

# Strategická analýza paradigmatu vibecodingu v roce 2026: Od intuitivního prototypování k agentckému inženýrství

Vývoj softwaru v roce 2026 prochází nejvýznamnější transformací od vynálezu vyšších programovacích jazyků. Tento posun, označovaný jako vibe coding, představuje přechod od manuálního psaní syntaxe k režii záměru a systémové architektury prostřednictvím přirozeného jazyka.<sup>1</sup> Zatímco rané fáze adopce umělé inteligence v kódování se soustředili na doplňování řádků (autocomplete), současný stav v dubnu 2026 vykazuje nástup plně autonomních agentů schopných provádět komplexní refaktorizace napříč celými repozitáři.<sup>3</sup> Analýza českého prostředí, ukazuje na snahu o komercializaci tohoto trendu, avšak hlubší zkoumání odhaluje propast mezi obecným vnímáním vibecodingu jako "snadného kódování" a profesionálními standardy vyžadovanými pro produkční systémy.<sup>5</sup>

Tato zpráva dekonstruuje mechanismy vibecodingu, hodnotí aktuální technologický stack, identifikuje kritická rizika spojená s tzv. "vibe coding kocovinou" a definuje nový kompetenční model pro moderního vývojáře, který se z autora kódu stává jeho editorem a architektem.<sup>7</sup>

## Dekonstrukce současného stavu a analýza lokálního trhu

Pojem vibecoding, který poprvé popularizoval Andrej Karpathy v roce 2025, se do roku 2026 stal mainstreamovou praxí, kterou denně využívá 92 % vývojářů ve Spojených státech.<sup>1</sup> V českém kontextu se platformy snaží tento fenomén uchopit skrze marketplace a vzdělávací kurzy, přičemž definuje vibecoding jako přístup, kde vývojář "udává směr" (vibe) a AI generuje technické řešení.<sup>5</sup> Tato definice je však v komunitě vnímána jako značně plochá, neboť opomíjí hloubkovou technologickou disciplínu nezbytnou pro správu rozsáhlých systémů.<sup>5</sup>

Lokální platformy identifikují specifické role, jako je Creative AI Engineer nebo AI-First Full Stack Developer, což naznačuje, že trh práce se již adaptuje na novou realitu, kde je schopnost efektivní interakce s AI agenty klíčovou konkurenční výhodou.<sup>5</sup> Přestože web nabízí základní vhled, profesionální realita roku 2026 vyžaduje pochopení kauzálních vztahů mezi kvalitou promptu, architektonickou integritou a dlouhodobou udržitelností kódu, což jsou aspekty, které přesahují rámec prostého "udávání směru".<sup>7</sup>

Statistická data pro rok 2026 ukazují na masivní adopci, ale zároveň na klesající důvěru v přesnost AI výstupů, což vytváří tlak na vyšší odbornost lidských kontrolorů:

Metrika	Hodnota v roce 2026	Trend oproti 2024
Adopce AI nástrojů u vývojářů (USA)	92 %	Nárůst o 20 % <sup>8</sup>
Podíl AI generovaného kódu globálně	41 %	Nárůst o 15 % <sup>1</sup>
Důvěra vývojářů v přesnost AI kódu	33 %	Pokles z 43 % <sup>6</sup>

Výskyt zranitelností v AI kódu	24,7 % - 45 %	Stabilní riziko <sup>12</sup>
Zvýšení rychlosti prototypování	300 %	Výrazný nárůst <sup>8</sup>

1

## Technologický workflow a agentické pilíře

V roce 2026 se workflow vibecodingu transformoval z lineárního psaní do cyklického procesu šesti fází: záměr (intent), specifikace (spec), generování, revize, iterace a nasazení.<sup>1</sup> Tento cyklus řeší propast mezi mentálním modelem vývojáře a technickou realizací strojem. Klíčem k úspěchu v tomto procesu je tzv. "context engineering" – schopnost poskytnout AI agentovi přesné hranice a data, v nichž má operovat.<sup>11</sup>

### Rámec čtyř pilířů: Vibes, Specs, Skills a Agents

Profesionální nasazení AI v kódování se opírá o metodiku, která odděluje intuitivní interakci od exaktních instrukcí.<sup>14</sup> Tato hierarchie umožňuje týmům škálovat vývoj bez ztráty kontroly nad architekturou.

