

# **Daneel - Pipelines**

Martin Sukany

*February 21, 2026*

## Obsah

|                                  |          |
|----------------------------------|----------|
| <b>1 Přehled pipeline typů</b>   | <b>1</b> |
| 1.1 Tvorba obsahu                | 1        |
| 1.2 Výzkum a analýza             | 1        |
| 1.3 Plánování a architektura     | 2        |
| 1.4 Kód                          | 2        |
| 1.5 Kariéra a prezentace         | 3        |
| 1.6 Reflexe                      | 3        |
| 1.7 Zážitek pedagogika           | 3        |
| 1.8 Temná komora / Fotografie    | 4        |
| 1.9 Technické poznámky           | 4        |
| 1.10 Příklad: /pipeline research | 4        |

## 1 Přehled pipeline typů

Každá pipeline se spouští příkazem `/pipeline <typ> <téma>` v Matrixu. Výsledek dorazí zpět do Matrixu automaticky — bez dalšího zásahu.

Legenda: S sekvenční, P paralelní (workery běží současně), +pauza čeká na tvé schválení před pokračováním.

### 1.1 Tvorba obsahu

#### 1.1.1 /pipeline write <téma>

Tok: S (3 workery) Researcher → Creator → Critic

Critic ohodnotí výstup 0–100. Pokud skóre pod 70, Creator dostane zpětnou vazbu a zkusí to znovu (max 2×). Použití: blog posty, dokumentace, články, programové texty.

#### 1.1.2 /pipeline edit <text>

Tok: S (3 workery) Analyzer → Editor → Proofreader

Nejdřív analýza slabých míst, pak editace, pak jazyková korektura. Použití: vylepšení existujícího textu.

#### 1.1.3 /pipeline post <téma>

Tok: S (3 workery) Researcher → Creator → Editor

Creator připraví varianty pro blog, Twitter/X a LinkedIn najednou. Editor vybere, vyleští a zkontroluje délku. Použití: publikování obsahu na více platformách.

### 1.2 Výzkum a analýza

#### 1.2.1 /pipeline research <téma>

Tok: P→S (3 workery) ResearcherA || ResearcherB → Synthesizer

Dva researcheři pracují paralelně, každý z jiného úhlu. Synthesizer spojí výsledky. Použití: analýza tématu, technické srovnání, zjišťování faktů.

### 1.2.2 /pipeline digest <oblast>

Tok: P→S→S (4 workery) WebCollector || InternalCollector → Filter → QA

Sběr z webu a interních zdrojů, deduplikace a filtrování, finální QA kontrola. Použití: sumarizace z více zdrojů.

### 1.2.3 /pipeline compare <A vs B>

Tok: P→S (3 workery) AdvocateA || AdvocateB → Judge

Každá varianta má svého advokata. Judge rozhodne s oběma podklady. Použití: rozhodování mezi dvěma technologiemi, přístupy nebo řešeními.

## 1.3 Plánování a architektura

### 1.3.1 /pipeline plan <cíl>

Tok: P→S (3 workery) ScopeAnalyst || RiskAnalyst → PM

Paralelní analýza rozsahu a rizik, PM sestaví work items a milníky. Použití: rozložení projektu nebo cíle na konkrétní kroky.

### 1.3.2 /pipeline spec <feature>

Tok: S+pauza (4 workery) Analyzer → Architect → [schválení] → SpecWriter

Po návrhu architektury pipeline čeká na tvé schválení, pak teprve zapíše finální spec. Použití: technická architektura, design document, RFC.

### 1.3.3 /pipeline learn <technologie>

Tok: P→S (3 workery) ContentResearcher || PedagogyResearcher → Educator

Mapa konceptů a zdrojů + pedagogická struktura → 30denní plán s hands-on cvičeními. Použití: strukturované vzdělávání místo ad-hoc googling.

## 1.4 Kód

### 1.4.1 /pipeline code <co postavit>

Tok: S+pauza (5 workerů) Analyzer → Architect → [schválení] → Coder → Reviewer

Pipeline čeká na tvé schválení architektury před psaním kódu. Použití: nové nástroje, scripty, features.

### 1.4.2 /pipeline review <repo/soubor>

Tok: P→S (3 workery) Scanner || SecurityAuditor → Synthesizer

Oddělený code quality a security review běží paralelně. Použití: code review před merge, bezpečnostní kontrola.

### 1.4.3 /pipeline audit <cíl>

Tok: S→P→S (4 workery) Recon → Auditor || Remediator → Synthesizer

Nejkomplexnější pipeline. Recon zmapuje attack surface, pak Auditor a Remediator běží paralelně na stejném vstupu. Použití: bezpečnostní audit systému, kódu, infrastruktury, webu.

## 1.5 Kariéra a prezentace

### 1.5.1 /pipeline apply <pozice/firma>

Tok: P→S (3 workery) JobResearcher || CVAnalyst → Writer

Výzkum firmy a kultury + analýza tvého CV vs. požadavky → tailored cover letter a strategie přihlášky. Použití: přihlášky na pracovní pozice.

### 1.5.2 /pipeline pitch <nápad>

Tok: P→S (3 workery) AudienceResearcher || ContentStrategist → PitchWriter

Výzkum cílové skupiny a klíčová sdělení běží paralelně → finální pitch narrative. Použití: pitch pro kurz, projekt, startup, grant.

