WHITEPAPER

Governed Enterprise Vibe Coding

Wie Großunternehmen die Eigenentwicklung ihrer Fachbereiche ermöglichen und dabei steuerungsfähig bleiben

Für CIOs sowie Verantwortliche für Corporate IT und Enterprise Architecture Management
Juli 2026

RLC
ROMAN LANGFELD CONSULTING

„Wer versteht, entscheidet.
Wer entscheidet, gestaltet."

Executive Summary

Ihre Fachbereiche entwickeln bereits Software. Mit generativer KI entstehen dort in Stunden Anwendungen, für die früher ein Projektantrag und zwei Quartale nötig waren. Diese Lösungen funktionieren, sie sehen professionell aus, und sie erfüllen in der Regel keinen Ihrer Standards. Ein pauschales Verbot verlagert die Nutzung lediglich in den privaten Browser. Sie geschieht weiter, nun ohne **Sichtbarkeit**.

Die tragfähige Antwort ist eine **kuratierte Plattform mit eingebauten Leitplanken**: Der offizielle Weg muss schneller sein als der Umweg. Den Ausschlag für die Steuerung geben **Reichweite und Kritikalität** der jeweiligen Lösung. Ein vierstufiges Risikomodell regelt, was ein Fachbereich selbst bauen darf, was automatisch geprüft wird und wann die Verantwortung an ein professionelles Entwicklungsteam übergeht. Dieses Papier beschreibt das Modell, den zugehörigen Plattform-Blueprint und eine Einführung in vier Etappen über 24 Monate.

10 %	40 %	45 %	90 %
der Beschäftigten in DE nutzen KI ohne Wissen des Arbeitgebers	neuer Produktionssoftware sollen bis 2028 so entstehen	der KI-Codeproben bestehen Sicherheitstests nicht	der Organisationen haben eine interne Plattform
Bitkom	Gartner, Prognose	Veracode	DORA

Abbildung 1 · Ausgangslage in vier Kennzahlen. Belege siehe Quellen 1 bis 4.

1 Der Umweg ist immer schneller

In der Verwaltung mittelalterlicher Städte gab es einen wiederkehrenden Konflikt. Die Stadtplaner legten Wege an, die Bewohner liefen andere. Irgendwann setzte sich eine pragmatische Erkenntnis durch: Man baute die Wege dorthin, wo das Gras bereits niedergetreten war.

> Ihre Fachbereiche legen gerade eigene Trampelpfade an.

Ein Controller baut sich ein Auswertungswerkzeug. Im Vertrieb entsteht eine Automatisierung für die Angebotserstellung. Und aus einem Formular wird an einem Nachmittag ein kleines Portal. Die Ergebnisse sind funktionsfähig und optisch überzeugend. Geprüft hat sie niemand.

Dieses Phänomen trägt seit Februar 2025 einen Namen: **Vibe Coding**. Der KI-Forscher Andrej Karpathy prägte ihn für eine Arbeitsweise, bei der Code per Sprachbefehl entsteht und der Autor ihn nicht mehr liest.[5] Personen ohne IT-Ausbildung entwickeln seit Jahren Anwendungen. Der Bruch liegt an anderer Stelle: **Es prüft niemand mehr, was tatsächlich gebaut wurde.**

2 Warum Verbote das Gegenteil bewirken

Eine naheliegende Reaktion: untersagen, sperren, ein Formular vorschalten. Die Risiken sind schließlich real. Ein Sicherheitsscan von 5.600 mit KI erstellten Anwendungen fand über 400 offen liegende **Zugangsschlüssel** und

175 Fälle ungeschützter **personenbezogener Daten**, darunter medizinische Datensätze und Bankverbindungen.[6] Bei der Plattform Lovable generierte die KI systematisch Datenbankkonfigurationen ohne Zugriffsschutz; mehr als 170 Anwendungen standen offen.[7]

> **Ein Verbot verschiebt dieses Risiko nur dorthin, wo es niemand mehr sieht.**

Die Bitkom-Zahl von zehn Prozent ist dafür der Beleg. Die Ursache dürfte selten Trotz sein. Es gibt eine Aufgabe, ein Werkzeug und einen offiziellen Weg, der langsamer ist als der inoffizielle. Solange sich an diesem Verhältnis nichts ändert, wandert die Nutzung auf das private Gerät und in den privaten Account.

