

Pravidlá pre AI v tíme: šablóna

Čo agent nesmie, čo ho zastaví a kto pravidlá vlastní. Do vlastného repozitára za jedno popoludnie.

Pre CTO, tech leadov a vývojárov, ktorých tím už používa Claude Code, Codex, Cursor alebo Copilot. Každý po svojom. Táto šablóna z toho spraví jednu spoločnú vrstvu.

AI dnes píše všetok náš kód a za výsledok ručí senior. Od roku 2023 sme dodali 15+ produkčných projektov. Toto je rámec, ktorý pri tom používame.

1. Prečo pravidlá v texte nestačia

Zakážete agentovi otvoriť súbor. Neotvorí ho. Napíše si skript, ktorý ho vypíše.

Nie je to zlomyseľnosť. Agent dostal úlohu a hľadá cestu. Veta v prompte je pre neho prekážka, nie múr. Preto má každé pravidlo nižšie stĺpec, ktorý väčšina „AI politik“ nemá: čo ho naozaj zastaví.

Úroveň	Čo to znamená	Sila
Len veta	Pravidlo v CLAUDE.md alebo AGENTS.md. Agent ho väčšinou dodrží.	najslabšia
Nastavenie alebo hook	Príkaz sa zastaví skôr, než sa spustí.	stredná
CI	Zmena sa do hlavnej vetvy nedostane, kým brána nie je zelená.	silná
Lint alebo test	Build spadne a hláška povie prečo. Agent si ju prečíta a opraví sám.	najsilnejšia

Pravidlo, ktoré sa porušilo dvakrát, prestáva byť len veta. Posuňte ho o stupeň vyššie.

2. Osem vecí, ktoré agent nesmie

#	Agent nesmie	Prečo	Čo ho zastaví
1	Pripojiť sa na produkčnú databázu	Zlý dopyt na produkcii sa nedá vziať späť. Na diagnostiku stačí read-only replika alebo export logov.	nastavenie (zákaz) + oddelené prístupové údaje
2	Čítať secrety (.env, kľúče, tokeny)	Čo agent prečíta, je v kontexte, v logoch a niekedy aj v commite.	nastavenie + secrety mimo repozitára
3	Použiť produkčné prístupy do cloudu	Rovnaký dôvod ako databáza, väčší dosah.	nastavenie (zákaz)
4	Spustiť deploy alebo migráciu	Čo sa nedá vrátiť, ide cez človeka.	nastavenie (pýtať sa) + schválenie prostredia v CI
5	Prepísať históriu alebo pushnúť do main	Force-push a reset --hard potichu zničia cudziu prácu.	nastavenie + hook + ochrana vetvy na serveri

#	Agent nesmie	Prečo	Čo ho zastaví
6	Písať test, podľa ktorého ho hodnotíte	Agent, ktorý si píše test, si píše aj známku.	kontrola v CI + pravidlo pre reviewera
7	Recenzovať vlastnú zmenu	Jeden model, ktorý píše aj kontroluje, má tie isté slepé miesta dvakrát.	povinné schválenie PR človekom
8	Vložiť zákaznické dáta do promptu	Osobné údaje nepatria do nástroja, ktorý ste na ne neschválili.	len veta + dohoda v tíme

Pri ôsmom bode poctivo píšeme „len veta“. Mechanicky ho dnes poriadne nedrží nič. Pravidlo, ktoré priznáva svoju slabinu, je lepšie ako pravidlo, ktoré predstiera silu.

Koľko z vašich pravidiel je len veta? Toľko máte dier.

3. Kam ktoré pravidlo patrí

Agent načíta súbor s pravidlami pri každom spustení. Čím viac textu, tým horšie sa drží toho podstatného. Preto tri úrovne:

1. **Koreň repozitára:** krátky CLAUDE.md alebo AGENTS.md. Čo je toto repo, ako spustiť testy, čo agent nesmie, ako vyzerá hotová práca.
2. **Priečinok:** pravidlá jedného modulu v súbore priamo v ňom. Načítajú sa, až keď agent pracuje tam.
3. **Lint alebo test:** to najdôležitejšie. Nikto to nemusí čítať, zlú zmenu to zastaví samo.

Používate viac nástrojov? Pravidlá majte v AGENTS.md a v CLAUDE.md naň odkážte riadkom `@AGENTS.md`. Dve kópie sa rozídu skôr, než si to niekto všimne.

Štruktúra súborov

```

CLAUDE.md alebo AGENTS.md    krátky: repo, príkazy, odkazy
.claude/
  settings.json               zákazy v strojovej podobe
  hooks/push-gate.sh          to, čo nastavenie nevie
  rules/
    00-nikdy.md               8 zákazov + stípec "čo to drží"
    10-zadanie.md             čo musí mať úloha, kým ju agent začne
    20-brany.md               lint, typecheck, testy: blokuje áno/nie
    30-review.md              kto recenzuje a čo dostane
    90-vlastnictvo.md         kto pravidlá vlastní a ako pribudne nové

```

4. Vzory do repozitára

Príklady sú pre Claude Code. Princíp (zákaz, hook, CI, lint) platí aj pre iné nástroje, líši sa len zápis.

