

Die Token-Steuer: Warum autonome KI-Agenten die Margen schmälern, die sie eigentlich schützen sollten

Ein Leitfaden für COOs, CXOs
und das operative Management



Die Chance ist real

KI hat die Software-Entwicklung grundlegend verändert, die heute schneller, zugänglicher und leistungstärker als je zuvor ist.

Entwickler können in einem Tempo neue Software programmieren und bereitstellen, das noch vor einem Jahr unmöglich schien.

84 % der Entwickler nutzen heute KI, um schneller voranzukommen ([Stack Overflow Developer Survey, 2025](#)). Komplette Anwendungsprototypen, deren Entwicklung früher Wochen dauerte, werden heute in Tagen fertiggestellt. Die Demokratisierung der Software-Entwicklung ist real und nimmt an Fahrt auf – ein Trend, den CIOs genau im Blick haben. Gartner geht davon aus, dass bis zum Jahr 2030 Teams 75 % der IT-Aufgaben mit KI-Unterstützung erledigen, während die KI die restlichen 25 % vollkommen autonom übernimmt

([Gartner, November 2025](#)). Das ist keine Prognose für die ferne Zukunft, sondern der Planungshorizont, an dem sich die meisten IT-Teams aktuell orientieren.

Die Investition ist gerechtfertigt, die Richtung stimmt, aber die höhere Entwicklungsgeschwindigkeit hat bis jetzt noch nicht die transformativen Geschäftsergebnisse gebracht, die der Vorstand erwartet. Und IT-Leiter müssen erklären, warum das so ist.

Diese Diskrepanz ist kein Technologieproblem, sondern ein Architekturproblem. Dazu gehören drei zentrale Fragen, auf die die meisten Unternehmen keine klare Antwort haben.

- 1. Ist die KI-Entwicklung mittlerweile teurer als die Neueinstellung weiterer Entwickler?**
- 2. Ist KI-generierter Code vertrauenswürdig genug für unternehmensweite Geschäftsprozesse?**
- 3. Wie erhält man die Transparenz und Kontrolle darüber, was KI entwickelt und implementiert?**



Die Kostenfrage: Zahlen Sie mehr, nur um neue Altlasten aufzubauen?

Das Versprechen der KI-gestützten Software-Entwicklung war simpel: weniger Kosten und kürzere Entwicklungszeiten. Tatsächlich hat sich jedoch in vielen Unternehmen die Kostenstruktur auf eine Weise verändert, die in der ursprünglichen Planung nicht vorgesehen war.

Programmieren ist nicht der kostspielige Teil der Software-Entwicklung und macht nur rund 10 % des gesamten Anwendungszyklus aus. Die restlichen 90 % bestehen aus Überprüfungen, Tests, Sicherheits-Patches, Debugging, Refactoring und der laufenden Softwarewartung, wenn sich Geschäftsanforderungen ändern. Durch KI-Coding-Assistenten sind die 10 % schneller geworden und, wie die Daten zeigen, sind die Kosten für die restlichen 90 % dadurch deutlich gestiegen.

Laut einer [Analyse von GitClear aus dem Jahr 2025](#) von 211 Millionen Codezeilen aus Repositories von Google, Microsoft, Meta und anderen Unternehmen finden sich in KI-gestützten Projekten heute achtmal mehr doppelte Code-Blöcke als vor zwei Jahren. Auch der Code Churn – Zeilen, die innerhalb von zwei Wochen nach der Programmierung überarbeitet oder gelöscht werden – hat sich von 2020 bis 2024 verdoppelt. GitClear-Gründer Bill Harding vergleicht KI-generierte Code-Funktionen treffenderweise mit einer „neuen Kreditkarte“ für die Anhäufung von technischen Schulden: Die Investition in KI wird als produktivitätssteigernd empfunden, aber die Rechnung kommt später. Wie der [Harness-Bericht „The State of Software Delivery 2025“](#) zeigt, verbringt der Großteil der Entwickler mehr Zeit mit dem Debugging von KI-generiertem Code als mit der Entwicklung von neuen Features.