- Vibes (Interakční fáze):** Tato fáze slouží k intuitivnímu průzkumu a sdílení nápadů. Vývojář komunikuje s AI na vysoké úrovni, testuje koncepty a nechává si "skórovat" své nápady z hlediska proveditelnosti.<sup>14</sup>
- Specs (Specifikace):** Nejdůležitější prvek workflow. Jedná se o statické, lidsky čitelné Markdown soubory (např. v adresáři /specs), které definují "co" a "jak".<sup>14</sup> Obsahují bezpečnostní standardy, architektonické vzory a testovací strategie. V roce 2026 se prosazuje trend "co-evoluce" specifikací, kdy agent po opravě chyby automaticky navrhuje aktualizaci příslušné specifikace v souboru LessonsLearned.md, aby zabránil opakování stejného pochybení.<sup>14</sup>
- Skills (Dovednosti):** Modulární balíčky, které agentům poskytují specifické schopnosti, například jak nasadit aplikaci na konkrétní cloudovou platformu (např. Red Hat OpenShift) nebo jak provést bezpečnostní audit podle standardů dané firmy.<sup>14</sup>
- Agents (Agenti):** Samotné softwarové entity, jako je Cursor Composer nebo Claude Code, které interpretují specifikace a využívají dovednosti k vykonání práce.<sup>14</sup>

Tento strukturovaný přístup umožňuje dosáhnout přesnosti generování kódu přesahující 95 %, což je v ostrém kontrastu s neřízeným vibe coding, který často končí architektonickým kolapsem.<sup>14</sup>

## Srovnávací analýza vedoucích nástrojů roku 2026

Trh s nástroji pro vibe coding se stabilizoval do dvou hlavních kategorií: AI nativní editory (IDE) a autonomní tvůrci aplikací.<sup>16</sup> Volba nástroje v roce 2026 nezávisí pouze na "inteligenci" modelu, ale především na způsobu, jakým nástroj spravuje kontext projektu a jakou míru autonomie vývojář vyžaduje.<sup>18</sup>

### Cursor vs. Windsurf: Souboj o flow-state

Cursor a Windsurf představují špičku v kategorii profesionálních IDE. Oba jsou postaveny na základech VS Code, což umožňuje snadnou migraci rozšíření a nastavení, ale jejich filozofie práce s kontextem se liší.<sup>1</sup>

Vlastnost	Cursor (Composer/Agent Mode)	Windsurf (Cascade/Memories)
-----------	------------------------------	-----------------------------

<b>Správa kontextu</b>	Manuální přes @mentions (soubory, dokumentace, kódová báze) <sup>18</sup>	Automatické indexování a sledování akcí v reálném čase <sup>18</sup>
<b>Míra autonomie</b>	Střední až vysoká (vyžaduje potvrzení kroků) <sup>18</sup>	Extrémní (agent plánuje a provádí víceřadkové úkoly samostatně) <sup>18</sup>
<b>Silná stránka</b>	Iterativní vývoj s plnou kontrolou nad každým rozhodnutím <sup>18</sup>	Velké a komplexní repozitáře, kde je manuální výběr kontextu neefektivní <sup>18</sup>
<b>Unikátní funkce</b>	Vlastní model "Composer" optimalizovaný pro rychlost <sup>3</sup>	"Cascade Memories" – učení se vzorcům projektu v čase <sup>18</sup>
<b>Cenová politika</b>	\$20/měsíc (omezené kredity pro rychlé modely) <sup>3</sup>	\$15/měsíc (konkurenceschopnější pro jednotlivce) <sup>21</sup>

