann eine Führungskraft behalten und gleichzeitig selbst über Mittel und lokale Entscheidungen bestimmen.

Auf Malorean habe ich bereits jene „weichen Entscheidungen“ beschrieben, die Einigkeit vortäuschen, ohne den eigentlichen Entschluss, die akzeptierten Nachteile oder den Entscheidungsberechtigten deutlich zu machen. Eine Ebene zu streichen ändert nichts, wenn die Macht ebenso schwer zu erkennen bleibt. [17]

Daraus entsteht ein mögliches Paradox: **weniger sichtbare Hierarchie, aber stärker zentralisierte Kontrolle durch Werkzeuge**.

Das ist nicht zwangsläufig so. Dieselben Systeme können Teams helfen, lokaler zu entscheiden, und die Regeln der Machtverteilung transparenter machen. Alles hängt davon ab, was das Unternehmen tatsächlich delegieren möchte – nicht davon, wie viele Kästchen es im Organigramm entfernt.

Was tatsächlich verschwinden könnte

Wird KI die mittleren Führungsebenen in Unternehmen verschwinden lassen?

Manche Ebenen könnten weniger notwendig werden. Doch ihr Verschwinden wäre weder automatisch noch gleichbedeutend mit dem Ende des Managements.

Aufgaben, bei denen vor allem Informationen übertragen, Fortschrittsberichte zusammengetragen oder wiederkehrende Genehmigungen organisiert werden, könnten an Bedeutung verlieren. Andere Aufgaben – schwierige Abwägungen, Konflikte, Weitergabe von Erfahrung und Verantwortung – müssen auch künftig erfüllt werden, möglicherweise auf andere Weise.

Bayer zeigt, dass eine massive Verringerung von Führungsebenen möglich ist, ohne dass KI als Ursache belegt wäre. Block schlägt vor, einen größeren Teil der Koordination einem gemeinsamen System zu übertragen. Meta erinnert daran, dass schnelleres Umsetzen ohne Blick auf Zuverlässigkeit und menschliche Belastung das Unternehmen schwächen kann. Noch ist nicht erwiesen, dass Agenten generell das Management verschwinden lassen.

Der Wandel könnte trotzdem tiefgreifend sein. Führungskräfte müssten seltener als notwendige Durchgangsstation für Informationen oder Genehmigungen dienen. Sie könnten sich stärker der Kohärenz, außergewöhnlichen Situationen und der Entwicklung von Menschen widmen. Anderswo würden manche Aufgaben direkt von den Teams übernommen, andere in Software integriert oder bei wenigen Entscheidern konzentriert.

Vielleicht wird nicht das Management überflüssig. Infrage gestellt wird vielmehr die Hierarchie als verpflichtender Weg zur Koordination.

Für Entwickler hat das konkrete Folgen. Wenn Arbeit weniger Vermittler durchläuft, erhalten sie nicht nur mehr Mittel, um etwas zu produzieren. Sie begegnen den Bedürfnissen, Kompromissen und Ergebnissen, für die jemand einstehen muss, viel unmittelbarer.

Genau diese Veränderung haben wir im gesamten ersten Buch betrachtet: Die Fähigkeit zu handeln nimmt zu – und damit auch die Notwendigkeit, zu verstehen, zu wählen und Verantwortung zu übernehmen.

Das garantiert weder bessere Entscheidungen noch menschlichere Unternehmen. Es zwingt uns aber, Organisation anders zu gestalten: Kontext teilen, Macht erkennbar machen, Erfahrung weitergeben und Verantwortung für Ergebnisse übernehmen.

Die Frage ist nicht nur, was KI uns heute zu produzieren erlaubt, sondern welche Art von Unternehmen wir um diese neue Fähigkeit herum aufbauen wollen.

In der letzten Episode kehren wir zu Malorean Corp. zurück und sehen, wie Führung und Teams ihre Entscheidungen jenseits des Organigramms tatsächlich erleben.

Literaturverzeichnis

[1] Le PO n'est pas un métier, c'est une fonction — et l'IA n'y change rien, elle accélère — Malorean

- URL: https://malorean.net/articles/2026-08-03-le-po-nest-pas-un-metier-cest-une-fonction.de.html
- Essay über den Unterschied zwischen einer nützlichen Funktion und einer Stelle, die hauptsächlich Entscheidungen weiterreicht.

[2] Handbook First — GitLab

- URL: https://handbook.gitlab.com/handbook/company/culture/all-remote/handbook-first/
- Beschreibt schriftliche Dokumentation als gemeinsamen Bezugspunkt mit weiterhin menschlicher Entscheidungsverantwortung.

[3] Comment prendre des décisions dans un environnement de travail asynchrone — Alan

- URL: https://alan.com/fr-fr/blog/healthy-business/a/decisions-asynchrones-methode-alan

- Erklärt schriftliche Entscheidungen, Zuständigkeiten und die Einbindung betroffener Menschen.

[4] The 37signals Guide to Internal Communication — 37signals

- URL: https://37signals.com/how-we-communicate
- Beschreibt asynchrone interne Kommunikation und die dauerhafte Bewahrung von Projektkontext.

[5] The effects of remote work on collaboration among information workers — Microsoft Research

- URL: https://www.microsoft.com/en-us/research/publication/the-effects-of-remote-work-on-collaboration-among-information-workers/
- Untersuchung zu stärker getrennten Zusammenarbeitsnetzwerken nach dem Übergang zur Fernarbeit.