### 1.5.3 /pipeline workshop <téma/cílová skupina>

Tok: P→S (3 workery) ContentDesigner || FacilitatorAnalyst → WorkshopDesigner

Obsah a facilitátorské metody běží paralelně → kompletní program s tipy a záložními plány. Použití: design facilitátorského workshopu.

## 1.6 Reflexe

### 1.6.1 /pipeline retrospect <projekt>

Tok: S (3 workery) Historian → Analyst → Advisor

Fakta → hluboká analýza → konkrétní doporučení. Pro delší projekty. Použití: lessons learned po projektu nebo sprintu.

### 1.6.2 /pipeline debrief <událost>

Tok: S (2 workery) Historian → Analyst

Kratší než retrospect, zaměřený na konkrétní událost. Použití: reflexe po pohovoru, kurzu nebo meetingu.

## 1.7 Zážitková pedagogika

### 1.7.1 /pipeline pedagogy <kurz/téma>

Tok: P→S (3 workery) Audience Profiler || Dramaturg → Course Director

Audience Profiler zmapuje cílovou skupinu a profil absolventa. Dramaturg navrhne dramaturgickou křivku kurzu. Course Director (Opus) spojí oboje do kompletního programu s facilitátorským plánem a rizikovými scénáři. Použití: celý rámec kurzu od cílů po program.

### 1.7.2 /pipeline goals <téma/kurz>

Tok: S (2 workery) Goals Analyst → Profile Writer

Rychlá práce pouze s cíli — bez návrhu programu. Goals Analyst prozkoumá, čím účastník projde a co potřebuje zažít. Profile Writer (Opus) formuluje profil absolventa: znalosti, dovednosti, postoje a kritéria hodnocení. Použití: definování cílů a profilu absolventa bez návrhu konkrétního programu.

### 1.7.3 /pipeline program <blok/aktivita>

Tok: S (2 workery) Activity Designer → Facilitator Guide

Pro jeden konkrétní blok nebo aktivitu. Activity Designer navrhne strukturu, timing, materiály a variace. Facilitator Guide (Opus) připraví kompletní facilitátorský plán krok za krokem s tipy a záložními scénáři. Použití: příprava jedné konkrétní aktivity nebo programového bloku.

## 1.8 Temná komora / Fotografie

### 1.8.1 /pipeline darkroom <situace>

Tok: P→S (3 workery) Film Expert || Print Expert → Darkroom Advisor

Obecná konzultace pokrývající jak vyvolávání, tak zvětšování. Pro situace kdy potřebuješ komplexní pohled (např. volba kombinace film+vývojka+papír pro celý projekt). Použití: obecná konzultace k analogové fotografii.

### 1.8.2 /pipeline develop <film+vývojka>

Tok: S (2 workery) Film Analyst → Development Guide

Specializovaný výhradně na vyvolávání filmů. Film Analyst analyzuje kombinaci film+vývojka (HP5+, Delta, Rollei, Fomapan, Tri-X atd. × Rodinal, HC-110, D-76, DDX, Xtol, Pyrocat-HD) — poměry, teploty, časy, agitační protokol, push/pull. Development Guide (Opus) dodá přesný krok za krokem postup. Použití: příprava konkrétního vyvolávání — správné časy, teploty, protokol.

### 1.8.3 /pipeline print <motiv+papír>

Tok: S (2 workery) Print Analyst → Print Guide

Specializovaný na zvětšování a práci s papíry. Pokrývá Heiland TRD Splitgrade (kalibrace, histogram, poměr filtrů 0/5, test strip workflow), výběr papíru (Ilford MGIV/MG5, FOMA, Adox, RC vs FB), vývojky pro papír, dry-down efekt, dodging/burning. Použití: příprava zvětšovacího procesu, práce s Heiland Splitgrade, volba papíru.

### 1.8.4 /pipeline photo-theory <téma>

Tok: S (2 workery) Theory Researcher → Theory Guide

Pro teorii fotografie bez praktického výstupu. Pokrývá zonální systém (Ansel Adams), senzitivitu, H&D křivky, reciproitu, optiku, světlo, kompozici nebo libovolné téma. Theory Guide (Opus) dodá strukturovaný výklad od základů k pokročilému s praktickými důsledky. Použití: pochopení teorie — zonální systém, senzitivita, optika, estetika.

## 1.9 Technické poznámky

Výsledek každé pipeline dorazí přímo do Matrixu — bez nutnosti čekat nebo se ptát. Pipelines s pauzou (+pauza) čekají na tvé schválení a pak pokračují samy.

Pokud pipeline přerušena (restart systému), obnova:

```
/pipeline resume
```

### 1.10 Příklad: /pipeline research

```
/pipeline research "Kubernetes Operators - kdy má smysl psát vlastní vs. použít existující"
```

Co se stane:

1. Spuštění dva researcheři paralelně (~2 min)
2. Synthesizer čeká na oba, pak spojí výsledky (~2 min)
3. Výsledek dorazí do Matrixu — strukturovaná analýza s doporučením

Celková doba: ~4 minuty, bez zásahu.