3

Zatímco Cursor je v roce 2026 preferován pro projekty vyžadující precizní dohled a kontrolu nad architekturou, Windsurf dominuje v situacích, kdy je potřeba rychle pochopit rozsáhlý legacy kód nebo provádět autonomní refaktorizace napříč desítkami souborů.<sup>18</sup> Pro vývojáře v terminálu se pak jako mocný nástroj prosadil Claude Code od Anthropicu, který vyniká v plánování a autonomním spouštění testů přímo v CLI prostředí.<sup>3</sup>

## Rizika a patologické jevy: Vibe Coding Kocovina

Fenomén "vibe coding kocoviny" (vibe coding hangover), poprvé popsany v roce 2025, se v roce 2026 stal hlavním tématem diskusí o efektivitě IT.<sup>8</sup> Projevuje se jako stav, kdy počáteční explozivní nárůst produktivity (až 300 %) je následně vykoupěn neúměrně vysokými náklady na údržbu, opravu chyb a řešení bezpečnostních incidentů.<sup>8</sup>

### Bezpečnostní krize a "Slopsquatting"

Bezpečnostní rizika spojená s AI generovaným kódem nejsou teoretická. V roce 2026 je dokumentováno, že AI modely mají tendenci k tzv. "halucinovaným bypassům" – náhodnému odstranění bezpečnostních kontrol, jako jsou autentizační middleware nebo validace vstupů, během refaktorizace.<sup>12</sup>

Zásadní hrozbou je "slopsquatting" (také známý jako AI supply chain attack). Vývojáři praktikující vibe coding často bezmyšlenkovitě přijímají návrhy na instalaci balíčků, které neexistují (např. pip install fastapi-security-pro). Útočníci začali tyto halucinované názvy registrovat v oficiálních repozitářích a plnit je malwarem. Pokud vibecoder nekontroluje, co instaluje, může do produkce nasadit kód s root přístupem pro kyberzločince.<sup>9</sup>

### Architektonický úpadek a technický dluh

AI exceluje v lokálních opravách, ale postrádá pochopení pro globální invarianty systému. Pokud je aplikace stavěna jako série izolovaných promptů, vzniká "architektura z duct tape" – kód, který sice běží, ale jehož vnitřní struktura je natolik roztržštěná, že jakákoliv další změna vede k nepředvídatelným kaskádovým chybám.<sup>15</sup> Forrester odhaduje, že do konce roku 2026 bude technický dluh generovaný AI nástroji konzumovat až 40 % rozpočtů na IT pouze v nákladech na údržbu.<sup>8</sup>

### Analýza dopadů nekontrolovaného vibecodingu

Oblast dopadu	Statistický následek v roce 2026	Příčina
Návratnost investic (ROI)	Snížení o 18–29 %	Vysoké náklady na následné opravy a debugging <sup>8</sup>
Čas na debugging	63 % vývojářů tráví více času opravami než psaním	Nizká kvalita AI výstupu bez dohledu <sup>6</sup>
Dovednosti juniorů	44 % pokles v chápání základů programování	Nadměrné spoléhání na AI bez studia principů <sup>8</sup>
Bezpečnost (Java/Web)	72 % selhání v bezpečnostních auditech	Reprodukce nezabezpečených vzorů z tréninkových dat <sup>13</sup>

6

## Kompetenční model vývojáře v roce 2026: Od programátora k architektovi

Aby vývojář v roce 2026 uspěl, musí se transformovat z "vibecodera" na "automatického programátora".<sup>9</sup> Tento rozdíl je definován hloubkou kontroly nad kódovou bází. Zatímco vibecoder je zákazníkem AI, který pasivně přijímá výsledky, automatický programátor je ředitelem, který AI využívá jako výkonný nástroj, ale sám si udržuje absolutní mentální mapu systému.<sup>9</sup>

