

# Vibe Coding 2026: Diese Tools machen Code überflüssig

Im Februar 2025 schrieb KI-Forscher Andrej Karpathy [auf X einen Satz](#), der die Welt der Softwareentwicklung verändern sollte: "There's a new kind of coding I call vibe coding, where you fully give in to the vibes, embrace exponentials, and forget that the code even exists." Was er als spontanen Gedanken postete, wurde innerhalb weniger Monate zum Branchenbegriff – und gewann im November 2025 den Titel [Word of the Year des Collins Dictionary](#). Heute schreiben nicht-technische Mitarbeitende in Unternehmen ganze Anwendungen, indem sie einer KI in natürlicher Sprache beschreiben, was sie wollen. Wie weit die Adoption inzwischen geht, zeigt der [Stack Overflow Developer Survey 2025](#): 84 Prozent der weltweit befragten Entwickler:innen nutzen oder planen AI-Tools im Workflow – ein Anstieg von 76 Prozent im Vorjahr. Softwareentwicklung war noch nie so zugänglich – und gleichzeitig so umstritten.

Doch was bedeutet Vibe Coding konkret? Wo schafft es echten Wert für Unternehmen, und wo liegen die Grenzen? Auf der [d:u26](#) haben Johannes Kliesch, Geschäftsführer von SNOCKS, und Calvin Blick, Geschäftsführer von HeroSoftware, genau diese Fragen beantwortet – mit überraschend ehrlichen Einblicken aus dem eigenen Arbeitsalltag. Dieser Artikel fasst die wichtigsten Erkenntnisse zusammen, ordnet sie in den größeren Kontext ein und zeigt, wann sich Vibe Coding für Mittelständler:innen lohnt und wann nicht. Die komplette Session der beiden Gründer kannst du dir – wie alle Vorträge der [d:u26](#) – on demand in der [d:u Education-Mediathek](#) ansehen.

## Was ist Vibe Coding?

Vibe Coding ist eine Form der Softwareentwicklung, bei der ein:e Anwender:in einer KI in natürlicher Sprache beschreibt, was eine Anwendung tun soll. Aus einem einzigen Prompt generiert das Sprachmodell den gesamten Code, ohne dass die Person, die den Prompt schreibt, jede Codezeile selbst versteht oder schreibt. Karpathy formulierte es zugespitzt: Man gibt sich der Sache hin, vertraut den Modellen und vergisst, dass der Code überhaupt existiert.

Möglich wird das durch die Fortschritte bei Large Language Models. Moderne Sprachmodelle wie Claude Opus, GPT-5 oder Gemini sind heute in der Lage, aus einem natürlichsprachlichen Prompt mehrere hundert Zeilen funktionsfähigen Code zu produzieren. Diese LLMs sind das technische Fundament, ohne das die Methode nicht funktionieren würde.

Das ist mehr als nur eine Marketing-Phrase. Es beschreibt einen echten Bruch mit der bisherigen Art, wie Software entsteht. Klassische Entwicklung beginnt mit einer technischen Spezifikation, dann folgen Architekturentscheidungen, Code, Tests, Reviews. Vibe Coding beginnt mit einer Idee – "ich brauche ein Reservierungssystem fürs Café" – und endet, im

besten Fall, mit einer funktionierenden Anwendung. Was dazwischen passiert, übernimmt die künstliche Intelligenz.

Wichtig ist die Abgrenzung: Vibe Coding ist nicht dasselbe wie KI-gestütztes Programmieren mit klassischen KI Tools. Wer GitHub Copilot oder Cursor nutzt und dabei jeden KI generierten Code zeilenweise reviewt, betreibt KI gestützte Programmierung. Wer dagegen einer KI ein Tool beschreibt und das Ergebnis übernimmt, ohne den generierten Code zu prüfen, vibe-coded. Der Unterschied liegt in der Kontrolle. Mehr zu den verfügbaren Lösungen gibt es in unserem [Überblick zu KI Coding Tools 2026](#).