Durch die Token-Steuer wird das alles noch schlimmer: Jede Codezeile, die die KI schreibt, kostet Token. Anders als bei einem Token für eine Kundeninteraktion entsteht durch ein Token, mit dem Anwendungscode generiert wird, ein Element, dessen Wartung laufende Investitionen verlangt. Wenn dieses Element schneller Schulden anhäuft als einen Wert zu liefern, ist es um die Wirtschaftlichkeit von Anfang an schlecht bestellt.

Dazu kommt, dass die Kostenstruktur der KI-gestützten Entwicklung selbst nicht länger vorhersagbar ist. Seit Juni 2026 rechnet GitHub Copilot nach Verbrauch ab und hat sein Flatrate-Modell für Premium-Anfragen durch tokenbasierte AI Credits ersetzt. In einer Erklärung räumte GitHub ein, dass „nach dem alten Modell eine kurze Chat-Frage und eine mehrstündige autonome Coding-Sitzung den Benutzer denselben Betrag kosten können“ und dass das Abfedern eskalierender Inferenzkosten „wirtschaftlich nicht mehr tragbar ist“ ([GitHub Blog, April 2026](#)). Wer das Budget für seine KI-Entwicklungskosten basierend auf einer Flatrate aufstellte, muss jetzt mit schwankenden Kosten und weniger Transparenz über die tatsächlichen Ausgaben pro Workflow klarkommen.

Die eigentliche Frage ist nicht, ob der Einsatz von KI-Tools Vorteile bringt. Darauf lautet die Antwort eindeutig „Ja“. Die wichtige Frage ist: Reduziert der aktuelle Ansatz – bei dem KI zur schnelleren Code-Generierung eingesetzt wird – tatsächlich die Gesamtentwicklungskosten oder lediglich die Bereitstellungszeiten, während sich die Kosten für die Softwarewartung akkumulieren und in den nächsten Jahren das Budget verschlingen werden?

Die Performance- und Sicherheitsfrage: Wie vertrauenswürdig sind KI-generierte Anwendungen?

Eine schnellere Implementierung rechnet sich nur dann, wenn man der Ausführung der bereitgestellten Software unternehmensweit vertrauen kann. Unter diesem Aspekt haben IT-Leiter nach den bisherigen Erfahrungen allen Grund zur Vorsicht.

Wie CodeRabbit bei der Analyse von 470 Pull-Requests herausfand, verursacht KI-generierter Code 1,7-mal mehr Probleme und 2,74-mal mehr Schwachstellen als von Menschen geschriebener Code ([CodeRabbit, „State of AI vs. Human Code Generation“, Dezember 2025](#)). Und eine auf der [ISSRE 2025 von IEEE vorgestellte Studie](#) kam zu dem Ergebnis, dass KI-generierter Code mehr Hochrisiko-Schwachstellen aufweist, obwohl er strukturell einfachere Ergebnisse liefert. Das legt nahe, dass das Risiko in der Art der Code-Erstellung liegt, nicht nur in der Code-Komplexität.

KI-Tools generieren Lösungen für unmittelbare, lokale Probleme. Sie kennen nicht Ihre allgemeine Systemarchitektur, Ihre internen Abstraktionen, Ihre gemeinsam genutzten Dienstprogramme noch die Sicherheitsbeschränkungen für den Datenverkehr in Ihrer Umgebung. Das Ergebnis ist ein Code, der Unit-Tests und interne Prüfungen besteht, aber Schwachstellen einführt, die sich nur in Ausnahmefällen oder bei hoher Produktionslast zeigen.

76 % der Entwickler, die mit KI-Coding-Tools arbeiten, räumen ein, den KI-generierten Code nicht immer vollkommen zu verstehen ([Stack Overflow, 2026](#)). Diese Zahl hat erhebliche Folgen für den operativen Einsatz. Wenn ein CIO von einem Regulierer, einem Rechtsberater oder dem Vorstand gefragt wird, warum es zu einem bestimmten Systemverhalten kam, ist „Die KI hat das programmiert und der Entwickler war sich nicht sicher, wie das genau funktioniert“ keine gute Antwort. Aber genau das ist die Haltung zum verantwortlichen KI-Einsatz, die dahinter steckt, wenn Unternehmen KI-generierten Code ohne einen strukturellen Governance-Layer in Produktionsumgebungen nutzen.

