Themenreihe KI-native Entwicklung · Teil 5 von 5 · Teil 1: Was ist KI-native Softwareentwicklung? · Teil 2: Vom KI-Prototyp zur Produktion · Teil 3: KI-generierten Code prüfen · Teil 4: Souveräne KI-Entwicklung

KI-native Entwicklung: Wo die Kosten sinken und wo sie bleiben

Von **Thomas Faß** · 30. Oktober 2026 · aktualisiert 12. November 2026

> **KI-native Entwicklung senkt vor allem die Kosten des Bauens. Anforderungsklärung, Prüfung, Betrieb und Weiterentwicklung bleiben überwiegend menschliche Arbeit, und dort entsteht über die Lebensdauer einer Software der größere Teil der Kosten. In randomisierten Feldexperimenten erledigten Entwicklerinnen und Entwickler mit einem KI-Assistenten rund 26 Prozent mehr Aufgaben (Cui et al., Management Science). Wirtschaftlich lohnt sich der Ansatz deshalb am meisten, wo viel gebaut werden muss und die Anforderungen klar sind, und am wenigsten, wo Unklarheit oder Regulierung den Aufwand bestimmen.**

Das Wichtigste in Kürze

- Die Baukosten sinken messbar. In drei randomisierten Feldexperimenten mit 4.867 Entwicklerinnen und Entwicklern stieg die Zahl abgeschlossener Aufgaben mit einem KI-Assistenten um rund 26 Prozent (Cui et al., Management Science).
- Der Effekt ist nicht überall gleich. Erfahrene Open-Source-Entwickler waren in einer Studie von 2025 mit KI 19 Prozent langsamer. Eine Folgestudie von 2026 deutet auf einen Zeitgewinn hin, die Autoren selbst nennen die Datenlage aber schwach (METR).
- Die Stabilität leidet, wenn die Prüfung nicht mitwächst. Der DORA-Report 2025 findet bei stärkerer KI-Nutzung mehr Durchsatz und zugleich weniger stabile Auslieferungen.
- Ein großer Teil der Kosten liegt nach dem Bau. Ältere Erhebungen beziffern den Anteil von Wartung und Weiterentwicklung an den Softwarekosten auf 40 bis 80 Prozent (Glass, 2002). Daran ändert schnelleres Bauen wenig.
- Die Kostentreiber liegen selten beim Modell. Unternehmen, die KI nutzen, nennen Infrastruktur (51 Prozent), Datenaufbereitung (50 Prozent) und Integration (41 Prozent) als große Kostenpunkte, Token-Kosten nur 8 Prozent (Bitkom 2026).
- Das Vertragsmodell entscheidet, wer das Risiko trägt. Festpreis, Aufwand oder ein Hybrid aus beidem passen zu unterschiedlich klaren Anforderungen.
- Standardsoftware bleibt oft die bessere Wahl. Wo ein Prozess kein Unterscheidungsmerkmal ist, verschiebt KI die Rechnung selten genug.

Wo sinken die Kosten durch KI-native Entwicklung?

Die Kosten sinken dort, wo Code entsteht. Die belastbarste Evidenz liefern drei randomisierte Feldexperimente bei Microsoft, Accenture und einem weiteren Großunternehmen mit zusammen 4.867 Entwicklerinnen und Entwicklern. Mit einem KI-Assistenten stieg die Zahl abgeschlossener Aufgaben, gemessen an Pull Requests, um rund 26 Prozent [1]. Bei weniger erfahrenen Entwicklern fiel der Zuwachs höher aus als bei erfahrenen, statistisch gesichert ist dieser Unterschied nicht [1].

Eine Auswertung bei Microsoft aus dem Jahr 2026 kommt mit einem anderen Verfahren auf eine ähnliche Größenordnung: rund 24 Prozent mehr zusammengeführte Pull Requests je Person und Tag nach Einführung agentischer Coding-Werkzeuge [2]. Die Autoren arbeiten beim Hersteller und weisen selbst darauf hin, dass ein Pull Request noch nichts über den gelieferten Wert sagt [2].