## Die wichtigsten Vibe Coding Tools

Der Markt hat sich 2025 und 2026 schnell konsolidiert. Vier Anbieter sind heute besonders relevant.

### Lovable

Lovable gilt heute als eine der reichweitenstärksten Plattformen im Vibe-Coding-Markt. Das liegt nicht an überlegener Technik, sondern an Nutzerfreundlichkeit. SNOCKS-Geschäftsführer Johannes Kliesch beschreibt es treffend: "Lovable ist so ein bisschen das iPhone. Das sieht am coolsten aus, das macht so interaktiv am meisten Spaß." Für ein Unternehmen mit 150 Mitarbeitenden wie SNOCKS ist genau das entscheidend – die Tools werden nur genutzt, wenn die Arbeit damit Spaß macht. SNOCKS selbst hat bisher rund 40.000 Euro an Lovable-Credits ausgegeben und sämtliche internen Webprojekte damit umgesetzt.

### Replit Agent

Replit zielt auf eine technischere Zielgruppe. Mehrere parallele Agenten, mehr Customization, mehr Konfigurationsmöglichkeiten. Für Solo-Unternehmer:innen und Ein-Personen-Teams oft die leistungsfähigere Wahl, für größere Teams aber häufig zu komplex. Kliesch empfiehlt es für genau diese Zielgruppe: "Wenn ihr eine Ein-Mann-Bude oder eine Ein-Frau-Bude seid, dann ist Replit so ein bisschen das Android."

### Bolt und v0

Bolt erzeugt aus einer Chat-Eingabe vollständige Web-Apps inklusive Backend und Datenbank. v0 von Vercel ist auf React-Komponenten und Frontend-Code spezialisiert. Beide eignen sich besonders für UI-Prototypen und schnelles Prototyping vor einer größeren Investition.

### Base44

Base44 hat 2025 und 2026 als ernstzunehmender Anbieter Aufmerksamkeit gewonnen, vor allem für komplexere Anwendungen mit Datenbankanbindung.

# Wo Vibe Coding tatsächlich Geld spart: Praxisbeispiele von SNOCKS

Was unterscheidet Hype von Praxis? Konkrete Zahlen. SNOCKS hat in der d:u26-Session aufgeschlüsselt, wo Vibe Coding messbar Geld einspart.

## Reservierungssystem fürs Café

SNOCKS betreibt ein eigenes Café in Mannheim, das rund eine Million Euro Umsatz im Jahr macht. Für das Reservierungssystem zahlte das Unternehmen monatlich 300 bis 400 Euro an einen externen Anbieter. Innerhalb einer Woche vibe-coded das interne Team die gesamte Anwendung nach. Einmalige Kosten an Lovable-Credits: rund 500 bis 1.000 Euro. Laufende Kosten danach: nahezu null. Die Amortisation ist eine Frage von Monaten.

## Internes Formular-Tool statt Typeform

Typeform ist ein beliebter Standarddienst für Online-Formulare. SNOCKS gab dafür bis zu 1.000 Euro im Monat aus. Die nachgebaute Lovable-Version geht heute live, der laufende Aufwand ist verschwindend gering.

## Webseiten und Landingpages

Für einfache B2B-Seiten und Agentur-Webseiten ohne komplexe Funktionalität sieht Kliesch keine Notwendigkeit mehr für klassische Webagenturen: "Wer da heutzutage noch Geld für eine Agentur ausgibt, komplett unnötig." Lovable und ähnliche Tools können URLs als Vorlage einlesen und Designs in der eigenen Markenfarbe nachbauen. Für reines Prototyping vor einer größeren Investition lohnt sich der Ansatz besonders, weil sich Ideen in wenigen Stunden testbar machen lassen.