Zu den wichtigsten Gartner-Vorhersagen für 2026 gehört die „Death by AI“-Prognose. Demnach wird mit weltweit mehr als 2.000 Rechtsstreitigkeiten wegen Vorfällen gerechnet, die aufgrund mangelnder Governance bei KI-generierten Anwendungsergebnissen entstehen. ([Gartner, Oktober 2025](#)). Dies sind keine hypothetischen Zukunftsrisiken. Dies stellt schon heute ein akutes Risiko für alle Unternehmen dar, die KI-generierten Code für entscheidende Aufgaben in regulierten oder Kunden-Workflows verwenden.

Die Zuverlässigkeitsfrage: Wissen Sie, was KI entwickelt?

Geschwindigkeit und Sicherheit sind operative Aspekte, wohingegen Überprüfbarkeit eine strukturelle Notwendigkeit ist, die CIOs und Unternehmensarchitekten im Nachhinein wahrscheinlich am schwersten regeln können.

Wenn eine Anwendung eine Entscheidung trifft – sei es zur Weiterleitung eines Vorgangs, zur Klärung einer Berechtigung, zur Gewährung von Zugang oder für die Erstellung von Ergebnissen, die Kunden präsentiert werden –, trägt dafür das IT-Team die Verantwortung, nicht die KI, die den Code geschrieben hat. Verantwortlich ist die IT-Abteilung, die die Anwendung implementiert hat, und die Führungskräfte, die dies genehmigt haben.

In einer klassischen Entwicklungsumgebung ist diese Verantwortlichkeit kein Problem, weil sich die Logik überprüfen lässt. Fragt ein Auditor, warum eine Entscheidung getroffen wurde, sieht sich der Entwickler den Codepfad an und kennt darauf die Antwort. In einer Umgebung, in der die Anwendungslogik KI-generiert wurde, ist diese Nachverfolgbarkeit nicht garantiert. KI-generierter Code erzeugt funktionierende Ergebnisse, aber das Reasoning hinter einer bestimmten Implementierungsentscheidung wird nicht auf die gleiche Weise dokumentiert wie die Intention eines Programmierers. Auf die Frage eines Auditors, warum ein System in einem bestimmten Grenzfall eine bestimmte Entscheidung getroffen hat, ist „weil die KI das so programmiert hat“ keine haltbare Antwort.

Dies ist das Zuverlässigkeitsproblem, das auf Unternehmensebene am meisten zählt. Es geht nicht darum, ob der Code meistens richtig ausgeführt wird. Es geht darum, ob Sie ein bestimmtes Ergebnis jedem, der es irgendwann in der Zukunft nachvollziehen möchte, in einer Sprache so erklären können, dass er mit Ihrer Antwort etwas anfangen kann.

Eine visuelle, deklarative Anwendungslogik löst dieses Problem auf struktureller Ebene. Liegen Geschäftsregeln, Entscheidungstabellen und Workflow-Weiterleitungen als überprüfbare Konfiguration statt als generierter Code vor, wird jeder Entscheidungspfad ohne forensische Analyse nachverfolgbar. Die Logik existiert auf einer Ebene, die Compliance-Teams, Rechtsberater und Beteiligte aus dem Unternehmen (und nicht nur Entwickler) verstehen. Bei einem Audit wird dann die Konfiguration überprüft, statt in mühevoller Kleinarbeit den Code aufzuschlüsseln.

Die Architektur, die alle drei Probleme löst

Die gute Nachricht: Die Umstellung auf eine Architektur, die alle drei Probleme bezüglich Kosten, Sicherheit und Zuverlässigkeit löst, schränkt den KI Einsatz in keiner Weise ein. Vielmehr handelt es sich um eine Neuausrichtung, wo in der Entwicklung mit KI gearbeitet wird.

KI ist besonders effektiv darin, die Erstellung von übergeordneten Anwendungsstrukturen zu beschleunigen, wie Datenmodelle, Workflow-Definitionen, Entscheidungstabellen und Weiterleitungslogik in deklarativer Form. Dies sind Artefakte, die beschreiben, was eine Anwendung macht, ohne die Implementierung in imperativem Code zu verschlüsseln. Sie können auf eine Weise gelesen, überprüft und gewartet werden, die bei generiertem Code nicht möglich ist. Ändern sich die Geschäftsanforderungen, ändert sich mit ihnen das Modell und das Anwendungsverhalten, aber ohne kaskadierende Code-Änderungen und Regressionsrisiken.