Ein Gegenbefund gehört dazu. In einer randomisierten Studie von METR brauchten 16 erfahrene Open-Source-Entwickler in großen, vertrauten Codebasen mit KI 19 Prozent länger als ohne [3]. Eine Folgestudie mit 57 Entwicklern deutet 2026 auf einen Zeitgewinn hin. METR selbst nennt das sehr schwache Evidenz, weil ein Teil der Teilnehmenden Aufgaben zurückhielt, die sie nicht ohne KI erledigen wollten [4].

Zusammen ergibt das ein klares Bild: Der Gewinn entsteht beim Schreiben von Code, er ist real, und seine Größe hängt stark von Aufgabe, Codebasis und Erfahrung ab. Eine pauschale Faktor-Zahl lässt sich daraus nicht ableiten. Wer sie in einem Angebot liest, sollte nach der Messmethode fragen.

Welche Kostentreiber bleiben?

Anforderungsklärung. Ein Agent baut schnell, was beschrieben ist. Ist es falsch beschrieben, baut er schnell das Falsche. Die Zeit, in der Fachbereich und Entwicklung klären, was gebraucht wird, sinkt durch KI kaum. Sie wird wichtiger, weil jede unklare Anforderung jetzt schneller zu Code wird, der später geändert werden muss.

Prüfung. Mehr Code heißt mehr zu prüfen. In der Stack-Overflow-Umfrage 2025 nannten 66 Prozent der Befragten KI-Lösungen, die „fast richtig, aber nicht ganz“ sind, als größte Frustration, 45 Prozent sagen, das Debuggen von KI-Code koste mehr Zeit [5]. Nur 3 Prozent vertrauen der Genauigkeit von KI-Ausgaben stark [5]. Der DORA-Report 2025 beschreibt KI als Verstärker: Mehr Durchsatz geht mit weniger stabilen Auslieferungen einher, wenn Prozesse und Prüfung nicht mitziehen [6]. Wie diese Prüfung aufgebaut sein kann, beschreibt Teil 3 dieser Reihe.

Codequalität über die Zeit. Eine Studie der Carnegie Mellon University findet in Open-Source-Projekten, die Coding-Agenten einsetzen, dauerhaft mehr Warnungen der statischen Codeanalyse (rund 18 Prozent) und höhere kognitive Komplexität (rund 39 Prozent) [7]. Komplexerer Code ist teurer zu ändern. Das zeigt sich nicht im ersten Projektmonat, wohl aber im dritten Jahr.

Betrieb und Weiterentwicklung. Der größte Kostenblock einer Individualsoftware liegt traditionell nach dem Bau. Ältere Erhebungen beziffern den Anteil von Wartung an den Softwarekosten auf 40 bis 80 Prozent, im Mittel 60 Prozent [8]. Diese Zahlen sind Jahrzehnte alt und fassen unter Wartung vor allem Erweiterungen, also neue Funktionen für ein bestehendes System. Die Richtung ist trotzdem stabil: Wer nur die Baukosten vergleicht, vergleicht den kleineren Teil.

Kompetenz im eigenen Haus. In einer randomisierten Studie schnitten Nachwuchsentwickler, die eine neue Bibliothek mit KI-Hilfe lernten, im anschließenden Wissenstest deutlich schlechter ab als die Vergleichsgruppe ohne KI (50 gegenüber 67 Prozent), ohne signifikant schneller zu sein [9]. Für die Wirtschaftlichkeit heißt das: Wer das System später selbst betreiben will, muss den Kompetenzaufbau einplanen.

Daten und Integration. Unternehmen, die KI bereits nutzen, nennen als große Kostenpunkte vor allem Infrastruktur (51 Prozent), Datenaufbereitung (50 Prozent) und Integration in bestehende Systeme (41 Prozent). Token-Kosten nennen nur 8 Prozent [10]. Die Preise für Modellaufrufe auf gleichem Leistungsniveau sind seit 2021 je nach Maßstab um den Faktor 9 bis 900 pro Jahr gefallen [11]. Das Modell ist selten der teure Teil.