[6] AI Automates Routine Management and Raises the Value of Leadership — HEC Paris

- URL: https://www.hec.edu/en/dare/tech-ai/ai-automates-routine-management-and-raises-value-leadership
- Explorative Untersuchung von Managementaufgaben auf Basis von Interviews und Modellbewertungen, ohne kausalen Nachweis von Stellenabbau.

[7] The Cybernetic Teammate: A Field Experiment on Generative AI Reshaping Teamwork and Expertise — NBER

- URL: https://www.nber.org/papers/w33641

- Experiment bei Procter & Gamble zum Einsatz von KI in Innovationsaufgaben mit verschiedenen Kompetenzen.

[8] Shifting Work Patterns with Generative AI — NBER

- URL: https://www.nber.org/papers/w33795
- Randomisiertes Experiment in 66 Unternehmen mit individueller Zeitersparnis, aber ohne signifikante Änderung der Tätigkeiten in sechs Monaten.

[9] Chairman's Letter — Bayer Annual Report 2025

- URL: https://reports.bayer.com/annual-report-2025/en/to-our-stockholders/chairmans-letter.html
- Bayer berichtet über eine deutliche Verringerung von Hierarchieebenen und Managementpositionen im Rahmen von DSO.

[10] From Hierarchy to Intelligence — Sequoia Capital / Block

- URL: https://sequoiacap.com/article/from-hierarchy-to-intelligence
- Jack Dorsey und Roelof Botha skizzieren Koordination über ein gemeinsames Informationssystem mit DRI und Player-Coaches.

[11] Mark Zuckerberg had a bold plan to replace Meta staff with AI. Here's how it imploded — Reuters

- URL: https://www.timeslive.co.za/news/sci-tech/2026-08-26-special-report-mark-zuckerberg-had-a-bold-plan-to-replace-meta-staff-with-ai-heres-how-it-imploded/

- Reuters-Recherche, wiedergegeben von TimesLIVE, zu organisatorischen und technischen Spannungen bei Meta.

[12] How team organization influences the ability to solve automation failures — AI & Society

- URL: https://link.springer.com/article/10.1007/s00146-025-02761-5
- Experiment mit selbstorganisierten und hierarchischen Zweierteams bei Fehlern in automatisierten Systemen.

[13] Using AI More Does Not Reassure Workers, Managers Do — Gallup

- URL: https://www.gallup.com/workplace/713231/ai-not-reassure-workers-managers-do.aspx
- Befragung zu Sorgen von Beschäftigten im KI-Wandel und wahrgenommener Unterstützung.

[14] L'IA ne devrait pas seulement servir à produire dix fois plus. Elle devrait aussi nous obliger à comprendre dix fois mieux ce que nous produisons — Malorean

- URL: https://malorean.net/articles/2026-07-23-lia-produire-dix-fois-plus-comprendre-dix-fois-mieux.de.html
- Reflexion über mögliche verlorene Lerngelegenheiten für jüngere Entwickler bei KI-generiertem Code.

[15] The distinct effects of Information Technology and Communication Technology on firm organization — NBER

- URL: https://www.nber.org/papers/w14975

- Unterscheidet den Einfluss besserer Informationszugänge und leichterer Kommunikation auf lokale Entscheidungsfreiheit.

[16] Algorithmic Management: From Technology to Politics and Theory — Weizenbaum Journal of the Digital Society

- URL: https://ojs.weizenbaum-institut.de/index.php/wjds/article/view/4_3_9/163
- Überblick über Führungs- und Bewertungsfunktionen algorithmischer Systeme und Möglichkeiten menschlicher Intervention.

[17] Pourquoi les organisations préfèrent les décisions molles — Malorean

- URL: https://malorean.net/articles/2026-06-02-pourquoi-les-organisations-preferent-les-decisions-molles.de.html
- Essay über scheinbare Einigkeit, die Entscheidungen, Verluste und Verantwortlichkeiten verschleiert.

11 — Malorean Corp. — Die wiedergewonnene Zeit

Zwei Monate nach der Vorführung des Urlaubsmoduls eröffnete Hubert Félix Duca die Montagsbesprechung, ohne nach dem Stand der Entwicklungen zu fragen.

— Also, was haben wir letzte Woche gelernt?

Alice teilte ihren Bildschirm.

Vor ihr lagen die Zusammenfassung der jüngsten Kundenrückmeldungen, die Ergebnisse laufender Experimente und einige noch offene Entscheidungen. Ein Agent hatte die Informationen über Nacht erstmals sortiert. Am Morgen hatte Alice die Übersicht geprüft, irreführende Verbindungen entfernt und zwei Fragen ergänzt, auf die der Agent allein nicht gekommen wäre.

— Bei der Chargenrückverfolgbarkeit findet unser Pilotkunde die betroffenen Produktionsvorgänge deutlich schneller, wenn die Daten vollständig sind, begann sie. Aber bei älteren Daten gibt es Fälle, in denen wir unsere Grenzen ausdrücklich zeigen müssen. Eine neue Darstellung werden wir mit seinem Qualitätsverantwortlichen testen.

Hubert nickte.

— Und bei den anderen Kunden?

— Für ein Urteil ist es zu früh, erwiderte Alice. Genau deshalb haben wir die Funktion noch nicht überall eingeführt.

