

Quo vadis, inżynierze?

*Inżynier z krwi i kości w epoce vibe codingu –
o złudzeniu tempa, złudzeniu bezpieczeństwa
i o tym, co nie staniało*





01

Nigdy tak wielu nie tworzyło oprogramowania, którego nie rozumie

Kiedy Karpathy pisał w lutym 2025 o vibe codingu, brzmiało to jak opis weekendowej zabawy: przyjmujesz wszystko, co proponuje model, i zapominasz, że kod istnieje.

Niecałe półtora roku później Google deklaruje, że AI generuje już około trzech czwartych nowego kodu w firmie. Jesienią 2024 była to jedna czwarta. Taka krzywa nie zna słowa „eksperyment”.

A piętro niżej biznes składa sobie własne narzędzia. Wszystko działa - dopóki nikt nie zapyta, kto to utrzyma i kto za to odpowie.



02

Może rzeczywiście nie potrzeba inżyniera

Powiem to uczciwiej, niż wypada człowiekowi, który prowadzi firmę inżynierską: do wielu rzeczy inżynier naprawdę przestał być potrzebny. Do prototypu - nie jest. Do narzędzia dla dziesięciu osób - zwykle też nie. I dobrze.

Inżynier staje się potrzebny w tej chwili, w której system zaczyna coś znaczyć. Kiedy przechodzą przez niego prawdziwe pieniądze i prawdziwe dane. Kiedy ma działać także drugiego stycznia o trzeciej w nocy, po awarii łącza i przy zdublowanym rekordzie.

Granica nie przebiega między „umie kodować” a „nie umie”. Przebiega między „działa” a „wiem, dlaczego działa”

Kod stał się tani

Osąd zdrożał

AI podnosi sufit temu, kto rozumie fundamenty,
i obniża podłogę temu, kto ich nie ma

03

Złudzenie tempa

Byli przekonani, że AI przyspiesza ich pracę o:

+20%

A ich zadania trwały dłużej o:

+19%

Doświadczeni programiści open source pracowali w eksperymencie METR nad zadaniami we własnych projektach - raz z AI, raz bez (lipiec 2025). Rok później METR przyznał, że kolejnej rundy nie umie już rzetelnie zmierzyć, bo coraz trudniej o programistów gotowych pracować bez AI - nowe dane sugerują przyspieszenie, ale badacze traktują je ostrożnie.

Jeden wniosek przetrwał obie rundy: nasze odczucie tempa potrafi rozjechać się z rzeczywistością o czterdzieści punktów procentowych. Jak bardzo mylimy się więc co do jakości tego, czego nie czytamy?

METR: eksperyment kontrolowany, lipiec 2025 + komunikat o zmianie metody, luty 2026

44%

prób generowania kodu kończy się podatnością ze znanych kategorii OWASP - wynik bez zmian od roku, choć modele wyraźnie zmadrzały

Veracode, GenAI Code Security Report 2026 (lipiec 2026)

Wrażenie kompletności to najtańsza rzecz, jaką generuje AI



04

Pod spodem nie ma nikogo

Modele wymyślają nazwy bibliotek, których nie ma - dotyczy to mniej więcej co piątej sugestii pakietu (USENIX Security 2025). Napastnicy rejestrują te nazwy z wyprzedzeniem i podkładają pod nie złośliwy kod. Zjawisko ma już nazwę: slopsquatting.

Agenci dopisali własny rozdział. Latem 2025 agent Replita usunął produkcyjną bazę danych podczas blokady zmian, a potem generował dane maskujące problem. W kwietniu 2026 agent w Cursorze, poproszony o zadanie na środowisku testowym, znalazł produkcyjny token i w dziewięć sekund skasował produkcyjny wolumen razem z kopiami zapasowymi.

Dwa różne narzędzia, ten sam wzorzec: intencja testowa, uprawnienia produkcyjne i żadnej bramki z człowiekiem.

05

Shadow IT drugiej generacji

Kiedyś biznes kupował sobie SaaS na firmową kartę poza wiedzą IT. Dziś sam sobie oprogramowanie generuje. Mechanizm ten sam, stawka nieporównywalna.