Was zeigt der Blick auf den Lebenszyklus?

Ordnet man die Befunde den Phasen einer Individualsoftware zu, verschiebt KI-native Entwicklung die Kosten ungleich:

Phase	Wirkung von KI-nativer Entwicklung	Belegt durch
Anforderungsklärung	kaum geringer, Fehler werden schneller teuer	qualitative Einordnung
Bau	deutlich geringer, stark abhängig von Aufgabe und Codebasis	Feldexperimente [1], Microsoft [2], METR [3][4]
Prüfung und Qualitätssicherung	eher höher, mehr Code und mehr Nacharbeit	Stack Overflow [5], DORA [6]
Betrieb	kaum verändert, Infrastruktur und Integration bleiben	Bitkom [10]
Wartung und Weiterentwicklung	offen, Risiko steigender Komplexität	CMU [7], Glass [8]

Prozentwerte je Phase gibt es dafür nicht. Jede Zahl, die das behauptet, beruht auf Annahmen. Die Tabelle zeigt deshalb nur die Richtung. Das Ausmaß hängt vom einzelnen Vorhaben ab.

Festpreis, Aufwand oder Hybrid: Welches Vertragsmodell passt?

Das Vertragsmodell bestimmt, wer das Risiko unklarer Anforderungen trägt. Diese Einordnung ist keine Rechtsberatung.

Festpreis (Werkvertrag). Geschuldet ist ein Ergebnis, das der Auftraggeber abnimmt (§ 631 BGB) [12]. Der Auftragnehmer trägt das Mengenrisiko. Das passt, wenn die Anforderungen vollständig und stabil beschrieben sind. Mit KI-nativer Entwicklung kann ein Festpreis attraktiver werden, weil der Bau schneller

kalkulierbar ist. Er bleibt riskant, wenn sich Anforderungen während des Projekts ändern, denn jede Änderung wird zur Nachtragsverhandlung.

Aufwand (Dienstvertrag). Geschuldet ist die Tätigkeit, kein Erfolg (§ 611 BGB) [13]. Der Auftraggeber trägt das Mengenrisiko und behält die volle Flexibilität. Das passt zu explorativen Vorhaben, deren Ziel sich erst im Projekt schärft. Ein Geschwindigkeitsgewinn durch KI kommt hier direkt beim Auftraggeber an, sofern der Auftragnehmer ihn nicht in längeren Stunden aufbraucht.

Hybrid. Viele agile Projekte kombinieren beides: Festpreise je abgegrenztem Arbeitspaket oder Sprint, Kündigungsmöglichkeiten zum Sprint-Ende, Aufwand für die laufende Weiterentwicklung. In der Schweiz entfällt bei Softwarefirmen der größte Umsatzanteil auf die Abrechnung nach Aufwand mit Ergebnisverantwortung (42 Prozent) [14]. Eine gefestigte höchstrichterliche Linie dazu, wie agile Verträge rechtlich einzuordnen sind, gibt es nach unserer Recherche in Deutschland nicht. Umso wichtiger sind klare Regeln zu Abnahme, Änderungen und Ausstieg im Vertrag selbst.

Für alle drei Modelle gilt eine Frage, die mit KI neu dazukommt: Wer prüft und verantwortet den generierten Code, und wie ist diese Prüfung im Angebot kalkuliert? Ein Angebot, das nur den Bau rechnet, ist unvollständig.

Wann ist Standardsoftware die bessere Wahl?

KI senkt die Kosten des Bauens. Die Kosten des Eigentums bleiben: Wer Individualsoftware baut, betreibt, prüft und entwickelt sie über Jahre weiter. Standardsoftware teilt diese Kosten mit allen anderen Kunden des Herstellers.

Standardsoftware ist meist die bessere Wahl, wenn

- der Prozess in vielen Unternehmen gleich abläuft, etwa Buchhaltung, Lohn oder Zeiterfassung,
- der Prozess kein Unterscheidungsmerkmal im Wettbewerb ist,
- regulatorische Änderungen häufig sind und der Hersteller sie für alle Kunden nachzieht,
- im eigenen Haus niemand die Software dauerhaft verantworten kann.