Hinzu kommt der verschenkte Produktivitätsgewinn. Der **DORA-Report** (Google Cloud, rund 5.000 Befragte) beschreibt KI-gestützte Entwicklung als **Verstärker**: Sie vergrößert bestehende Stärken ebenso wie bestehende Schwächen.[4] Organisationen mit einer guten internen Plattform gewinnen dabei. Ohne diese Grundlage beschleunigen sich vor allem die Probleme.

Die eigentliche Frage lautet daher: **Wer baut den Weg?**

3 Was hier tatsächlich neu ist

Für Citizen Development und Low-Code haben Sie vermutlich bereits Regeln. Ein Teil davon trägt weiterhin: Kompetenzzentren, Datenklassifikation, Abstufung nach Reichweite. Drei Unterschiede machen Vibe Coding jedoch zu einer anderen Kategorie.

Erstens entsteht echter Code. Eine Low-Code-Plattform hegt das Ergebnis in einer kontrollierten Laufzeitumgebung ein. Vibe Coding erzeugt vollwertige Anwendungen mit eigenen Abhängigkeiten, eigener Datenhaltung und eigenen Schnittstellen. **Es gibt keinen Rahmen, der den Fehler auffängt.**

Zweitens fehlt die Prüfinstanz. Ein Entwickler verwirft den Vorschlag der KI, wenn er nicht trägt. Der Fachanwender kann ihn kaum bewerten und übernimmt ihn. Halluzinierte Bibliotheken, veraltete Verschlüsselung, fehlende Rechteprüfung: All das sieht im Ergebnis identisch aus.

Drittens verlassen Daten das Unternehmen. Für eine Analyse muss das Modell mit genau den Daten gefüttert werden, die analysiert werden sollen. Bei einem externen Dienst ohne Vertrag entsteht daraus ein **Datenschutzvorfall**.

Tabelle 1 · Chancen und Risiken im Überblick

Dimension	Chance	Risiko
Geschwindigkeit	Prototyp innerhalb von Stunden, spürbare Entlastung des IT-Backlogs	Ungeprüfter Code erreicht den Produktivbetrieb
Datenzugang	Lösungen entstehen dort, wo das Fachwissen sitzt	Unternehmensdaten fließen an externe Modelle ab
Innovation	Teilweise Kompensation des Fachkräftemangels	Redundante Insellösungen, unbemerkte Doppelentwicklung
Sicherheit	Leitplanken wirken zentral für alle Lösungen zugleich	Ein erheblicher Teil des KI-Codes enthält bekannte Schwachstellen
Betrieb	Ergebnisse sind von Beginn an übernahmefähig	Einzellösungen, die niemand betreiben kann und niemand verantwortet

4 Reichweite und Kritikalität steuern das Risiko

Ein typisches Bild: Die Steuerung setzt bei den Werkzeugen an. Welches Modell ist erlaubt, welcher Assistent freigegeben? Diese Fragen haben ihre Berechtigung. Sie greifen nur zu kurz. Ein Wegwerf-Prototyp bleibt harmlos, auch mit dem stärksten Modell. Dieselbe Technik in einer kundenwirksamen Anwendung verändert die Lage vollständig.

Über das Risiko entscheiden vier Größen: Wer nutzt die Lösung, welche Daten verarbeitet sie, wie viel darf sie selbstständig auslösen, und was passiert bei einem Ausfall. Daraus ergibt sich ein Modell mit vier Stufen. Es beantwortet die Frage, wie viel Kontrolle im Einzelfall angemessen ist, und bildet die Grundlage für alles Weitere.

Tabelle 2 · Vier Risikoklassen, vier Freiheitsgrade

Klasse	Typische Lösung	Fachbereich baut selbst	Automatische Prüfung	Freigabe	Übergang an die IT
Niedrig	Persönlicher Prototyp, persönliche Arbeitshilfe	Ja, ohne Vorbedingung	Zugangsschlüssel-Prüfung	Keine	Nein
Mittel	Teamlösung, wenige Nutzer, keine sensiblen Daten	Ja, auf der Plattform	Abhängigkeiten, Lizenzen, Materialliste	Selbstregistrierung im Verzeichnis	Optional
Hoch	Abteilungsübergreifend oder unternehmensweit	Nur mit Begleitung durch die IT	Zusätzlich Codeanalyse und Architekturprüfung	Formelle Freigabe durch EAM	Bei weiterer Skalierung
Kritisch	Kundenwirksam, geschäftskritisch, regulatorisch relevant	Professionelle Entwicklung erforderlich	Zusätzlich Penetrationstest und Datenschutzfolgenabschätzung	Freigabe durch EAM, Informationssicherheit und Datenschutz	Verpflichtend

Vier Fälle gehören **grundsätzlich nicht** auf diesen Weg: Hochrisiko-KI im Sinne des EU AI Act, Eingriffe in Kernsysteme mit weitreichenden Schreibrechten, autonome Agenten ohne menschliche Kontrollinstanz und alles, was unmittelbar Kunden erreicht.