To już nie przeczucie, tylko policzone zjawisko: według IBM (Cost of a Data Breach 2025) co piąte naruszenie danych miało związek z niekontrolowanymi narzędziami AI, a tam, gdzie shadow AI kwitło, koszt naruszenia rósł o kilkaset tysięcy dolarów.

Z zewnątrz nie odróżnicie solidnej konstrukcji od sklejki z podatności i sekretów wpisanych na twardo. To widać tylko od środka - a od środka nikt nie zaglądał.

*Demo wygląda znakomicie. Interfejs jest ładniejszy niż w niejednym produkcie.
Wrażenie kompletności kosztuje zero*



06

Co się nie zdewało

Nie staniało myślenie systemowe: umiejętność zobaczenia całości, od warstwy sieciowej po aplikację, i przewidzenia, gdzie ta całość pęknie przy dziesięciokrotnym obciążeniu. Model widzi pliki, które mu pokażecie. Człowiek widzi architekturę razem z jej historią i kompromisami.

Nie staniało diagnozowanie. Kiedy o trzeciej w nocy monitoring pokazuje zieleń, a klaster odmawia współpracy, różnica między „umiem wygenerować playbook” a „coś mi mówi, że to certyfikat” to różnica między kwadransiem a dobą przestoju. Tego przeczucia nie da się pobrać - ono osadza się latami, po jednej awarii na raz.

Nie staniała odpowiedzialność. Model nie podpisze się pod systemem przed zarządem, audytorem ani regulatorem.



07

Nowy warsztat

Dekompozycja i specyfikacja

cała wartość przenosi się do umiejętności poproszenia o właściwą rzecz

Przegląd kodu jako rdzeń pracy

czytanie maszynowej produkcji zamiast pisanie od zera

Bezpieczeństwo jako odruch

skąd pakiet, co robi zależność, gdzie sekrety, co może agent

Fundamenty

sieci, protokoły, systemy, bazy - jedyne narzędzie oceny tego, co produkuje model

Czego na tej liście nie ma? Prompt engineeringu. Rozmowy z modelem nauczycie się w tydzień



o8

Jak się w tym odnaleźć

Używajcie AI codziennie, ale w trybie pilota, nie pasażera. Pilot wie, dokąd leci, i co chwilę zerka na przyrządy. Pasażer ma wygodnie i nie ma wpływu na lądowanie. Obaj lecą tym samym samolotem.

Raz w tygodniu przeczytajcie do dna coś, co model dla Was wygenerował - razem z zależnościami, z konfiguracją, z pytaniem „dlaczego akurat tak”. To najtańszy trening osądu, jaki znam. Wynik bywa ożywiający.

I pielęgnujcie jedną domenę, w której jesteście głębiej niż model. Wiecie wtedy, jak smakuje prawdziwe zrozumienie - i czujecie, kiedy gdzie indziej go brakuje.

Dokąd zmierzasz, inżynierze?

Przez trzydzieści lat można było uważać, że inżynieria to pisanie kodu, a rozumienie przychodzi samo, przy okazji. AI właśnie zabrało nam tę wygodną wersję zawodu.

Maszyny przejęły pisanie - najłatwiejszą część tej pracy. Nie przejęły rozumienia, rozstrzygania ani odpowiedzialności.

Moja odpowiedź na tytułowe pytanie: w głąb własnego zawodu. Bliżej fundamentów, bliżej architektury, bliżej odpowiedzialności. Tam jest dziś więcej pracy niż kiedykolwiek - i znacznie mniej tłoku.

*Maszyny przejęły pisanie.
Rozumienie wciąż jest nasze*

**A kiedy Wy ostatnio przeczytaliście do końca kod,
który wygenerował Wam model? Dyskusja pod postem**

Jacek Bugajski • jacek.bugajski@snok.ai • snok.ai

Źródła: METR (VII 2025, II 2026) • Veracode 2026 • Google/Business

Insider (IV 2026) • IBM (2025) • USENIX Security 2025 • Replit (VII 2025) • Cursor (IV 2026)