Individualsoftware lohnt sich eher, wenn

- der Prozess das Geschäft vom Wettbewerb unterscheidet,
- Standardsoftware nur mit hohem Anpassungsaufwand passt,
- Daten oder Abläufe so speziell sind, dass Integration teurer würde als ein eigener Bau,
- Anforderungen klar sind und jemand im Haus die fachliche Verantwortung übernimmt.

Wo KI-native Entwicklung die Grenze verschiebt, dann vor allem bei der dritten Gruppe: Ein eigener Bau, der früher zu teuer war, kann sich heute rechnen, wenn die Folgekosten mitkalkuliert sind. Am anderen Ende steht ein Befund des IAB: Von den Betrieben, die KI weder nutzen noch planen, sehen 88 Prozent keine passende Anwendungsmöglichkeit [15]. Auch das ist eine legitime wirtschaftliche Entscheidung.

Wie sieht das an einem konkreten Vorhaben aus?

Das Beispiel aus den Teilen 2 bis 4 geht zu Ende. Unternehmen, Personen und Details sind konstruiert, Messwerte enthält das Beispiel nicht.

Die Ausgangslage. Der Großhändler für Sanitärtechnik hat seine Angebots-App mit KI-nativer Entwicklung gebaut und um die Erkennung von Freitext-Anfragen erweitert. Jetzt will die Geschäftsführung wissen, ob sich das gerechnet hat und ob als Nächstes auch die Lagerverwaltung neu gebaut werden soll.

Die Angebots-App. Der Bau ging schneller als bei früheren Projekten. Die eingesparte Zeit floss zum Teil in die Prüfung, in den Aufbau der Testdaten und in die Nachkontrolle bei handschriftlichen Listen. Den größten Aufwand verursachte die Abstimmung mit dem Innendienst darüber, was ein „brauchbarer Angebotsentwurf“ ist. Die App unterscheidet den Großhändler von Wettbewerbern, weil Kunden schneller ein Angebot bekommen. Hier hat sich Individualsoftware gelohnt.

Die Lagerverwaltung. Die IT-Leitung sieht sich die Abläufe an. Wareneingang, Einlagerung und Inventur laufen wie in jedem anderen Großhandel. Es gibt mehrere Standardlösungen, die das abdecken und regelmäßig aktualisiert werden. Ein Neubau wäre mit KI schneller als früher, aber der Betrieb, die Prüfung und jede künftige Anpassung blieben dauerhaft im eigenen Haus. Die Entscheidung fällt für Standardsoftware, mit einer schmalen, selbst gebauten Schnittstelle zur Angebots-App.

Das Ergebnis. Dieselbe Technik, zwei verschiedene Entscheidungen. Den Ausschlag gab beide Male die Frage, ob der Prozess ein Unterscheidungsmerkmal ist und wer die Software nach dem Bau trägt.

Fünf Fragen vor dem Start

1. **Sind die Anforderungen klar genug, um sie einem Agenten zu beschreiben?** Wenn nicht, beginnt das Projekt mit Klärung. Code kommt danach.
2. **Unterscheidet der Prozess Ihr Unternehmen vom Wettbewerb?** Wenn nicht, prüfen Sie zuerst Standardsoftware.
3. **Wer prüft den generierten Code, und ist diese Prüfung im Budget?** Ohne Prüfung wird schneller Bau zu teurer Wartung.
4. **Wer betreibt und entwickelt das System in drei Jahren weiter?** Rechnen Sie die Kosten über die gesamte Lebensdauer. Die Abnahme ist erst der Anfang.
5. **Welches Vertragsmodell passt zur Klarheit Ihrer Anforderungen?** Festpreis für Klares, Aufwand für Exploratives, Hybrid für alles dazwischen.