Bob übernahm. Er betreute andere Produktprojekte und hatte kürzlich mehrere Gespräche mit Presales und Support geführt.
— Wir haben außerdem drei Anfragen, die auf den ersten Blick ähnlich wirken, sagte er. Der Agent hat sie unter einer einzigen Funktionsidee zusammengefasst. Dahinter stecken aber mindestens zwei unterschiedliche Probleme. Ich möchte zuerst klären, welche Kunden wir ansprechen wollen, bevor wir investieren.
— Wie viele Entwickler brauchst du? fragte Hubert.
Bob lächelte.
— Im Augenblick keinen.
Hubert lächelte zurück.
Bob sah auf eine Benachrichtigung und fügte hinzu:
— Übrigens probiert der Support seit gestern ein neues System zur Einordnung wiederkehrender Anfragen aus. Jemand aus dem Team hat es mit einem Agenten und anonymisierten Kopien vorbereitet. Sie prüfen, ob die Informationen dadurch schneller bei den richtigen Personen ankommen.
— Wer hat das gestartet? fragte Hubert.
— Der Support. Alice hat bestätigt, dass der Versuch intern und reversibel bleibt und die Ergebnisse von Menschen geprüft werden.
— Wusstest du davon?
— Von der Idee, ja. Nicht von den Einzelheiten.

Huberts erster Impuls war zu fragen, warum ihn niemand vorher informiert hatte. Er hielt sich zurück. Sie hatten ja gerade vereinbart, dass begrenzte, rückgängig zu machende Initiativen nicht mehr seine Genehmigung erforderten.

— Sagt mir Bescheid, ob es funktioniert, sagte er schließlich. Und falls das jemand dauerhaft betreuen muss, möchte ich wissen, wer.

Er hatte den Test nicht genehmigen müssen. Aber über die Ressourcen für eine mögliche breitere Einführung entschied weiterhin er.

Noch vor Kurzem wäre eine solche Antwort in einer Produktsitzung ungewöhnlich gewesen. Damals versuchte man vor allem festzulegen, was die Entwickler in welcher Reihenfolge erledigen sollten. Jetzt verbrachten Alice und Bob mehr Zeit damit, die richtigen Probleme auszuwählen, die Kohärenz der Initiativen zu überprüfen und bei Entscheidungen einzugreifen, die den vereinbarten Rahmen überschritten.

Ihre Stellenbezeichnungen waren unverändert. In den Personalunterlagen hießen sie weiterhin Product Manager. Doch ihr Arbeitsalltag hatte nur noch bedingt Ähnlichkeit mit dem von vor zwei Monaten.

Am deutlichsten war vielleicht, was inzwischen niemanden mehr überraschte.

Ein Entwickler sprach mit einem Kunden. Ein anderer legte ein Mini PRD zur Verbesserung seines Moduls vor. Eine Vertriebsmitarbeiterin erstellte in der dafür vorgesehenen Umgebung eine neue Vorführungsvariante. Ein Mitarbeiter aus dem Support meldete unmittelbar einen Widerspruch in einer Produktregel.

Niemand begann mehr eine Grundsatzdiskussion über die Grenzen der Berufe.

Malorean hatte nicht verkündet, dass alle alles tun dürften. Das Unternehmen hatte lediglich aufgehört anzunehmen, dass jede Tätigkeit zwangsläufig über die Person laufen müsse, die sie traditionell ausübte.

Die beiden Teams mit jeweils fünf Entwicklern gab es weiterhin. Die meisten übernahmen inzwischen bestimmte Projekte von Anfang bis Ende: das Problem verstehen, einen Projektzuschnitt vorschlagen, mit den Betroffenen sprechen, entwickeln, testen und schließlich die Ergebnisse beobachten. Manche gewöhnten sich schneller daran als andere. Zwei Entwickler wollten bei der Eingrenzung ihrer Projekte weiterhin eng begleitet werden. Ein anderer war hervorragend bei komplizierten technischen Problemen, hatte aber überhaupt keine Lust auf zahlreiche Kundentermine. Niemand versuchte, aus ihnen gegen ihren Willen weitere Lucas zu machen.

Es gab auch weiterhin Tickets.

Sie dienten dazu, eine konkrete Tätigkeit zu koordinieren, eine Aufgabe festzuhalten oder eine Korrektur zu verfolgen. Aber niemand versuchte mehr, darin die ganze Begründung eines Projekts, sämtliche früheren Entscheidungen und jeden für einen Agenten nützlichen Hinweis unterzubringen.

Stattdessen pflegte jeder Projektverantwortliche einen eigenen Kontextbereich im gemeinsamen Dokumentationswerkzeug. Darin standen Ziele, Entscheidungen, Quellen, Risiken, offene Fragen und Links zu bereits vorhandenen Informationen: Tickets, Angebote, Supportberichte, Besprechungsnotizen, Tests und manchmal wiedergefundene alte Diskussionen.

Es war nicht nötig, alles zu kopieren. Man sollte vor allem verstehen können, warum eine Entscheidung getroffen worden war, wo sich eine Information überprüfen ließ und wer berechtigt war, die Entscheidung zu ändern.

Alices erstes Dokument diente als Vorbild.

Bob übernahm die Methode, dann die Entwickler und später mehrere Personen aus Support, Vertrieb und Implementierung. Viele entdeckten wie Alice, dass sie ChatGPT in einer Stunde mündlich Entscheidungen erklären konnten, deren schriftliche Ausarbeitung allein sie Tage gekostet hätte.

Nicht plötzlich war alles Wissen aus den Köpfen der Menschen zugänglich geworden. Doch ein Teil der Gründe, Ausnahmen

und stillschweigenden Erfahrungen wurde nach und nach sichtbar.