Der praktische Wert dieses Modells liegt in seiner Selbsterklärungskraft. Ein Fachbereich ordnet sich **in zwei Minuten selbst ein**. Genau das ist die Voraussetzung dafür, dass er es überhaupt tut.

5 Das Zielbild: die Umgebung um das gewohnte Werkzeug

Ein Portal, in das Fachbereiche erst gelockt werden müssen, verliert gegen das Werkzeug, das sie bereits geöffnet haben. Der produktive Weg führt durch diese Werkzeuge hindurch. Die Aufgabe der IT besteht darin, die Umgebung darum herum zu bauen.

Fünf Bausteine tragen dieses Zielbild. Sie sind in der Praxis erprobt und lassen sich weitgehend aus bestehenden Komponenten zusammensetzen.

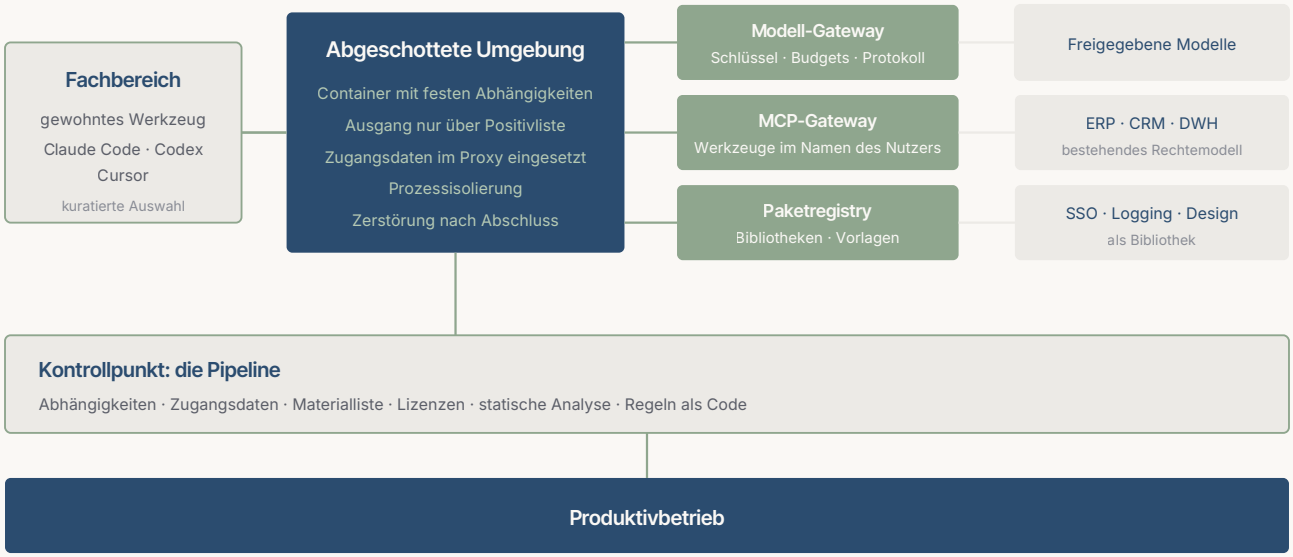

Abbildung 2 · Die Umgebung um das Werkzeug. Der Agent bleibt, wo der Fachbereich ihn gewohnt ist; kontrolliert wird der Weg nach außen und der Weg in die Produktion.

5.1 Der Werkzeugkasten: eine kuratierte Auswahl

Freigegeben werden ein bis zwei Agenten, etwa Claude Code, Codex CLI oder Cursor. Diese Begrenzung hat betriebswirtschaftliche Gründe. Lizenzverhandlung, Sicherheitsprüfung, Support, Schulung und Konfigurationspflege skalieren nur bei Fokussierung. Die Fachbereiche behalten damit ein Werkzeug, das sie kennen, und die IT behält eine Umgebung, die sie beherrscht.