### Nezbytné technické dovednosti

- Architektonická gramotnost:** Schopnost manuálně definovat rozhraní (interfaces), typy a systémové hranice dříve, než AI začne generovat implementaci. Platí pravidlo: "Piš architekturu, generuj implementaci".<sup>9</sup>
- Hloubková revize kódu:** Vývojář musí být schopen vysvětlit každý řádek AI generovaného kódu juniorovi. Pokud kód není schopen vysvětlit, nesmí být nasazen do produkce.<sup>9</sup>
- Pokročilý prompt engineering:** Přejít od vágních zadání k vrstveným promptům, které obsahují technický kontext, funkční požadavky, omezení a explicitní definici edge-cases (např. "Co se stane, když API neodpoví?").<sup>1</sup>
- Znalost bezpečnostních scannerů:** Integrace nástrojů jako Snyk, SonarQube nebo Checkmarx přímo do agentického workflow k okamžité validaci AI výstupů.<sup>12</sup>

### Psychologický přístup a mindset

Vítězové v éře vibecodingu se vyznačují specifickým přístupem k technologii, který kombinuje rychlost s extrémní skepsí.<sup>7</sup>

- Vlastnictví místo delegování:** I když AI píše kód, vývojář nese 100% zodpovědnost za jeho funkčnost a bezpečnost. AI je vnímáno jako "super-výkonný junior," který nikdy nespí, ale je schopen s naprostou jistotou navrhnout totální nesmysl.<sup>1</sup>
- Embrace the Struggle (Přijetí zápasu):** Skutečné učení se děje ve chvílích frustrace a hlubokého

debuggingu. Vibe Coding nesmí sloužit jako únik před nutností porozumět základům (např. jak funguje asynchronní I/O nebo správa paměti), jinak se vývojář stává nepoužitelným v momentě, kdy AI narazí na své limity.<sup>30</sup>

- **Taste (Vkus) a úsudek:** V roce 2026, kdy je kód komoditou, se nejcennější dovedností stává "vkus" – schopnost poznat, které řešení je elegantní, udržitelné a uživatelsky přívětivé.<sup>2</sup>

## Strategické řízení vibecodingu v organizaci

Pro firmy v roce 2026 již není otázkou, zda vibe coding povolit, ale jak jej bezpečně a efektivně řídit.<sup>6</sup> Úspěšné týmy implementují systémy "agentické správy" (agentic governance), které omezují autonomii AI tam, kde je riziko příliš vysoké.<sup>2</sup>

### Karanténa kódové báze (The Quarantine Principle)

Organizace rozdělují své systémy do zón podle rizika, což určuje míru povolené AI intervence<sup>9</sup>:

Zóna	Charakteristika	AI Pravidla
<b>Červená zóna</b>	Kritické systémy: Autentizace, platby, kryptografie, PII.	AI zakázáno nebo omezeno na čtení. Kód píše senioři manuálně. <sup>9</sup>
<b>Žlutá zóna</b>	Business logika, API endpointy, integrace.	AI asistuje pod přísným dohledem. Povinná 2-stupňová revize člověkem. <sup>9</sup>
<b>Zelená zóna</b>	Interní nástroje, prototypy, skripty pro zpracování dat.	Plný vibecoding povolen. Důraz na rychlost před dokonalou strukturou. <sup>9</sup>

### Implementace.cursorrules a standardizace promptů

Standardem pro rok 2026 je využívání konfiguračních souborů jako .cursorrules nebo .clauderules na úrovni projektu.<sup>22</sup> Tyto soubory definují "ústavu" projektu, kterou agenti nesmí porušit. Příklad moderního pravidla pro rok 2026:

#.cursorrules: Bezpečnostní a architektonické standardy

- VŽDY používej asynchronní operace pro I/O (Python 3.12 syntax).
- NIKDY nepoužívej hardkódované API klíče; vyžaduj environmentální proměnné.
- PŘED importem nového balíčku spust' npm list <balíček> k ověření jeho existence.
- ZÁKAZ používání TODO komentářů jako náhrady za implementaci.
- PŘED zápisem rozsáhlých změn (>3 soubory) vytvoř git checkpoint.