Was diese Reihe gezeigt hat

Fünf Teile, ein roter Faden: KI-native Entwicklung verschiebt Arbeit vom Schreiben zum Entscheiden. Teil 1 hat geklärt, was KI-native Softwareentwicklung ist und wie sie sich von Copilot-Nutzung und Vibe Coding unterscheidet. Teil 2 hat gezeigt, was zwischen einem KI-Prototyp und dem Betrieb liegt. Teil 3 hat beschrieben, wie man KI-generierten Code prüft. Teil 4 hat erklärt, warum Souveränität eine Frage der Architektur ist. Teil 5 rechnet nach: Der Bau wird günstiger, die Verantwortung bleibt.

Alle Teile und weitere Fachartikel finden Sie unter Einblicke.

Ihr nächster Schritt

Sie stehen vor der Entscheidung, ob ein Vorhaben selbst gebaut oder eingekauft werden soll? Sprechen Sie mit uns.

Häufige Fragen

Wird Individualsoftware durch KI günstiger als Standardsoftware?

Der Bau wird günstiger, die Kosten über die Lebensdauer deutlich weniger. Betrieb, Prüfung und Weiterentwicklung bleiben beim Eigentümer der Software. Standardsoftware teilt diese Kosten mit allen Kunden des Herstellers. Für Prozesse, die in vielen Unternehmen gleich ablaufen, bleibt sie deshalb meist günstiger.

Wie viel schneller ist KI-native Entwicklung?

Eine seriöse Pauschalzahl gibt es nicht. Randomisierte Feldexperimente zeigen rund 26 Prozent mehr abgeschlossene Aufgaben mit einem KI-Assistenten. Eine Studie mit erfahrenen Entwicklern in vertrauten Codebasen fand dagegen eine Verlangsamung. Die Wirkung hängt von Aufgabe, Codebasis und Erfahrung ab.

Was kostet die Wartung von KI-generiertem Code?

Belastbare Langzeitzahlen gibt es noch nicht. Erste Studien zeigen in Projekten mit Coding-Agenten mehr Warnungen der statischen Analyse und höhere Komplexität. Beides macht Änderungen teurer. Feste Prüfstufen und Tests senken dieses Risiko.

Welches Vertragsmodell passt zu KI-nativer Entwicklung?

Das hängt von der Klarheit der Anforderungen ab. Ein Festpreis passt zu vollständig beschriebenen Vorhaben, Aufwand zu explorativen, ein Hybrid aus Festpreisen je Arbeitspaket und Aufwand für die Weiterentwicklung zu allem dazwischen. In jedem Fall sollte das Angebot die Prüfung des generierten Codes ausweisen.

Wie vergleicht man Angebote seriös?

Achten Sie darauf, ob neben dem Bau auch Anforderungsklärung, Prüfung, Tests und Übergabe kalkuliert sind. Fragen Sie nach der Methode hinter jeder Zahl zur Geschwindigkeit. Und vergleichen Sie die erwarteten Kosten über mehrere Jahre. Der Preis bis zur Abnahme ist nur ein Teil davon.

Sind die Kosten für KI-Modelle ein großer Posten?

Meist nicht. In einer Bitkom-Befragung nennen nur 8 Prozent der Unternehmen, die KI nutzen, Token-Kosten als großen Kostenpunkt. Infrastruktur, Datenaufbereitung und Integration liegen weit davor.

Lohnt sich KI-native Entwicklung auch für kleine Vorhaben?

Oft ja, weil kleine, klar beschriebene Anwendungen besonders vom schnelleren Bau profitieren. Die Folgekosten bleiben aber auch bei kleinen Anwendungen. Wer sie nicht tragen will, sollte zuerst nach einer Standardlösung suchen.

Thematische Vertiefungen

KI-native Softwareentwicklung für den Mittelstand

Was KI in der Softwareentwicklung wirklich beschleunigt

Quellen

[1] Cui, Z. K., Demirer, M., Jaffe, S., Musolff, L., Peng, S., Salz, T.: *The Effects of Generative AI on High-Skilled Work: Evidence from Three Field Experiments with Software Developers* – *https://doi.org/10.1287/mnsc.2025.00535*. Management Science, 2026 (Arbeitspapier Februar 2025). Drei randomisierte Feldexperimente, 4.867 Entwicklerinnen und Entwickler.