Die Fernarbeit hatte das erleichtert. Seit der Pandemie hatte Malorean ohnehin gelernt, mehr aufzuschreiben, damit Kollegen in ganz Frankreich nicht auf ein Gespräch auf dem Flur warten mussten. KI beschleunigte das Erfassen und Ordnen des Wissens. Die Notwendigkeit, es zu teilen, hatte sie nicht erfunden.

Alice öffnete eine Seite mit dem Titel *Produktentscheidungen — gemeinsame Regeln*.

— Letzte Woche gab es schon wieder ein Missverständnis, erklärte sie Bob. In der Dokumentation eines Teams bedeutete „freigegeben“ so viel wie „technisch geprüft“. Im Support bedeutete es „Produktverhalten mit dem Kunden bestätigt“.

Bob seufzte.

— Dieses Wort hat noch nie für alle dasselbe bedeutet.

— Genau. Ich habe auf die Entscheidung verwiesen, die wir nach der Chargengeschichte getroffen haben.

Einige Wochen zuvor hätten Lucas und Thomas beinahe zwei unvereinbare Verhaltensweisen in Rückverfolgbarkeit und Versand nebeneinander eingeführt. Die Änderungen waren beim Test gestoppt worden. Seither misstrauten die Teams Begriffen, die in einem Modul eindeutig schienen, in einem anderen aber etwas anderes bedeuteten.

Der geteilte Kontext half. Er schuf aber auch neue Probleme. Manche Seiten waren veraltet. Vorläufige Entscheidungen wurden kopiert, als wären sie endgültige Regeln. Einmal hatte ein Agent eine alte Vertriebsannahme hervorgeholt und als verbindliche Unternehmenszusage dargestellt.

Deshalb führten sie klare Regeln ein: Jeder Kontext benannte seinen Verantwortlichen, seine Quellen, bestätigte Entscheidungen und weiterhin unsichere Punkte. Grundlegende Änderungen mussten dokumentiert werden; sensible Informationen durften nicht unterschiedslos jedem Werkzeug zugänglich sein.

— Vielleicht hätten wir uns all diese Seiten sparen können, meinte Bob.

— Vielleicht. Aber vorher gab es die Widersprüche auch schon. Sie steckten nur in unseren Köpfen, und niemand konnte sie sehen.

Alice las die letzte Änderung ihres Kollegen.

— Den Grundsatz finde ich richtig. Aber nicht die Formulierung zu den deutschen Kunden: Dafür wissen wir noch zu wenig.

Bob korrigierte die Zeile.

So sah die Arbeit eines Garanten oft aus: nicht jede Bildschirmansicht genehmigen, sondern verhindern, dass eine Annahme unbemerkt zur Regel wurde.

Im anderen Team hatte Thomas gerade ein Gespräch mit einem neuen Entwickler beendet.
Eine Stunde lang hatte er ihm die Geschichte des ERP-Codes erläutert. Der neue Kollege hatte einen Assistenten benutzt, um einen alten Mechanismus der Lohnabrechnung zu verstehen. Dann hatte er gefragt, warum ein Sonderfall anscheinend der allgemeinen Regel widersprach.
Thomas fand den Grund für diese Ausnahme in einer früheren Auslieferung.
— Wenn du nur die Zusammenfassung liest, hältst du den Code vielleicht für aufräumbedürftig, erklärte er. Aber bevor wir etwas aufräumen, müssen wir verstehen, was wir entfernen würden.
— Kannst du das im Teamkontext festhalten?
— Genau das mache ich.
Thomas teilte eine neue Notiz. Er hatte die Quelle, die Erklärung und zwei Testfälle ergänzt.
Noch immer ließ er seine technischen Änderungen nicht von Agenten schreiben wie Lucas. Seine Skepsis war nicht verschwunden. Doch die neuen Dokumentationsbereiche nutzte er gern, und er arbeitete häufiger mit anderen zusammen, um seine Erfahrung weiterzugeben.
Vivien, der den Einstieg des neuen Entwicklers begleitet hatte, rief Thomas wenig später per Video an.
— Wie läuft’s?
— Gut. Er ist neugierig und fragt nach, bevor er etwas löscht.

— Das beruhigt mich.

— Und ich habe endlich aufgeschrieben, warum manche Prüfungen existieren.

Vivien lächelte.

— Du hast also deine Meinung über KI geändert?

— Nein. Ich habe meine Meinung darüber geändert, diese Erklärungen für mich zu behalten.

Er machte eine Pause.

— Nur braucht man dafür Zeit. Wenn ich anderen helfen soll und gleichzeitig sämtliche Tage mit Entwicklungsaufgaben vollgepackt sind, wird das nicht funktionieren.

— Ich habe mit Hubert gesprochen. Wir haben Zeit für die Begleitung anderer ausdrücklich in der Planung vorgesehen. Das soll keine Aufgabe sein, die erst erledigt wird, wenn alles andere fertig ist.

Vivien wusste genau, was er meinte. Jahrelang hatten Teams versprochen, Neueinsteiger einzuarbeiten, „sobald wieder etwas Zeit ist“. Dieser Zeitpunkt kam nie.

Sie sprachen noch über die Bedürfnisse des neuen Entwicklers, dann beendete Vivien die Videokonferenz. Seine Auslieferungsübersicht hatte er kein einziges Mal geöffnet.

Zur Mittagszeit bat Hubert Vivien um einen Anruf.

— Ich habe eine Frage, begann er. Wie viel Zeit verbringst du heute noch mit den Dingen, die du letztes Jahr getan hast?