<sup>32</sup>

Tento přístup automatizuje vymáhání standardů a snižuje počet chyb v code review o 70 %.<sup>3</sup>

## Budoucí výhled: Směrem k agentickému marketingu a APA

Vibecoding v roce 2026 přestává být doménou pouze vývojářů a expanduje do obchodu a marketingu.<sup>31</sup>

Nastává éra "Agentic Process Automation" (APA), kde se procesy v organizaci samy adaptují na základě výsledků.<sup>31</sup>

Nemusíte již umět kódovat v tradičním smyslu, abyste vytvořili komplexní automatizaci, která propojí váš CRM se sociálními sítěmi a analytickými nástroji. Stačí popsat "ekosystém" a delegovat práci na orchestrátory jako n8n, Zapier Central nebo CrewAI.<sup>33</sup> Tato demokratizace tvorby softwaru však klade ještě větší důraz na architektonické myšlení – i nemarketingový manažer musí pochopit logiku datových toků, aby jeho "vibem vytvořená" kampaň nezpůsobila bezpečnostní incident.<sup>33</sup>

## Klíčové predikce pro zbytek roku 2026

- **Kód jako spotřební materiál:** Prototypy a internal tools budou vnímány jako jednorázové. Pokud přestanou vyhovovat, nebudou se refaktorovat, ale znovu "vyvibují" od nuly s novým kontextem.<sup>6</sup>
- **Vzestup "Agentic Governance":** Firmy budou investovat do specializovaných AI agentů, jejichž jedinou rolí je kontrolovat práci ostatních kódovacích agentů (např. agent "Security Auditor" vs. agent "Backend Architect").<sup>2</sup>
- **Změna vzdělávání:** Technické školy přestanou klást důraz na memorování syntaxe a zaměří se na systémový design, debugging a etiku AI.<sup>7</sup>

## Závěr a strategická doporučení

Vibecoding v roce 2026 není hrozbou pro kvalifikované inženýry, ale nemilosrdným sitem pro ty, kteří se odmítají vyvíjet.<sup>7</sup> Analýza ukazuje, že rychlost, kterou tyto nástroje nabízejí, je dvojsečnou zbraní: může vybudovat impérium za víkend, nebo zničit firmu během 72 hodin skrze fatální bezpečnostní chybu.<sup>8</sup>

Pro jednotlivce, kteří chtějí v tomto prostředí uspět, platí následující imperativy:

1. **Nespokojte se s plochou definicí vibecodingu.** Vibe Coding není o "kódování bez kódování," ale o precizní režii komplexních systémů skrze vyšší úroveň abstrakce.<sup>2</sup>
2. **Pěstujte architektonickou integritu.** Vaším úkolem není psát funkce, ale definovat, jak funkce spolupracují. Kód je halucinace, architektura je realita.<sup>15</sup>
3. **Důvěřujte, ale prověřujte.** Každý řádek kódu, který AI vygeneruje, je vaším dluhem, dokud jej nepochopíte a neschválíte.<sup>9</sup>
4. **Investujte do kontextu.** Kvalita vašeho produktu je přímo úměrná kvalitě vašich specifikací a rule-files. Špatný kontext vede k architektonickému chaosu.<sup>11</sup>

Platformy jako vibecoderi.cz jsou pouze špičkou ledovce. Skutečný vibe coding vyžaduje transformaci mindsetu z "dělníka kódu" na "command inženýra," který ovládá armádu digitálních asistentů s přisností a vizí architekta.<sup>2</sup> Pouze tak lze překonat "vibe coding kocovinu" a využít exponenciální potenciál AI k tvorbě softwaru, který je nejen rychlý, ale i bezpečný a udržitelný.<sup>6</sup>