[2] Murphy-Hill, E., Butler, J., Savelieva, A.: *Adoption and Impact of Command-Line AI Coding Agents: A Study of Microsoft's Early 2026 Rollout* – *https://arxiv.org/abs/2607.01418*. arXiv 2607.01418, 01.07.2026. Synthetische Kontrollgruppe, Autoren bei Microsoft angestellt, nicht begutachtet.

[3] METR: *Measuring the Impact of Early-2025 AI on Experienced Open-Source Developer Productivity* – *https://metr.org/blog/2025-07-10-early-2025-ai-experienced-os-dev-study/*. 10.07.2025. Randomisierte Studie, 16 Entwickler, 246 Aufgaben.

[4] METR: *We are Changing our Developer Productivity Experiment Design* – *https://metr.org/blog/2026-02-24-uplift-update/*. 24.02.2026. 57 Entwickler, über 800 Aufgaben.

[5] Stack Overflow: *2025 Developer Survey: AI* – *https://survey.stackoverflow.co/2025/ai*. Online-Befragung, 49.009 Antworten, 29.05.–23.06.2025, nicht repräsentativ.

[6] Google Cloud: *DORA State of AI-assisted Software Development 2025* – *https://dora.dev/research/2025/dora-report/*. September 2025. Befragung von knapp 5.000 Fachleuten.

[7] Agarwal, S., He, H., Vasilescu, B.: *AI IDEs or Autonomous Agents? Measuring the Impact of Coding Agents on Software Development* – *https://arxiv.org/abs/2601.13597*. Carnegie Mellon University, arXiv 2601.13597, Januar 2026.

[8] Glass, R. L.: *Facts and Fallacies of Software Engineering*. Addison-Wesley, 2002, Fact 41. Zusammenstellung älterer Erhebungen: Koskinen, J.: *Software Maintenance Costs* – *https://ocw.unican.es/pluginfile.php/2171/course/section/2000/SMCOSTS.pdf*. University of Eastern Finland, 2015.

[9] Shen, J. H., Tamkin, A.: *How AI Impacts Skill Formation* – *https://arxiv.org/abs/2601.20245*. Anthropic, arXiv 2601.20245, 28.01.2026. Randomisierte Studie, 52 Teilnehmende.

[10] Bitkom: *Erstmals nutzt die Mehrheit der Unternehmen KI* – *https://www.bitkom.org/Presse/Presseinformation/Erstmals-nutzt-Mehrheit-Unternehmen-KI*. Presseinformation, 14.09.2026. 603 Unternehmen ab 20 Beschäftigten, telefonische Befragung.

[11] Epoch AI: *LLM inference prices have fallen rapidly but unequally across tasks – https://epoch.ai/data-insights/llm-inference-price-trends*. 12.03.2025.

[12] *§ 631 BGB, Vertragstypische Pflichten beim Werkvertrag – https://www.gesetze-im-internet.de/bgb/__631.html*. gesetze-im-internet.de.

[13] *§ 611 BGB, Vertragstypische Pflichten beim Dienstvertrag – https://www.gesetze-im-internet.de/bgb/__611.html*. gesetze-im-internet.de.

[14] Swico, Universität Bern: *Swiss Software Industry Survey 2026 – https://www.swico.ch/media/filer_public/3d/57/3d57aa05-9a2c-4d8e-955d-0332241a0655/report_2026_final.pdf*. Oktober 2026. Online-Befragung Schweizer Softwarefirmen, je Frage 52 bis 115 Antworten.

[15] Kagerl, C.: *Weshalb die meisten Betriebe Künstliche Intelligenz (noch) nicht nutzen – https://iab-forum.de/weshalb-die-meisten-betriebe-kuenstliche-intelligenz-noch-nicht-nutzen/*. IAB-Forum, 20.10.2026. IAB-Betriebspanel 2025.

Über den Autor

Thomas Faß ist Gründer und CEO von InnoShore, einem IT-Dienstleister für den DACH-Mittelstand, und leitet die Regionalgruppe NRW im KI Bundesverband.