Das Muster ist klar: Überall dort, wo eine Anwendung im Kern eine schönere Datenbank mit einfacher Logik ist – Reservierungen, Formulare, einfache Dashboards, Webseiten – schafft Vibe Coding messbaren Mehrwert. Auch eigene Ideen aus den Fachbereichen lassen sich so direkt in lauffähige Anwendungen übersetzen.

## Wo Vibe Coding klar an Grenzen kommt

Genauso ehrlich war Calvin Blick von HeroSoftware in seiner Einschätzung, wo der Ansatz nicht funktioniert. Er erzählte in seinem Vortrag auf der d:u26 offen von einem Projekt, das im eigenen Unternehmen schiefging.

## Das CRM-Experiment

"Ich hab für uns einfach ein Vibe-Coded CRM geschrieben", berichtete Blick. "Ich dachte so, ey, ultra smart, HubSpot ist eh insolvent in ein paar Jahren, lass es mal einfach selbst machen." Die nüchterne Bilanz: 30 bis 40 Stunden investiert, am Ende nutzt HeroSoftware heute Attio – ein externes CRM. Der Grund: Wartbarkeit. Sobald der Vertriebler neue

Integrationen forderte, kam Blick an die Grenze. "Da kommt Vibe Coding sehr an die Grenzen, weil diese ganzen Schnittstellen funktionieren mal, mal funktionieren sie nicht." Die 30 Stunden, so sein ehrliches Fazit, "waren einfach für den \*\*\*\*\*".

Die Schlussfolgerung der beiden Gründer: Vibe Coding ist stark bei Anwendungen mit wenigen externen Schnittstellen. Sobald CRM-, ERP- oder andere komplexe Systemintegrationen ins Spiel kommen, wird der Wartungsaufwand zur Hürde, die kein Mittelstand sinnvoll alleine stemmen kann.

## Die letzten 20 Prozent

Ein wiederkehrendes Muster in der Vibe-Coding-Praxis: Die ersten 80 Prozent eines Tools sind in Stunden gebaut. Die letzten 20 Prozent, also Bugfixes, Edge Cases, Performance-Optimierung, kosten unverhältnismäßig viel Zeit. SNOCKS hat darauf intern reagiert: Zwei Vollzeitkräfte sind dort ausschließlich für Vibe Coding zuständig und übernehmen genau diese letzten 20 Prozent für die Fachbereiche. Sie sammeln Feedback aus den Teams, übersetzen das Feedback in präzisere Prompts und reduzieren so Fehler im generierten Code.

## Sicherheit als unterschätztes Risiko

Was in vielen Vibe-Coding-Berichten fehlt, ist die Sicherheitsfrage. Die [Wiz Research](#), das Sicherheitsforschungsteam des israelischen Cybersecurity-Unternehmens, untersuchte 2025 die Sicherheit von Vibe-Coded Anwendungen. Das Ergebnis: Eine von fünf Organisationen, die auf solchen Plattformen entwickeln, war systemischen Risiken ausgesetzt. Die vier häufigsten Probleme bei der Verwendung von KI generiertem Code:

- Authentifizierungslogik komplett im Browser, also für jeden einsehbar
- API-Keys und Secrets hartcodiert im Client-Code
- Datenbanktabellen ohne Zugriffskontrolle, öffentlich zugänglich
- Interne Anwendungen ohne Authentifizierung im offenen Internet

Diese Risiken sind technisch lösbar, aber wer ohne Code-Verständnis vibe-coded, erkennt sie oft nicht. "Solange es funktioniert, ist es hammer, hammer geil", sagt Kliesch. "Aber sobald es anfängt nicht mehr zu funktionieren, ist es eine komplette Blackbox." Wer sensible Daten in Vibe-Coded Anwendungen verarbeitet, sollte deshalb spätestens vor dem Produktiv-Start eine sicherheitstechnische Prüfung einplanen. Mehr zu dem Thema steht in unserem Artikel zu [Datensicherheit und KI](#).

