

Pravidla pro AI v týmu: šablona

Co agent nesmí, co ho zastaví a kdo pravidla vlastní. Do vlastního repozitáře za jedno odpoledne.

Pro CTO, tech leady i vývojáře. Váš tým už používá Claude Code, Codex, Cursor nebo Copilot. Každý po svém. Tato šablona z toho vytvoří jednu společnou vrstvu.

AI dnes píše veškerý náš kód a za výsledek ručí senior. Od roku 2023 jsme dodali 15+ produkčních projektů. Toto je rámec, který při tom používáme.

1. Proč pravidla v textu nestačí

Zakážete agentovi otevřít soubor. Neotevře ho. Napiše si skript, který ho vypíše.

Není to zlomyslnost. Agent dostal úkol a hledá cestu. Věta v promptu je pro něj překážka, ne zeď. Proto má každé pravidlo níže sloupec, který většina „AI politik“ nemá: co ho doopravdy zastaví.

Úroveň	Co to znamená	Síla
Jen věta	Pravidlo v CLAUDE.md nebo AGENTS.md. Agent ho většinou dodrží.	nejslabší
Nastavení nebo hook	Příkaz se zastaví dřív, než se spustí.	střední
CI	Změna se do hlavní větve nedostane, dokud není brána zelená.	silná
Lint nebo test	Build spadne a chybová hláška řekne proč. Agent ji přečte a sám opraví.	nejsilnější

Pravidlo, které agent porušil dvakrát, přestává být jen větou. Posuňte ho o stupeň výš.

2. Osm věcí, které agent nesmí

#	Agent nesmí	Proč	Co ho zastaví
1	Připojit se k produkční databázi	Špatný dotaz na produkci nelze vzít zpět. Pro diagnostiku stačí replika pro čtení nebo export logů.	nastavení (zákaz) + oddělené přístupové údaje
2	Číst tajné údaje (.env, klíče, tokeny)	Co agent přečte, zůstane v kontextu, v logu a občas i v commitu.	nastavení + tajné údaje mimo repozitář
3	Použít produkční přístupy do cloudu	Stejný důvod jako u databáze, ale s větším dopadem.	nastavení (zákaz)
4	Spustit nasazení nebo migraci	Co nelze vrátit, musí projít přes člověka.	nastavení (dotaz) + schválení prostředí v CI
5	Přepsat historii nebo nahrát do větve main	Force-push a reset --hard potichu zničí cizí práci.	nastavení + hook + ochrana větve na serveru

#	Agent nesmí	Proč	Co ho zastaví
6	Psát test, podle kterého ho hodnotíte	Agent píšící vlastní test si sám píše i známku.	kontrola v CI + pravidlo pro recenzenta
7	Recenzovat vlastní změnu	Jeden model, který píše i kontroluje, má stejná slepá místa dvakrát.	povinné schválení PR člověkem
8	Vložit zákaznická data do promptu	Osobní údaje nepatří do nástroje, který jste k tomu neschválili.	jen věta + dohoda v týmu

U osmého bodu poctivě píšeme „jen věta“. Dnes ho mechanicky nic spolehlivě neudrží. Pravidlo přiznávající slabinu funguje lépe než pravidlo předstírající sílu.

Kolik z vašich pravidel je jen věta? Tolik máte děr.

3. Kam které pravidlo patří

Agent načte soubor s pravidly při každém spuštění. Čím více textu má, tím hůře se drží podstaty. Proto máme tři úrovně:

1. **Kořen repozitáře:** krátký CLAUDE.md nebo AGENTS.md. Co je toto repo a jak spustit testy. Co agent nesmí a jak vypadá hotová práce.
2. **Složka:** pravidla konkrétního modulu přímo v souboru v ní. Načtou se, až když agent pracuje v dané složce.
3. **Lint nebo test:** to nejdůležitější. Nikdo to nemusí číst, špatnou změnu zastaví sám.

Používáte více nástrojů? Pravidla mějte v AGENTS.md a v CLAUDE.md na něj odkažte řádkem `@AGENTS.md`. Dvě kopie se rozejdou dřív, než si toho někdo všimne.