5.2 Der Arbeitsplatz: eine abgeschottete Umgebung

Der Agent läuft in einem vorkonfigurierten Container. Der private Laptop scheidet damit als Ausführungsort aus. Vier Eigenschaften machen den Unterschied:

- **Ausgehender Verkehr über einen Proxy mit Positivliste.** Erreichbar sind nur freigegebene Ziele: die interne Paketregistry, das Modell-Gateway, die Versionsverwaltung. Alles andere fällt. In der Praxis wird das über Firewallregeln und einen Proxy im SNI-Peek-Verfahren gelöst, der die Zieladresse im TLS-Handshake liest, ohne die Verbindung aufzubrechen.
- **Zugangsdaten werden im Proxy eingesetzt.** Der Agent sieht Schlüssel und Passwörter nie. Damit läuft der häufigste Abflussweg ins Leere, nämlich Zugangsdaten in Umgebungsvariablen.
- **Prozessisolierung und automatische Zerstörung** nach Abschluss der Aufgabe. Kein Zugriff auf das Wirtssystem, keine verwaisten Umgebungen.
- **Reproduzierbarkeit.** Ein Abbild mit festgeschriebenen Abhängigkeiten, identisch für alle.

Der entscheidende Nebeneffekt betrifft die Zustimmungsermüdung. Wenn ein Agent bei jeder Aktion um Erlaubnis fragt, klicken Anwender reflexhaft auf Zustimmen, und die Kontrolle wird zur Fassade. Eine abgeschottete Umgebung nimmt diese Entscheidung ab und verlagert sie in die Konfiguration.

5.3 Der Modellzugang: ein Gateway als Vordertür

Alle Modellaufrufe laufen über eine zentrale Stelle. Das Gateway leistet, was klassische API-Verwaltung für Schnittstellen leistet: Identitätsprüfung, virtuelle Schlüssel je Team, Budgetgrenzen, Kostenzuordnung, Protokollierung jedes Aufrufs, Filterung personenbezogener Inhalte.

Der Gewinn liegt in der Vollständigkeit. Ohne diese Stelle ruft jedes Team seinen Anbieter direkt auf, jedes protokolliert anders, und niemand hat eine vollständige Spur. Mit ihr entstehen Kostentransparenz und Nachweisführung, ohne dass eine einzige Zeile in den Lösungen geändert werden muss.

5.4 Die Unternehmensbausteine: Bibliotheken und Werkzeuge

Hier entscheidet sich, ob die Ergebnisse übernahmefähig sind. Zwei Wege ergänzen einander.

Der erste Weg ist eine interne Paketregistry. Die IT veröffentlicht dort freigegebene Bibliotheken, die der Agent wie jedes andere Paket installiert:

Tabelle 3 · Freigegebene Bibliotheken und was sie dem Fachbereich abnehmen

Baustein	Was die Bibliothek abnimmt
Anmeldung und Identität	Anbindung an das Unternehmens-SSO, Sitzungsverwaltung, Abmeldung
Rollen und Berechtigungen	Prüfung gegen das bestehende Rechtemodell, Mandantentrennung
Protokollierung und Überwachung	Strukturierte Protokolle, Ablaufverfolgung, Anbindung an die Überwachung
Erscheinungsbild	Komponenten im Unternehmensdesign, Barrierefreiheit, Sprachdateien
Fehlerbehandlung	Einheitliche Fehlerbilder, Wiederholungslogik, Zeitgrenzen
Datenzugriff	Verbindungsaufbau, Verschlüsselung, Nachvollziehbarkeit von Zugriffen

Dazu kommen **Projektvorlagen**, die eine neue Lösung mit Verzeichnisstruktur, Prüfpipeline, Überwachung und Dokumentationsgerüst erzeugen. Der Agent beginnt damit auf einem Stand, den ein Fachanwender allein nie erreicht hätte.