Vivien dachte nach.

— Nicht genug, um meine alte Vollzeitstelle zu rechtfertigen, wenn du das wissen willst.

Hubert antwortete zunächst nicht.

Lange Zeit war Vivien die zentrale Anlaufstelle für zwei technische Teams gewesen. Er betreute die Auslieferungen, Abhängigkeiten, einen Teil der dringenden Aufgaben und den Austausch zwischen Entwicklern und Fachverantwortlichen. Als die Organisation weniger autonom gewesen war, hatte ihn das mehr als ausreichend beschäftigt.

Heute war das anders.

Für jedes Projekt gab es eine klar benannte verantwortliche Person. Entwickler trafen viele Entscheidungen vor Ort. Informationen zirkulierten, ohne dass Vivien sie weiterreichen musste. Die Werkzeuge erstellten Übersichten und wiesen auf Abhängigkeiten hin, auch wenn man ihre Aussagen weiterhin überprüfen musste.

Zwischenmenschliche Probleme, Konflikte und die Unterstützung anderer waren nicht verschwunden. Vivien half noch immer dabei. Doch dafür brauchte es keine Stelle mehr, deren Hauptaufgabe die tägliche Überwachung sämtlicher Auslieferungen war.

— Ich bin nicht traurig, dass ich weniger sinnlose Besprechungen habe, fuhr er fort. Im Gegenteil. Aber ich habe nicht jahrelang gelernt, Teams zu führen, um jetzt nur noch zu überprüfen, ob ein Agent seine Tabelle richtig ausgefüllt hat.

— Das werde ich auch nicht von dir verlangen, sagte Hubert.
Sie hatten einen Übergang vorbereitet. Wichtige technische Entscheidungen lagen bei den jeweiligen Garanten. Menschen, die andere begleiteten, erhielten ausdrücklich dafür reservierte Zeit. Hubert behielt die Verantwortung für Karriereentscheidungen und menschliche Schwierigkeiten, die nicht einfach einem Dokument überlassen werden konnten. Vivien sollte bei den letzten Koordinationsaufgaben die Übergabe organisieren.
Das funktionierte nicht immer perfekt. Einmal hatten zwei Personen jeweils geglaubt, die andere kümmere sich um einen Kollegen in Schwierigkeiten. Ein anderes Team hatte zu lange auf eine Entscheidung gewartet, weil niemand einen Verantwortlichen benannt hatte.
Jedes Mal klärten sie die Zuständigkeiten und verbesserten ihre Sichtbarkeit.
— Eines möchte ich auf keinen Fall, sagte Hubert. Deine Stelle abschaffen und so tun, als wäre deine bisherige Arbeit wertlos gewesen.
— Das möchte ich auch nicht.
— Dann müssen wir herausfinden, wo deine Erfahrung uns heute helfen kann.
Vivien betrachtete seine Tasse.
— Ich glaube, ich habe eine Idee.

Einige Tage zuvor hatte seine Frau wieder von Deutschland gesprochen.
Sie war Deutsche. Seit einiger Zeit diskutierten sie darüber, vielleicht dorthin zu ziehen. Es eilte nicht. Sie hatten ihr Leben in Frankreich, ihre Gewohnheiten und Freunde. Doch die Möglichkeit kam immer häufiger zur Sprache, zunehmend mit konkreten Details.
Vivien hatte das Thema stets aufgeschoben.
Er liebte seine Arbeit bei Malorean zu sehr. Er kannte die Teams, ihre Geschichte, die Software und die Kunden. Zu gehen hätte sich angefühlt, als würde er etwas verlassen, das er selbst mit aufgebaut hatte.
Andererseits mochte er Herausforderungen. In dieser Phase seiner Laufbahn ertappte er sich bei dem Wunsch, noch einmal etwas Neues zu wagen.
Am Samstag zuvor hatten sie sich Wohnungsanzeigen in Deutschland angesehen, ohne sich ganz einzugestehen, wie ernst das inzwischen war. Dann hatte seine Frau den Computer geschlossen.
— Weißt du, sagte sie, wir müssen nicht noch zehn Jahre darüber reden.
Vivien hatte gelacht.
Aber am Montag dachte er immer noch daran.

— Deutschland? wiederholte Hubert.
Vivien nickte.

— Seit Jahren sprechen wir in Strategiebesprechungen darüber. Wir haben dort ein paar Kunden, aber hauptsächlich Tochtergesellschaften französischer Konzerne. Wir haben noch nicht gelernt, unser ERP wirklich an deutsche Unternehmen zu verkaufen.

— Stimmt.

— Und wir sagen immer, dass wir es versuchen, sobald das Produkt bereit ist. Aber wir werden nie erfahren, was fehlt, solange wir nicht ernsthaft mit den Unternehmen vor Ort sprechen.

Hubert hörte aufmerksam zu.

— Du möchtest die Markterschließung übernehmen?

— Ja. Aber ich werde nicht so tun, als wäre ich über Nacht Experte für den deutschen Markt geworden. Ich kenne unser Produkt, seine Grenzen, unsere Teams und die Art, wie Kunden es einführen. Ich weiß, wie man Menschen für die Lösung eines komplizierten Problems zusammenbringt. Den Rest muss ich lernen.

— Hast du schon im Vertrieb gearbeitet?

— Nicht so wie Julie oder Philippe. Und ich will ihnen sicher nicht erklären, wie sie ihren Beruf machen sollen. Ich möchte mit ihnen zusammenarbeiten, mit unserem Implementierungsteam und den Fachleuten, um die Bedürfnisse deutscher Unternehmen zu verstehen und zu sehen, ob wir sie erfüllen können.