Struktura souborů

```

CLAUDE.md   nebo   AGENTS.md   krátký: repo, příkazy, odkazy
.claude/
  settings.json                zázaky ve strojové podobě
  hooks/push-gate.sh           to, co nastavení neumí
  rules/
    00-never.md                8 zákazů + sloupec "co to drží"
    10-task.md                  co musí mít úkol, než ho agent začne
    20-gates.md                 lint, typecheck, testy: blokuje ano/ne
    30-review.md                kdo recenzuje a co dostane
    90-ownership.md            kdo pravidla vlastní a jak přibude nové

```

4. Vzory do repozitáře

Příklady uvádíme pro Claude Code. Princip (zákaz, hook, CI, lint) platí i pro jiné nástroje. Liší se pouze 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 příkazu, nikoli záměr. Pokud má agent povolený `cat`, příkaz `cat .env` zákaz čtení obejde. Proto tajné údaje patří mimo repozitář. Skripty je načítají, ale nikdy nevypisují.

Hook proti pushu do main (bod 5)

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

Pozor na dvě chyby, které hook potichu vypnou. Zápis `|| true` za hookem přepíše exit 2 na 0. Parametr `2>/dev/null` zahodí hlášku, takže agent neví, proč neprošel. Hook proto ověřte naostro. Na serveru zapněte ochranu větve. Hook totiž běží jen tam, kde je nainstalovaný.

Testy a review (body 6 a 7)

- **Test musí nejdřív selhat.** Na původním kódu. Jinak dokazuje jen to, že agent souhlasí sám se sebou.
- **Testy ze zadání schvaluje člověk.** CI upozorní, když PR mění testy i kód naráz. Rozhoduje recenzent.
- **Kdo psal, nerecenzuje.** Review spusťte v novém kontextu, ideálně jiným modelem. Dostane diff a vaše pravidla, ne konverzaci s autorem.
- **CI musí blokovat.** Pipeline, která jen hlásí, je pouhá dekorace.

5. Odkud brát nová pravidla

Z review. Komentář, který zazní podruhé, není chybou vývojáře. Je to díra v pravidlech.

Kolikrát	Co udělat
Poprvé	Opravit a jít dál
Podruhé	Zapsat pravidlo pro agenta
Potřetí	Napsat lint nebo test, který problém zachytí sám

Podmínka: review pište do pull requestu, ne ústně při hovoru. Z ústního review nevznikne nic, z čeho vytáhnete pravidlo.

6. Vlastnictví a měření

Pravidla bez vlastníka zastarají za pár měsíců. Agent je bude dál poslouchat, i když už neplatí. Zkopírujte si toto do [90-ownership.md](#) :

```
Vlastník: [jméno]
Nové pravidlo: PR do .claude/rules/, schvaluje vlastník.
Čištění: první pondělí v měsíci, 30 minut.
Žebřík: porušeno dvakrát -> o stupeň výš
        jen věta -> nastavení nebo hook -> CI -> lint
Měsíčně zapíšeme: kolik pravidel je ještě jen věta.
```

Tři čísla, která se vyplatí sledovat každý měsíc:

- **Komentáře z review na jeden PR.** Pokud pravidla fungují, jejich počet klesá.
- **Kolikrát brána zastavila změnu.** A kolikrát ji agent potom opravil sám.
- **Kolik pravidel je ještě jen věta.** To je seznam práce na další měsíc.

Počet commitů ani řádky kódu neměřte. AI je nafoukne za sekundu.

7. Pracovní list pro tým (45 minut)

Vytiskněte ho, položte na stůl a vyplňte společně.

#	Agent nikdy	Co to drží? (věta / nastavení / CI / lint)	Vlastník
1	Produkční databáze, jen replika nebo export		
2	Tajné údaje mimo kontext agenta		
3	Produkční přístupy do cloudu		
4	Deploy a migrace pouze přes člověka		
5	Force-push a push do main		

#	Agent nikdy	Co to drží? (věta / nastavení / CI / lint)	Vlastník
6	Test, podle kterého ho hodnotíme		
7	Review vlastní změny		
8	Zákaznická data v promptu		
9			
10			

Máte hotovo, jakmile u každého řádku víte, co ho drží a kdo ho vlastní.

Chcete pravidla nastavit s někým, kdo s nimi pracuje denně?

Za 15 minut s Filipem projdeme váš vyplněný list. Řekneme vám, která dvě pravidla bychom posunuli výš jako první. Žádná prezentace.

Domluvte si 15 minut s Filipem: digitaliko.com/cs/contact

Digitaliko s.r.o. · www.digitaliko.com · Software na míru za týdny. Od prvního dne váš.