Der zweite Weg sind Werkzeugzugänge zu den Kernsystemen. ERP, CRM und Datenprodukte werden über MCP-Server, Kommandozeilenwerkzeuge oder Schnittstellen bereitgestellt und hinter einem Gateway gebündelt. Drei Eigenschaften sind dabei wesentlich:

- **Der Agent erhält Werkzeuge, keine Zugangsdaten.** Die Anmeldung übernimmt das Gateway.
- **Handeln im Namen des Nutzers.** Der Agent arbeitet mit den Rechten der Person, die ihn startet, und nicht mit einer allmächtigen Systemidentität. Damit greift das bestehende Rechtemodell auch für KI-Zugriffe.
- **Werkzeug-Positivlisten und Bestätigungspflicht.** Lesende Zugriffe laufen durch, schreibende Operationen auf Kernsysteme brauchen eine menschliche Bestätigung.

5.5 Der Kontext: Konventionen im Verzeichnis

Agenten lesen Konventionsdateien im Projektverzeichnis, inzwischen weitgehend standardisiert als `AGENTS.md` und von über 60.000 öffentlichen Verzeichnissen verwendet.[10] Dort stehen Baubefehle, Prüfregeln und Randbedingungen, die sich aus dem Code allein nicht ableiten lassen.

Hier ist Zurückhaltung angebracht. Eine Untersuchung der ETH Zürich fand, dass solche Kontextdateien die Aufgabenlösung nicht verbesserten und den Aufwand teilweise erhöhten.[11] Die praktische Folgerung: Kurz halten, nur projektspezifische Randbedingungen aufnehmen, und alles Deterministische den Werkzeugen überlassen. Formatierung gehört in den Formatierer, Stilregeln in den Prüfer.

5.6 Der Kontrollpunkt bleibt die Pipeline

Welches Werkzeug ein Fachbereich verwendet, bleibt für die Prüfung ohne Bedeutung. Der Kontrollpunkt liegt beim Zusammenführen in die Versionsverwaltung: Abhängigkeitsprüfung, Suche nach Zugangsdaten, Materialliste, Lizenzprüfung, statische Analyse, Regeln als Code. Was diese Kette besteht, geht weiter. Was sie nicht besteht, bleibt liegen.

Eine Warnung zum Werkzeugprotokoll

Das MCP-Ökosystem ist selbst eine neue Angriffsfläche. Zwischen Januar und Februar 2026 wurden über 30 Schwachstellen an MCP-Servern und -Clients gemeldet; die schwerwiegendste betraf ein weit verbreitetes Proxy-Paket mit einem Schweregrad von 9,6 in über 437.000 Umgebungen.[12] Der zugrunde liegende Angriff heißt Werkzeugvergiftung: Ein Server liefert in seiner Beschreibung oder Antwort versteckte Anweisungen, die der Agent als vertrauenswürdige Eingabe behandelt. Die OWASP führt das seit Dezember 2025 als eigene Kategorie.[13]

Die Konsequenz für Ihre Architektur: ein internes Verzeichnis geprüfter MCP-Server, festgeschriebene Versionen, ein Abgleich der Werkzeugbeschreibungen gegen einen bekannten Stand, und keine schreibenden Operationen ohne Bestätigung.

6 Governance ohne Flaschenhals

Jede zentrale Freigabestelle wird zur Warteschlange, sobald die Zahl der Anfragen steigt. Das ist der klassische Konstruktionsfehler von Citizen-Development-Programmen: **Die Nachfrage wächst, die Prüfkapazität bleibt.**

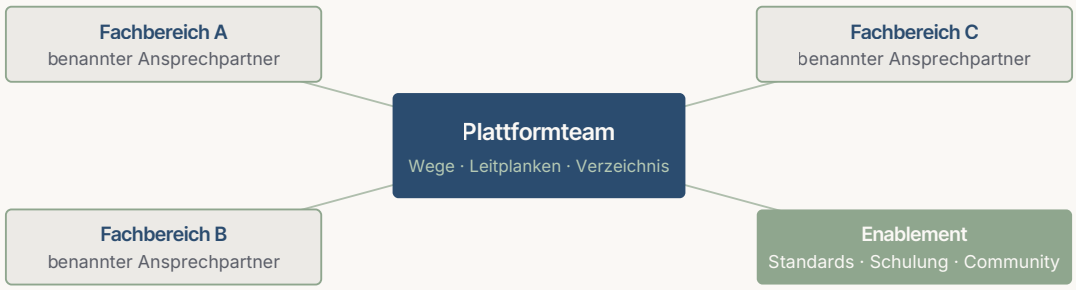

Abbildung 3 · Föderiertes Modell. Lokale Ansprechpartner lösen den Regelfall, das Zentrum entscheidet die Ausnahme.

