

kapitola 26

Jak nepustíš agenta na produkci naslepo?

Vrátíš se pro kávu

„Přepiš celou práci s uživateli, ať je to čistší.“ Za dvacet minut hlásí: hotovo, přepsáno dvě stě řádků, jeden soubor smazán. Přihlášení nefunguje.

Půlka je vážně lepší. Druhá půlka tiše rozbila věci, co fungovaly roky.

*Rozdíl je jediný příkaz předem — **git commit**.*

Metr nad matrací

Odvaha není povaha. Odvaha je **levná, když existuje síť.**

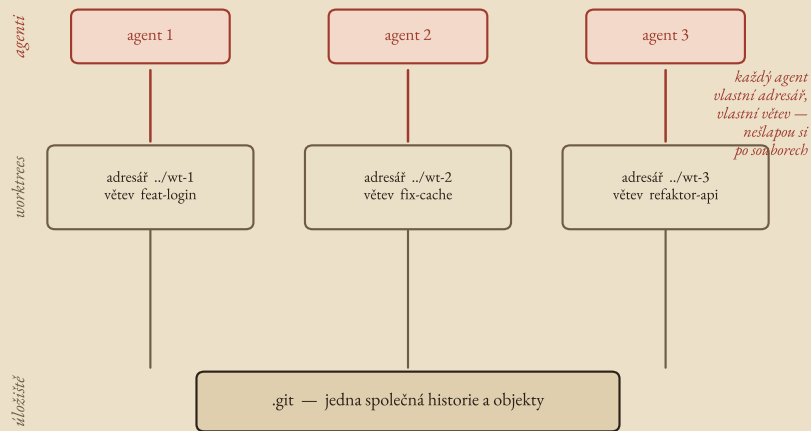
Provazochodec nad propastí je hrdina; metr nad matrací jen zkouší, kolik unese — spadne, vstane, zkusí to líp.

Git je ta matrace. **Dokud existuje undo, smíš zkoušet cokoli.**

Čtyři pojistky každý den

- **Commit** před každým „přepiš to celé“ — jako pás dřív, než nastartuješ.
- **.gitignore** — co agentovi ani gitu nikdy nedáš do ruky (klíče, hesla).
- **/diff** před každým souhlasem — přečti, co agent napsal.
- **Větev na experiment** — hlavní tok zůstane čistý.

Worktree: klec pro každého agenta



Jedno úložiště .git, tři adresáře na vlastních větvích — agenti si nepřepíší práci.

Plynulost není správnost

Model se nestydí a negratuluje si k nesmyslu. Napíše sebejistě funkci, co **vypadá hotově** — a tiše počítá slevu ze zlevněné ceny.

U agenta chybí i základ: neví, co je v projektu posvátné. Devět desetin správně, desetina katastrofa — a ta se pozná jen **čtením**.

Nekomitni nic, co bys neuměl vysvětlit. — Simon Willison

Když už to rozbil: bisect

Agent nadělal dvě stě commitů. Teď je tam bug, dřív nebyl. Který ho zavlekl?

Git to udělá **půlením**: každá odpověď „good/bad“ zahodí polovinu kandidátů. Z 256 commitů najdeš viníka na **osm testů**.

A s testem běží sám: `git bisect run pytest ...`

„Jak bych poznal, že se pletu?„

Než pustíš agenta na něco velkého: udělal jsem teď commit?

Kdyby to zkazil — ztratím minutu, nebo odpoledne? „Odpoledne“ = pustil jsi ho bezbranně.