## Vibe Coding vs. KI Coding: Eine wichtige Unterscheidung

Die ehrlichste Botschaft der d:u26-Session: Vibe Coding ist nicht der Endpunkt der Entwicklung. Es ist eine Ergänzung. HeroSoftware selbst, ein Software-Unternehmen mit über 10 Entwicklern und Entwicklerinnen, das in den vergangenen zwei Jahren um 2.500 Prozent gewachsen ist, setzt auf KI gestützte Codegenerierung mit menschlichem Review – nicht auf Vibe Coding im engeren Sinne.

Bei HeroSoftware werden inzwischen 41 Prozent des produktiv eingesetzten Codes von KI geschrieben. Aber jede einzelne Zeile durchläuft Review durch erfahrene Senior-Entwickler:innen. Die genutzten Tools: Cursor für die alltägliche Entwicklung mit verschiedenen LLMs im Hintergrund, Augment Code für Frontend-Aufgaben mit größerem Kontext, Devin für autonome Pull-Request-Reviews.

Blicks Einordnung in der Session war klar: "KI Coding ist jetzt nicht so shiny object, ey geil, wir gehen morgen damit live. Es ist langsamer und es braucht Entwickler." Der entscheidende Vorteil: "Wir haben eine Codebase, die echt ist und die wir kontrollieren können. Und das ist die größte Hürde, die wir bei Vibe Coding haben, weil wir einfach zu viel mit Vertrauen arbeiten und zu wenig mit Kontrolle." Mit dieser Skepsis ist HeroSoftware nicht allein: Laut [Stack Overflow Developer Survey 2025](#) misstrauen 46 Prozent aller Entwickler:innen den Outputs von AI-Tools – mehr als ein Anstieg von 15 Prozentpunkten gegenüber dem Vorjahr.

## **Wann lohnt sich Vibe Coding im Mittelstand?**

Aus der Praxis von SNOCKS und HeroSoftware lassen sich klare Kriterien ableiten, wann Vibe Coding sich für Mittelständler:innen wirklich lohnt.

### **Gute Einsatzfelder**

Vibe Coding spielt seine Stärken aus bei einfachen Webseiten ohne komplexe Backend-Logik, internen Tools mit überschaubarem Funktionsumfang, Prototypen zur Validierung von Geschäftsideen, Excel-Ersatzlösungen mit schickerem Interface und Anwendungen mit wenigen externen Schnittstellen. Auch für Mitarbeitende ohne formale Programmierkenntnisse ist der Ansatz ideal – sie können eigene Ideen direkt umsetzen, ohne erst eine klassische Programmierung erlernen zu müssen.

### **Wo Vibe Coding nicht passt**

Schwierig wird es bei CRM-, ERP- oder anderen Systemintegrationen mit vielen Schnittstellen, bei Anwendungen mit sensiblen Personendaten ohne externe Security-Prüfung, bei Software, die produktiv über Jahre laufen und gewartet werden soll, sowie bei komplexen Backend-Logiken mit hohen Performance-Anforderungen.

Wer diese Grenzen kennt, kann Vibe Coding strategisch einsetzen, ohne in die typische Falle zu tappen – nämlich zu glauben, dass damit alle Software-Probleme gelöst sind.

# Vibe Coding im Unternehmen einführen: Drei Tipps aus der Praxis

Aus den Erfahrungen von SNOCKS lassen sich drei konkrete Empfehlungen ableiten, wenn ein Unternehmen Vibe Coding ernsthaft einführen will.

## Klein anfangen, schnell Wert zeigen

SNOCKS startete nicht mit dem Anspruch, das gesamte Unternehmen umzustellen. Stattdessen wurden konkrete Use Cases identifiziert – Reservierungssystem, Formulare, einfache Dashboards – die schnell sichtbar Geld sparen. Diese Erfolge schufen die Akzeptanz für weitere Projekte.