Ein dokumentiertes Gegenmodell ist **föderiert**. Die Deutsche Bahn betreibt ihr Citizen-Developer-Programm mit einem zentralen Kompetenzzentrum für Standards und Bausteine sowie lokalen Kompetenzstellen in den Tochtergesellschaften.[8] Der zentrale Verantwortliche formuliert den Vorteil nüchtern: Er prüfe nur jene Fragen, die lokal ungelöst bleiben.

Übertragen auf Ihre Organisation bedeutet das drei Elemente: ein **zentrales Plattformteam**, das die Wege und Leitplanken baut und pflegt; ein **Enablement-Team** für Standards, Schulung und Community; und **benannte Ansprechpartner** in den Fachbereichen selbst. Enterprise Architecture verschiebt sich dabei von der nachträglichen Dokumentation zur vorausschauenden Definition maschinenprüfbarer Regeln. Corporate IT wird vom Torwächter zum Anbieter.

In Deutschland kommt ein rechtlicher Punkt hinzu. Die Einführung von KI-Werkzeugen unterliegt der **Mitbestimmung nach § 87 Abs. 1 Nr. 6 BetrVG**, sobald sie zur Verhaltens- oder Leistungsüberwachung geeignet sind. Die bloße Eignung genügt. Binden Sie die Arbeitnehmervertretung vor dem Rollout ein.

7 Sprache entscheidet über Akzeptanz

Eine Plattform, die nach Fachbegriffen fragt, wird von Fachbereichen kaum genutzt. Diese Übersetzungsleistung entscheidet darüber, ob die Plattform angenommen oder umgangen wird.

Tabelle 4 · Übersetzung technischer Konfiguration in fachliche Fragen

Statt dieser Frage	fragen Sie besser
Container-Deployment konfigurieren	Für wie viele Personen soll die Lösung verfügbar sein?
OAuth-Scopes festlegen	Auf welche Unternehmensdaten soll die Lösung zugreifen dürfen?
Row-Level-Security definieren	Wer darf welche Datensätze sehen?
Datenklassifikation vornehmen	Wären diese Daten ein Problem, wenn sie öffentlich würden?
Retention-Policy setzen	Wie lange soll die Lösung Daten aufbewahren?

8 Der Weg dorthin

Abbildung 4 · Einführung in vier Etappen.

Monat 0 bis 3: Sichtbarkeit herstellen. Rufen Sie Ihre Fachbereiche **sanktionsfrei** dazu auf, bestehende Lösungen zu melden. Diese Amnestie ist unbequem, aber sie liefert die Datenbasis, die sich sonst kaum beschaffen lässt. Parallel: Liste freigegebener Werkzeuge, die vier Risikoklassen, ein bis zwei Pilotbereiche.

Monat 3 bis 6: Den Weg öffnen. Ein Portal, **zwei bis drei** vorbereitete Standardwege, ein zentrales Verzeichnis aller Lösungen, automatisierte Sicherheits- und Lizenzprüfung, eine Community mit regelmäßigem Format.

Monat 6 bis 12: Anschluss an das Unternehmen. Freigegebene Schnittstellen und Datenprodukte integrieren, das föderierte Modell ausrollen, Regeln als Code ausbauen, den Übergabeprozess in den Betrieb standardisieren, Kennzahlen etablieren.

Monat 12 bis 24: Skalieren und automatisieren. Unternehmensweite Ausbreitung, vollständige Steuerung über den gesamten Lebenszyklus bis zur Stilllegung, Integration in Enterprise Architecture und Anwendungsportfolio, **risikobasierte Automatisierung** der Freigaben für die unteren Klassen.

Tabelle 5 · Woran Sie Fortschritt erkennen

Kennzahl	Aussage	Warnschwelle
Anteil registrierter Lösungen	Sichtbarkeit des Bestands	unter 60 %: Plattform zu umständlich
Anteil über die Plattform entwickelt	Akzeptanz des offiziellen Wegs	sinkend: der Umweg ist attraktiver
Zeit von der Idee zur produktiven Nutzung	Tatsächlicher Geschwindigkeitsgewinn	steigend: Governance-Überhang
Sicherheitsbefunde je Lösung	Wirksamkeit der Leitplanken	konstant: Leitplanken wirken nur beratend
Wiederverwendungsquote der Bausteine	Vermiedene Doppelentwicklung	niedrig: Bausteine sind schwer auffindbar
Übernahmeaufwand in Personentagen	Übernahmefähigkeit der Ergebnisse	hoch: Standards greifen zu spät
Zustimmungswert der Fachbereiche	Frühwarnung für neue Schatten-IT	fallend: Gegensteuern erforderlich