Vivien zögerte.

— Meine Frau ist Deutsche. Seit einiger Zeit reden wir darüber, dort zu leben. Das wäre die Gelegenheit, etwas zu machen, das mich wirklich interessiert, ohne Malorean zu verlassen.

Hubert schwieg eine Weile.

— Du weißt, dass es nicht reicht, die Benutzeroberfläche zu übersetzen?

— Natürlich. Ein ERP muss Regeln und Abläufe abbilden, die sich von Land zu Land unterscheiden. Die Beschriftungen auf Deutsch zu übersetzen genügt nicht. Wir müssen klären, was wir tatsächlich versprechen können und was wir entwickeln, anpassen oder ablehnen müssen.

— Wir könnten uns auch irren.

— Wahrscheinlich werden wir bei manchen Dingen falschliegen. Das möchte ich lieber in Gesprächen mit Kunden herausfinden, als nach einem Jahr Entwicklungsarbeit an Funktionen, die niemand braucht.

Hubert sah auf seine Projektliste.

Lange Zeit war Deutschland eine Ambition für später gewesen. Zuerst mussten Rückstände aufgearbeitet, weitere Entwickler eingestellt, Zeit für Anpassungen des ERP geschaffen, Vorführungen vorbereitet und Teams geschult werden. Manche Schwierigkeiten waren noch da. Doch einige hatten eine andere Größenordnung bekommen.

Eine Vorführung vorzubereiten kostete weniger Mühe. Produktvarianten ließen sich erproben, ohne jedes Mal die

Entwickler zu beanspruchen. Support und Presales erstellten Dokumente leichter und tauschten Wissen besser aus. Den Entwicklern fehlte seltener die Kapazität, begrenzte Ansätze zu untersuchen.

Das hieß nicht, dass ihnen die Arbeit ausgegangen war.

Es hieß, dass sie einen Teil ihrer Energie für andere Dinge einsetzen konnten, statt nur die aufgestauten Wünsche französischer Kunden abzuarbeiten.

— Mach mir einen Vorschlag für eine erste Phase, sagte Hubert. Keine Ankündigung einer Markteröffnung. Gespräche, die Fragen, die wir beantworten wollen, die Kompetenzen, die du brauchst, und einen Weg, um anschließend über das Weitergehen zu entscheiden.

Vivien lächelte.

— Ein Mini PRD für Deutschland?

— Mach dich nicht lustig. Wenn wir ein technisches Projekt so eingrenzen können, sollten wir wenigstens versuchen, die Annahmen eines Markterschließungsprojekts ebenso klar zu formulieren.

— Gut. Aber ich möchte mir das Recht bewahren herauszufinden, dass wir falschlagen.

— Darauf verlasse ich mich.

In der folgenden Woche versammelte Hubert Alice, Bob, Vivien, Julie und Philippe.

Er stellte den Vorschlag vor, ohne zunächst zu fragen, wie viele Funktionen dafür entwickelt werden müssten.

— Wir haben heute Möglichkeiten, die wir vor zwei Monaten noch nicht hatten. Ich möchte einen Teil davon nutzen, um herauszufinden, was eine echte Präsenz auf dem deutschen Markt für uns bedeuten würde.

Julie reagierte zuerst.

— Ich kann schon Vorführungen auf Deutsch vorbereiten. Wir könnten auch ziemlich schnell potenzielle Kunden kontaktieren. Mit KI lassen sich die Unterlagen in wenigen Stunden anpassen.

— Die Unterlagen vielleicht, sagte Philippe. Aber ein deutscher Geschäftsführer kauft nicht einfach ein ERP, weil die Präsentation in seiner Sprache ist.

Julie verschränkte die Arme.

— Das habe ich auch nicht gesagt.

— Weiß ich. Wir müssen aber verstehen, wie sie Lieferanten auswählen, wen sie einbeziehen und welche Art von Vertrauen nötig ist. Wir kennen den Markt noch zu wenig.

Bevor daraus eine Debatte über Werkzeuge wurde, mischte sich Vivien ein.

— Genau das möchte ich lernen. Julie, wir brauchen deine Fähigkeit, schnell Ansätze und Vorführungen vorzubereiten. Philippe, wir brauchen deine Erfahrung mit großen Kunden und den Gesprächen, die man nicht mit einer schönen Oberfläche entscheiden kann. Und unsere

Implementierungsspezialisten kennen die Fragen, die Kunden oft erst zu spät entdecken.

Alice meldete sich.

— Auf der Produktseite müssen wir die Grenzen sehr deutlich machen. Wir können schnell etwas ausprobieren, aber keine Funktionen zu gesetzlichen Anforderungen versprechen, die wir nicht geprüft haben.

Bob nickte.

— Und wir müssen uns entscheiden, welche Art von Unternehmen wir zuerst kennenlernen wollen. „Deutschland“ ist kein Kundentyp. Wenn wir versuchen, jeden zu bedienen, machen wir dieselben Fehler wie früher in Frankreich.

Hubert betrachtete die Gesichter auf dem Bildschirm.

Er hatte Vivien gebeten, einen Ansatz vorzuschlagen. Niemand hatte verlangt, sofort ein dauerhaftes Team aufzubauen, sämtliche Entwicklungskapazität zu reservieren oder einen Termin für den Markteintritt zu versprechen.