## Champions in Fachbereichen aufbauen

Heute ist Vibe Coding bei SNOCKS keine IT-Aufgabe mehr. Führungskräfte aus Marketing, Retail und anderen Bereichen arbeiten selbst mit Lovable. Diese Champions sind das wirksamste Instrument, um Vibe-Coding-Kompetenz in einer Organisation aufzubauen. Sie lernen durch Ausprobieren – und geben dieses Lernen direkt an ihre Teams weiter.

## Senior-Support für die letzten 20 Prozent

SNOCKS beschäftigt zwei Vollzeitkräfte, die ausschließlich Vibe Coding machen und Fachbereichen bei den letzten 20 Prozent helfen. Wer in der gleichen Größenordnung Vibe Coding einsetzen will, braucht solche Expert:innen – sonst bleiben viele Projekte bei 80 Prozent Fertigstellung liegen.

Wie weitere Mittelständler:innen Vibe Coding strategisch einsetzen und wo sie an Grenzen gestoßen sind, ist eines der Themen auf dem [data:unplugged Festival 2027, der d:u27, 7 vom 13. bis 14. April 2027 in Münster](#). Auf der Bühne sprechen Unternehmen über ihre konkreten Anwendungsfälle, Tools und Lessons Learned aus dem Alltag.

## Was ist 2026 noch zu erwarten?

Die Entwicklungsgeschwindigkeit ist hoch. Replit hat zuletzt ein Feature live geschaltet, mit dem sich jede beliebige URL als Design-Vorlage einlesen lässt – das Tool kopiert komplette Webseiten und passt sie an die eigenen Markenvorgaben an. Lovable hat seine Sicherheits-Defaults nach der Wiz-Studie überarbeitet und integriert nun standardmäßig Best-Practice-Authentifizierung. Mit jedem neuen Modell-Sprung werden zudem Fehler im KI generierten Code seltener.

Karpathys ursprüngliche Definition, "embrace exponentials", wirkt heute wieder zutreffend: Was vor einem Jahr noch experimentell war, ist heute alltagstauglich für viele Use Cases. Was heute noch unzureichend ist – etwa komplexe System-Integrationen oder produktive Enterprise-Software – kommt mit dem nächsten Modell-Sprung womöglich in Reichweite.

Für Unternehmen heißt das: Wer heute pragmatisch mit Vibe Coding beginnt, baut die Kompetenzen auf, die in den kommenden Jahren wertvoll werden. Wer wartet, verliert genau diese Lernkurve.

## **Fazit: Pragmatisch nutzen, kritisch bleiben**

Vibe Coding ist weder Allheilmittel noch Hype. Es ist ein neues Werkzeug, das bei den richtigen Aufgaben messbaren Wert schafft. SNOCKS spart heute hohe vierstellige Beträge pro Monat durch selbst vibe-coded Tools. HeroSoftware skaliert sein Entwicklungsteam, ohne neue Köpfe einzustellen, weil 41 Prozent des Codes mit KI-Unterstützung entstehen.

Gleichzeitig zeigen die ehrlichen Berichte von Calvin Blick und die Wiz-Sicherheitsstudie: Vibe Coding hat klare Grenzen. Wer sie kennt und respektiert, gewinnt. Wer sie ignoriert, baut Schatten-IT, die in zwei Jahren zur Belastung wird.

Wie andere Mittelständler:innen genau diese Balance finden, welche Tools sich bewähren und welche Fallstricke vermieden werden müssen, ist Thema auf dem [data:unplugged Festival](#). Auf der [d:u27 vom 13. bis 14. April 2027 in Münster](#) teilen Praktiker:innen ihre Erfahrungen mit Vibe Coding und KI Coding – auf der Bühne und in den Masterclasses. Im Austausch mit anderen Entscheider:innen entstehen praxisnahe Einblicke, die über reine Anleitungen hinausgehen. [Sichere dir jetzt dein Ticket](#) für dich und dein Business Team und profitiere direkt vom Austausch.