### Citovaná díla

1. Vibe Coding in 2026: How AI Is Changing the Way Developers Write Code - Daily.dev, použito dubna 12, 2026, <https://daily.dev/blog/vibe-coding-how-ai-changing-developers-code>
2. Beyond Cursor: The "Vibe Coding" Stack That Will Dominate 2026 | by Techie Fellow, použito dubna 12, 2026, <https://medium.com/@techie.fellow/beyond-cursor-the-vibe-coding-stack-that-will-dominate-2026-01b590b09f80>
3. Best AI Coding Assistants 2026: Cursor, Copilot & More, použito dubna 12, 2026, <https://www.vibecodingacademy.ai/blog/best-ai-coding-assistant-2026>
4. Best practices for coding with agents - Cursor, použito dubna 12, 2026, <https://cursor.com/blog/agent-best-practices>

5. Co je Vibecoding? | Kompletní průvodce | Vibecoderi.cz, použito dubna 12, 2026, <https://vibecoderi.cz/o-vibecodingu>
6. The state of vibe coding in 2026: Adoption won, now what? - Hashnode, použito dubna 12, 2026, <https://hashnode.com/blog/state-of-vibe-coding-2026>
7. Why the Pichai-Vembu Vibe Coding Debate Is 2026's Literacy Test (And How to Pass It) | by Rekha Pankaj Kumar | CodeToDeploy - Medium, použito dubna 12, 2026, <https://medium.com/codetodeploy/vibe-coding-isnt-the-revolution-it-s-the-literacy-test-30928c4a41c0>
8. AI Development in 2026: Vibe Coding Guide by Tizbi Raleigh NC, použito dubna 12, 2026, <https://tizbi.com/articles/vibe-coding-business-guide-2026>
9. The Vibe Coding Hangover: A Senior Engineer's Survival Guide | by ..., použito dubna 12, 2026, <https://medium.com/@vishnukqcherupuzha/the-vibe-coding-hangover-a-senior-engineers-survival-guide-0c5bf051fffe>
10. Názory na "Vibe Coding je skutečné kódování" : r/ClaudeCode - Reddit, použito dubna 12, 2026, [https://www.reddit.com/r/ClaudeCode/comments/1re9k6n/opinions\\_on\\_vibe\\_coding\\_is\\_real\\_coding/?t=cs](https://www.reddit.com/r/ClaudeCode/comments/1re9k6n/opinions_on_vibe_coding_is_real_coding/?t=cs)
11. Vibe Coding Explained: Platforms, Prompts & Best Practices - Clarifai, použito dubna 12, 2026, <https://www.clarifai.com/blog/vibe-coding-explained>
12. How to Secure Vibe Coded Applications in 2026 - DEV Community, použito dubna 12, 2026, <https://dev.to/devin-rosario/how-to-secure-vibe-coded-applications-in-2026-208d>
13. The State of AI Code Security in 2026: What You Need to Know Right Now : r/vibecoding, použito dubna 12, 2026, [https://www.reddit.com/r/vibecoding/comments/1qa1pye/the\\_state\\_of\\_ai\\_code\\_security\\_in\\_2026\\_what\\_you/](https://www.reddit.com/r/vibecoding/comments/1qa1pye/the_state_of_ai_code_security_in_2026_what_you/)
14. Vibe, specs, skills, and agents: The four pillars of AI coding | Red ..., použito dubna 12, 2026, <https://developers.redhat.com/articles/2026/03/30/vibes-specs-skills-agents-ai-coding>
15. You can't vibe code your way out of a vibe coding mess - Weavy, použito dubna 12, 2026, <https://www.weavy.com/blog/you-cant-vibe-code-your-way-out-of-a-vibe-coding-mess>
16. Best Vibe Coding Tools in 2026 (Honest Ranking), použito dubna 12, 2026, <https://www.getautonoma.com/blog/best-vibe-coding-tools>
17. The 12 Best Vibe Coding Tools in 2026: Build Apps Without Writing Code - NinjaTech AI, použito dubna 12, 2026, <https://www.ninatech.ai/blog/the-12-best-vibe-coding-tools-in-2026>