Und doch begannen die Beteiligten bereits, ihre möglichen Beiträge zu benennen.

Das wurde bei Malorean allmählich zur Gewohnheit.

— Gut, sagte Hubert. Ich entscheide Folgendes: Wir finanzieren eine erste Erkundungsphase. Vivien trägt die Verantwortung. Er soll herausfinden, welche Probleme deutscher Unternehmen wir schon lösen können, welche Anpassungen benötigen und welche wir bewusst nicht

übernehmen wollen. Danach entscheiden wir über weitere Investitionen.
Er fügte hinzu:
— Und niemand wird diesem Projekt allein deshalb zugeteilt, weil ein Werkzeug ihm Zeit gespart hat. Wir prüfen, welche Kapazität tatsächlich frei wird, ohne die heutigen Kunden im Stich zu lassen.
Vivien notierte den letzten Punkt. Er hatte lange genug Pläne erstellt, um zu wissen, dass auf dem Papier frei gewordene Zeit nicht unbedingt wirklich verfügbar war.

In den nächsten Tagen sprach er mit den Teams aus Presales und Implementierung.
Ein Teil ihrer Arbeit hatte sich verändert: Vorführungen ließen sich nun aus wiederverwendbaren Konfigurationen zusammenstellen, und Gespräche mit Kunden waren leichter zusammenzufassen. Zwei Personen erklärten ihm, dass sie nicht mehr so viel Zeit mit der Neuerstellung derselben Unterlagen verbrächten und gern an den ersten Gesprächen mit deutschen Unternehmen teilnehmen würden.
— Ich kann dir helfen, die Fragen zur Einführung vorzubereiten, schlug eine von ihnen vor. Und bei ein paar Videoterminen dabei sein.
— Und deine französischen Kunden? fragte Vivien.
— Alles stehen lassen kann ich nicht. Aber wenn wir es einplanen, kann ich etwas Zeit reservieren.

Eine andere Person war durch mehrere schwierige Einführungen voll ausgelastet und hatte überhaupt keine Zeit. Vivien drängte nicht.

Gemeinsam vereinbarten sie eine vorläufige Aufteilung: einige fest reservierte Zeitfenster für das Verständnis deutscher Kundenbedürfnisse, bei Bedarf Termine mit bestehenden Tochtergesellschaften als Kunden, aber keine Zusage zu einer dauerhaften Besetzung, bevor der tatsächliche Aufwand bekannt war.

Die gewonnene Zeit verwandelte sich nicht von allein in Wachstum. Jemand musste entscheiden, was man damit anfangen wollte – und bereit sein, die freigewordenen Stunden nicht sofort wieder mit den Anforderungen von gestern zu füllen.

Am Abend erhielt Alice eine Nachricht von Lucas.

Ich habe vom Deutschlandprojekt gehört. Unser Konfigurationsmechanismus könnte helfen, bestimmte Unterschiede in den Abläufen auszuprobieren. Ich könnte eine Testumgebung vorbereiten, ohne das Standardprodukt anzufassen. Interessiert euch das?

Alice las die Nachricht zweimal.

Zwei Monate zuvor hätte ein solcher Vorschlag vermutlich eine Diskussion darüber ausgelöst, ob ein Entwickler auf einen offiziellen Auftrag warten müsse. Jetzt ging es zuerst um

einfachere Fragen: War das strategisch sinnvoll und innerhalb welcher Grenzen durfte er handeln?

Alice öffnete den Kontext zur ersten deutschen Erkundungsphase. Vivien hatte dort bereits Annahmen, Verantwortliche, Grenzen und offene Fragen dokumentiert.

Sie antwortete:

Ja, wenn du das Projekt gemeinsam mit Vivien eingrenzt und wir eine geschäftliche Annahme nicht ohne Prüfung in eine Standardfunktion verwandeln. Beginne bei dem, was wir lernen wollen, nicht bei dem, was wir bauen können.

Lucas antwortete fast sofort.

Verstanden. Ich bereite ein Mini PRD vor.

Alice lächelte.

Sie erinnerte sich an jenen Freitag, als Lucas ihnen ein Urlaubsmodul gezeigt hatte, das er in zwei Tagen gebaut hatte. Damals wollten alle verstehen, wie er so viel so schnell schaffen konnte.

An diesem Abend überraschte es niemanden mehr, dass er rasch eine Vorführung entwickeln konnte.

Jetzt ging es darum herauszufinden, wozu man sie brauchte.

Vivien beendete den Tag, indem er seiner Frau die Termine der ersten Gespräche mit Unternehmen in Deutschland zeigte. Noch war kein Vertrag unterschrieben. Malorean hatte weder ein Büro eröffnet noch das ERP an sämtliche Anforderungen des Marktes angepasst. Sie mussten die Gewohnheiten vor Ort

kennenlernen, rechtliche und regulatorische Grenzen prüfen und herausfinden, wie man Unternehmen überzeugte, die Malorean nicht kannten.

Es gab weit mehr Unbekanntes als Gewissheiten.

— Macht dir das Angst? fragte seine Frau.

— Ein bisschen.

— Als du deine Teams koordiniert hast, warst du immer so sicher.

Vivien lächelte.

— Ich hatte jahrelang gelernt, wie dieser Beruf funktioniert.

— Und jetzt?

Er betrachtete das Dokument, das er für seinen ersten Besuch angelegt hatte.

Fragen. Annahmen. Namen von Menschen, mit denen er sprechen wollte. Eine erste Liste der Dinge, die er noch nicht wusste.