Bei diesem Entwicklungsansatz steht das Modell an erster Stelle. Die KI übernimmt die Aufgaben, in denen sie wirklich gut ist: Übersetzung der Absicht in eine Struktur, Generierung der Konfiguration aus natürlicher Sprache und Beschleunigung der Architekturdefinition. Das Ergebnis ist ein Modell, kein Code. Wenn Code generiert werden muss, geschieht dies „unter der Haube“ als Teil der Implementierung, nicht als primäres Artefakt.

Aus operativer Sicht bringt das konkrete Vorteile: Teams investieren weniger Zeit in die Softwarewartung, weil am Modell, nicht an der Codebasis gearbeitet wird. Sicherheitsprüfungen lassen sich besser nachverfolgen, weil die Business-Logik deklarativ ausgedrückt wird. Compliance-Audits nehmen weniger Zeit in Anspruch, weil die Entscheidungsebene ohne IT-Unterstützung überprüfbar ist. Und der Schwerpunkt bei den Token-Kosten liegt auf dem wertvollsten Teil der Entwicklung: der Definition dessen, was die Anwendung machen soll (statt die technische Umsetzung der KI zu überlassen).

Nach [Gartners I&O-Befragung Ende 2025](#) erfüllen nur 28 % der KI-Infrastruktur-Anwendungsfälle ausnahmslos die ROI-Erwartungen. Diese Diskrepanz liegt nicht an der Leistungsfähigkeit von KI, sondern an der Architektur. Unternehmen, bei denen sich KI in der Entwicklung rechnet, verwenden nicht unbedingt fortschrittlichere Modelle. Der Unterschied liegt darin, dass diese Unternehmen KI zur Erstellung von Artefakten nutzen, die nicht dem raschen Wertverlust von generiertem Code unterliegen.

Die Architekturentscheidung

IT-Leiter müssen sich nicht zwischen Geschwindigkeit und Sicherheit entscheiden. Das wird jedoch oft fälschlicherweise angenommen, wodurch die aktuelle Situation als unzumutbare Einschränkung empfunden wird. Die tatsächliche Entscheidung ist die zwischen zwei verschiedenen Architekturen mit recht unterschiedlichen langfristigen Kosten und Risikoprofilen.

Eine Architektur, die im Kern auf KI-generiertem Code beruht, liefert anfangs schnell Ergebnisse und geht mit einem wachsenden Wartungsaufwand, begrenzter Überprüfbarkeit und Sicherheitsrisiken einher – und diese Nachteile nehmen zu, je größer die Codebasis wird. Basiert die Architektur dagegen im Kern auf der modellorientierten Entwicklung mit eingebetteter KI, liefert sie Ergebnisse, deren Design die Überprüfbarkeit unterstützt, die sich auf Modellebene warten lassen und die den Token-Verbrauch auf die Entwicklungsaufgaben begrenzt, die dauerhaft von Wert sind.

Die IT-Leitung muss weder die geschäftliche Forderung nach einer KI-beschleunigten Implementierung ablehnen, noch alles langsamer angehen. Die Lösung besteht in einer anderen Entwicklungsphilosophie, die für das Unternehmen tragbar und für die IT vertretbar ist.

Die richtige KI-Investition besteht nicht darin, schneller programmieren zu können, sondern sollte in die Entwicklung von Anwendungsmodellen fließen, die nicht ständig überarbeitet werden müssen.





Vielen Dank!

Über uns

Pega bietet die führende KI-gestützte Plattform für die Transformation von Unternehmen. Viele der weltweit einflussreichsten Unternehmen vertrauen auf unsere Technologie zur Neugestaltung von Arbeitsabläufen – durch Workflow-Automatisierung, persönliche Kundenerlebnisse und modernisierte Legacy-Systeme. Seit 1983 treibt die skalierbare, flexible Architektur kontinuierlich Innovationen voran und hilft Kunden dabei, ihren Weg zum autonomen Unternehmen zu beschleunigen. **[PEGA.COM/DE](https://www.pega.com/de)**