18. Windsurf vs Cursor: Which AI Code Editor Wins for Vibe Coding in 2026?, použito dubna 12, 2026, <https://www.vibecodingacademy.ai/blog/windsurf-vs-cursor>
19. Cursor vs Windsurf: Which Code Editor Fits Your Workflow? [2025] - Blott, použito dubna 12, 2026, <https://www.blott.com/blog/post/cursor-vs-windsurf-which-code-editor-fits-your-workflow>
20. Your AI-coding tools buying checklist for 2026 - LeadDev, použito dubna 12, 2026, <https://leaddev.com/ai/your-ai-coding-tools-buying-checklist-for-2026>
21. Cursor vs. Windsurf: The Developer Civil War - Noca AI, použito dubna 12, 2026, <https://noca.ai/cursor-vs-windsurf/>
22. Cursor AI Best Practices: Complete Guide to Coding 10x Faster in 2026, použito dubna 12, 2026, <https://www.vibecodingacademy.ai/blog/cursor-ai-best-practices-guide-2026>
23. Cursor vs Windsurf vs GitHub Copilot - Builder.io, použito dubna 12, 2026, <https://www.builder.io/blog/cursor-vs-windsurf-vs-github-copilot>
24. Vibe Coding in 2026: How AI Is Changing the Way Developers Write Code - Daily.dev, použito dubna 12, 2026, <https://daily.dev/blog/vibe-coding-2026-ai-changing-how-developers-write-code>
25. Windsurf vs Cursor | AI IDE Comparison, použito dubna 12, 2026, <https://windsurf.com/compare/windsurf-vs-cursor>
26. Vibe Coding and the SaaS Shakeup: What Business Leaders Need to Know Before Building Custom Software | Cyber Unit, použito dubna 12, 2026, <https://cyberunit.com/insights/vibe-coding-saas-shakeup-security-risks/>
27. The vibe coding crisis: Why you need a dual-track engineering strategy | CIO, použito dubna 12, 2026, <https://www.cio.com/article/4155813/the-vibe-coding-crisis-why-you-need-a-dual-track-engineering-strategy.html>
28. Vibe Coding in 2026: A Working Developer's Honest Guide - Ricardo Gil, použito dubna 12, 2026, <https://www.gilricardo.com/blog/vibe-coding-2026-developers-honest-guide>
29. Roadmap to Vibe Coding in 2026 - Grabui, použito dubna 12, 2026, <https://grabui.com/blog/roadmap-to-vibe-coding>
30. How to Avoid "Vibe Coding Hell" - A Blog about Software Development, použito dubna 12, 2026, <https://mahdix.com/how-to-avoid-vibe-coding-hell>
31. 2026 Enterprise Predictions: Expect New Heights for Vibe Coding and Retaining Tribal Knowledge - IT Connection, použito dubna 12, 2026, <https://itcblogs.currentanalysis.com/2026/01/13/2026-enterprise-predictions-expect-new-heights-for-vibe-coding-and-retaining-tribal-knowledge/>
32. cursor-ai-tips/rules/cursorrules-2026-best-practices.md at main ..., použito dubna 12, 2026, <https://github.com/murataslan1/cursor-ai-tips/blob/main/rules/cursorrules-2026-best-practices.md>
33. Beyond the Prompt: Mastering Agentic Workflows and "Vibe Coding" in 2026 - Tutorials Dojo, použito dubna 12, 2026, <https://tutorialsdojo.com/beyond-the-prompt-mastering-agentic-workflows-and-vibe-coding-in-2026/>
34. Agentic AI Frameworks: Top 10 Options in 2026 - NetApp InstaClustr, použito dubna 12, 2026, <https://www.instaclustr.com/education/agentic-ai/agentic-ai-frameworks-top-10-options-in-2026/>
35. Invest in Cursor Rules: A Four-Level Maturity Framework - Guides, použito dubna 12, 2026, <https://forum.cursor.com/t/invest-in-cursor-rules-a-four-level-maturity-framework/148569>