— Jetzt, antwortete er, habe ich wieder etwas zu lernen.

Sie trat näher an den Bildschirm.

— Und was ist mit uns?

— Jetzt können wir endlich ernsthaft über unseren Umzug sprechen.

Sie legte ihm eine Hand auf die Schulter.

Vivien dachte an seine alte Auslieferungsübersicht, die Montagsbesprechungen und all die Arbeit, mit der er Projekte zusammengehalten hatte. Er hatte sie gern gemacht. Er bereute diese Jahre nicht.

Und er bereute auch nicht, akzeptiert zu haben, dass die Teams ihn nicht mehr auf dieselbe Weise brauchten.
Auf dem Bildschirm stand die Bestätigung des ersten Termins in Deutschland.
Für Malorean war das noch keine Eroberung und kein Erfolg.
Es war ein Anfang.

12 — Schlusswort

Durchlässigere Berufsgrenzen, klarere Verantwortlichkeiten

Unsere Berufe legen nicht mehr genau fest, was jeder Einzelne tun kann. Ein Entwickler kann Kundenbedürfnisse untersuchen, ein PM selbst eine Lösung bauen und ein Vertriebsmitarbeiter einen Prototyp erstellen. KI öffnet Möglichkeiten, die früher anderen Fachgebieten vorbehalten waren. Das macht deren Kompetenzen aber nicht austauschbar.

Wir sind nicht mehr nur für das verantwortlich, was wir eigenhändig ausführen. Wir können ein Projekt übernehmen, das mehrere Funktionen berührt, während die Garanten die Kohärenz ihrer jeweiligen Fachbereiche sichern. Die Fähigkeit zu handeln verteilt sich; die Verantwortung darf sich dabei nicht auflösen.

Daraus ergeben sich zwei gegensätzliche Risiken: weiter zu sagen „Das ist nicht mein Beruf“ oder anzunehmen „Wenn ich es machen kann, darf ich auch allein entscheiden“. Weder das Verteidigen der alten Zuständigkeitsgrenzen noch ihre vollständige Auflösung reicht aus, um Zusammenarbeit zu organisieren.

Autonomie braucht einen gemeinsamen Kontext

Mehr Freiheit ist nur sinnvoll, wenn jeder die Informationen besitzt, die für ein fundiertes Urteil nötig sind. Ohne

gemeinsamen Kontext kann eine Entscheidung vor Ort richtig erscheinen und dennoch einem umfassenderen Bedarf, einer früheren Entscheidung oder einer Einschränkung widersprechen.

Dokumentation bedeutet deshalb nicht, Berichte anzuhäufen. Sie soll das Problem, das Ziel, die Annahmen, die Gründe für Entscheidungen, die Grenzen und die Eskalationsregeln zugänglich machen. Wenn diese Elemente ausdrücklich festgehalten werden, können Menschen handeln, ohne bei jedem Schritt die Person suchen zu müssen, die das nötige Wissen besitzt. Der Kontext wird zum Werkzeug für Entscheidungen – für Menschen und Agenten.

Auch die Garanten verändern dabei ihre Haltung. Sie kontrollieren nicht jede Initiative. Sie machen die Regeln verständlich, achten auf ihre Kohärenz und greifen ein, wenn eine Entscheidung den Rahmen überschreitet. Ohne Schutzmechanismen kann Freiheit im Chaos enden. Muss jedoch jede Entscheidung nach oben zur Freigabe, bleibt Autonomie ein leeres Versprechen.

Auch das Management entwickelt sich weiter. Ein Teil der Überwachung und Koordination kann anders erledigt werden. Doch Menschen müssen weiterhin begleitet, Erfahrung weitergegeben, Konflikte gelöst und schwierige Entscheidungen verantwortet werden. Diese Aufgaben verschwinden nicht mit den Tätigkeiten, die ihnen früher ihre Form verliehen haben.

Mehr von dem tun, was wirklich zählt

Die Versuchung wird groß sein, diesen Wandel an der Zahl entwickelter Funktionen oder eingesparter Stunden zu messen. KI kann uns tatsächlich mehr produzieren lassen. Doch wenn wir die gewonnene Zeit sofort mit neuen Aufgaben füllen, beschleunigen wir nur die bisherige Arbeitsweise, ohne nach ihrem Zweck zu fragen.

Wir können die neuen Möglichkeiten ebenso dazu nutzen, Bedürfnisse besser zu verstehen, Annahmen zu überprüfen, Bestehendes zu verbessern und auf Dinge zu verzichten, die keinen Wert schaffen. Nichts davon geschieht automatisch. Wir müssen entscheiden, worin wir die frei gewordene Aufmerksamkeit und die gewonnenen Mittel investieren.

Vivien bei Malorean Corp. entdeckt nicht nur, dass seine bisherige Arbeit weniger Zeit beansprucht. Er erkennt, dass er seine Erfahrung für ein anderes Problem einsetzen kann, ohne Erfolgsgarantie. Fortschritt bemisst sich nicht allein daran, was automatisiert wurde, sondern auch daran, was dieser Wandel möglich macht.

Am Anfang der Geschichte staunten alle darüber, wie schnell Lucas etwas entwickeln konnte. Am Ende lautet die Frage: Warum bauen wir etwas, für wen und mit welchen Folgen?

Die Herausforderung durch KI besteht nicht darin, uns immer mehr erledigen zu lassen. Sie besteht darin, uns die Möglichkeit zu geben, mehr von dem zu tun, was wirklich zählt.