Zákazy v nastavení (body 1 až 5)

```
{
  "permissions": {
    "deny": [
      "Bash(psql:*)",
      "Bash(git push --force:*)",
      "Bash(git reset --hard:*)",
      "Read(.env.prod*)",
      "Read(**/*.pem)"
    ],
    "ask": [
      "Read(**/.env*)",
      "Bash(npx prisma migrate deploy:*)"
    ]
  }
}
```

Pozor: pravidlo kontroluje text príkazu, nie zámer. Ak má agent povolený `cat`, `cat .env` zákaz čítania obíde. Preto secrety patria mimo repozitára. Skripty ich načítajú, nikdy nevypisujú.

Hook proti pushu do main (bod 5)

```
#!/bin/bash
# .claude/hooks/push-gate.sh (PreToolUse, Bash)
# exit 0 = povoliť, exit 2 = zastaviť
CMD=$(jq -r '.tool_input.command // empty')
[[ "$CMD" != *"git push"* ]] && exit 0
case "$CMD" in
  *" main"*|" master"*)
    echo "Push do main je zakázaný. Otvorte PR." >&2
    exit 2 ;;
esac
exit 0
```

Dve chyby, ktoré hook potichu vypnú. `|| true` za hookom prepíše exit 2 na 0. A `2>/dev/null` zahodí hlášku, takže agent nevie, prečo neprešiel. Hook overte raz naostro. Na serveri zapnite ochranu vetvy, lebo hook beží len tam, kde je nainštalovaný.

Testy a review (body 6 a 7)

- **Test musí najprv zlyhať.** Na pôvodnom kóde. Inak dokazuje len, že agent súhlasí sám so sebou.
- **Testy zo zadania schvaľuje človek.** CI upozorní, keď PR mení testy aj kód naraz. Rozhoduje reviewer.
- **Kto písal, nerecenzuje.** Review spustíte v novom kontexte, ideálne iným modelom. Dostane diff a vaše pravidlá, nie konverzáciu s autorom.
- **CI musí blokovat.** Pipeline, ktorá len hlási, je dekorácia.

5. Odkiaľ brať nové pravidlá

Z review. Komentár, ktorý zaznie druhýkrát, nie je chyba vývojára. Je to diera v pravidlách.

Koľkokrát	Čo urobiť
Prvýkrát	Opraviť a ísť ďalej
Druhýkrát	Zapísať pravidlo pre agenta
Tretíkrát	Napísať lint alebo test, ktorý to chytí sám

Podmienka: review sa píše do pull requestu, nie len ústne na hovore. Z ústneho review nevznikne nič, z čoho by sa dalo pravidlo vytiahnuť.

6. Vlastníctvo a meranie

Pravidlá bez vlastníka zastarávajú za pár mesiacov. Agent ich bude ďalej poslúchať, aj keď už neplatia. Skopírujte si toto do [90-vlastnictvo.md](#) :

```
Vlastník: [meno]
Nové pravidlo: PR do .claude/rules/, schvaľuje vlastník.
Čistenie: prvý pondelok v mesiaci, 30 minút.
Rebrík: porušené dvakrát -> o stupeň vyššie
    len veta -> nastavenie alebo hook -> CI -> lint
Mesačne zapíšeme: koľko pravidiel je ešte len veta.
```

Tri čísla, ktoré stojí za to sledovať každý mesiac:

- **Komentáre z review na jeden PR.** Ak pravidlá fungujú, klesajú.
- **Koľkokrát brána zastavila zmenu.** A koľkokrát ju agent potom opravil sám.
- **Koľko pravidiel je ešte len veta.** To je zoznam práce na ďalší mesiac.

Počet commitov ani riadky kódu nemerajte. Tie AI nafúkne za sekundu.

7. Pracovný hárok pre tím (45 minút)

Vytlačte, dajte na stôl a vyplňte spolu.

#	Agent nikdy	Čo to drží? (veta / nastavenie / CI / lint)	Vlastník
1	Produkčná databáza, len replika alebo export		
2	Secrety mimo kontextu agenta		
3	Produkčné prístupy do cloudu		
4	Deploy a migrácia iba cez človeka		
5	Force-push a push do main		

#	Agent nikdy	Čo to drží? (veta / nastavenie / CI / lint)	Vlastník
6	Test, podľa ktorého ho hodnotíme		
7	Review vlastnej zmeny		
8	Zákaznícke dáta v prompte		
9			
10			

Hotové, keď pri každom riadku viete povedať, čo ho drží a kto ho vlastní.

Chcete pravidlá nastaviť s niekým, kto s nimi pracuje denne?

Za 15 minút s Filipom prejdeme váš vyplnený hárok. Povieme vám, ktoré dve pravidlá by sme posunuli vyššie ako prvé. Žiadna prezentácia.

Dohodnite si 15 minút s Filipom: digitaliko.com/sk/contact

Digitaliko s.r.o. · www.digitaliko.com · Softvér na mieru za týždne. Od prvého dňa váš.