9 Fünf Fragen für Ihre nächste Leitungsrunde

Reifegrad-Check

1. Wissen Sie, **wie viele** mit KI erstellte Anwendungen in Ihrem Unternehmen produktiv laufen?
2. Können Ihre Fachbereiche **in zwei Minuten** selbst bestimmen, welcher Risikoklasse ihre Idee angehört?
3. Gibt es einen offiziellen Weg, der **schneller** ist als der Umweg?
4. Werden Ihre Sicherheitsregeln **automatisch durchgesetzt** oder in Dokumenten beschrieben?
5. Wissen Sie, **wer eine dieser Lösungen betreibt**, wenn ihr Erbauer das Unternehmen verlässt?

Bleiben drei dieser Fragen offen, lohnt der Blick auf drei Felder: die Wirtschaftlichkeit der einzelnen Lösung, die Befähigung der Fachbereiche und die Strukturen, die den Betrieb tragen sollen.

10 Schluss

Ein methodischer Hinweis, weil er zur Einordnung gehört: **Die Studienlage ist widersprüchlich.** Ein kontrollierter Versuch des Forschungsinstituts METR ergab, dass erfahrene Entwickler mit KI-Unterstützung 19 Prozent langsamer arbeiteten, sich dabei aber 20 Prozent schneller fühlten.[9] Andere Untersuchungen zeigen deutliche Gewinne. Die Wahrheit hängt am Kontext. Eine eindeutige Zahl zu diesem Thema sollte misstrauisch machen.

Aus beiden Befunden folgt dieselbe Managementkonsequenz: **Was eine KI erzeugt, ist ein Anfang. Das Ergebnis entsteht danach.**

Menschen nehmen den bequemsten Weg. Vorn liegen werden die Organisationen, die aus dieser Beobachtung eine **Bauaufgabe** gemacht haben.

> Das Gras ist bereits niedergetreten. Offen ist nur, wer den Weg pflastert.

Roman Langfeld
Consulting

Roman Langfeld

consulting@romanlangfeld.de
+49 (0)174 3228558
www.romanlangfeld.de

www.romanlangfeld.de

Quellen

1. Bitkom e. V., *Künstliche Intelligenz in Deutschland*, Berlin, September 2025. Befragung von 604 Unternehmen ab 20 Beschäftigten.
2. Gartner, *Why Vibe Coding Needs to Be Taken Seriously*, Mai 2025. Kostenpflichtige Publikation; die Zahl ist eine Prognose, keine Messung. Gartner unterhält wirtschaftliche Beziehungen zu Anbietern im betrachteten Markt.
3. Veracode, *2025 GenAI Code Security Report*, 2025. Untersuchung von über 100 Sprachmodellen. Der Herausgeber vertreibt Sicherheitsprüfwerkzeuge.
4. Google Cloud / DORA, *State of AI-assisted Software Development*, 2025. Rund 5.000 Befragte.
5. Andrej Karpathy, Beitrag auf X, 2. Februar 2025.
6. Escape.tech, Sicherheitsuntersuchung von 5.600 mit KI erstellten Anwendungen, Oktober 2025.
7. CVE-2025-48757, National Vulnerability Database.
8. Microsoft Learn, *Deutsche Bahn empowers citizen developers with Power Platform*, Fallstudie.
9. METR, *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity*, Juli 2025. Randomisierter Versuch mit 16 Entwicklern; die Übertragbarkeit auf Fachanwender ist begrenzt.
10. AGENTS.md, offene Konvention für Kontextdateien, seit 2025 unter dem Dach der Agentic AI Foundation (Linux Foundation).
11. ETH Zürich und LogicStar.ai, *Evaluating AGENTS.md: Are Repository-Level Context Files Helpful for Coding Agents?*, 2026.
12. Cloud Security Alliance, *Agentic MCP Security Best Practices*, Mai 2026; CVE-2025-6514.
13. OWASP GenAI Security Project, *Top 10 for Agentic Applications*, Dezember 2025, Kategorien ASI01 und ASI02.