

AI workflow, které se dá řídit

Celodenní workshop pro vývojáře a produktáky

Od zadání a rozpadu práce až po testy, review, rizika a zavedení do týmu.

První blok: workflow a zadávání práce AI

Druhý blok: kvalita, review, nástroje a rizika

Rytmus dne

9:30–10:00

Společný rámeček dne,
očekávání a výchozí situace
účastníků.

10:00–12:30

První blok: jak zadávat práci
AI a držet změny pod
kontrolou.

13:30–16:00

Druhý blok: jak výstup
kontrolovat, testovat a
bezpečně rozhodovat.

16:00–17:00

Diskuse nad vašimi
situacemi, workflow
blueprint a další kroky.

Celý den je postavený jako pracovní dílna: málo teorie, hodně rozhodování, rozpadů úkolů a komentovaných situací.

Jak budeme během dne pracovat

FORMÁT

- krátké rámování
- jeden konkrétní princip
- společný rozbor
- převod do vlastní praxe

CÍL

- ne „co všechno AI umí“
- ale **jak ji řídit**
- jak poznat, že pomáhá
- a jak rychle zastavit špatný směr

Neřešíme showreel. Řešíme workflow, které obstojí i po desáté iteraci.

PRVNÍ BLOK · 10:00–12:30

Jak zadávat práci AI a držet změny pod kontrolou

Od nápadu přes rozpad feature, kontext a issue až po malé změny, branch a feedback loop.

Co se v prvním bloku učíme

1. ZADÁNÍ

Jak převést vágní nápad na práci, kterou AI opravdu unese.

2. ŘÍZENÍ

Kde být hands-on a kde naopak nechat AI pracovat samostatně.

3. TEMPO

Jak držet změny malé, čitelné a levné na opravu.

Člověk není pisatel všeho. Je architekt směru.

AI zrychluje psaní.

Člověk drží smysl, hranice a prioritu.

směr

rozhodnutí

kontext

schválení

KDYŽ SE ROLE PLETOU

- člověk promptuje bez systému
- AI skládá architekturu po kouskách
- review přichází příliš pozdě
- diff je velký, ale nikdo neví proč

Od nápadu k merge

Nápad

Co přesně se má změnit a proč to má hodnotu.

Rozpad

Malé kroky, které se dají samostatně dodat a zkontrolovat.

Provedení

AI pracuje v jasném kontextu a s limity.

Kontrola

Testy, review, případně stop a návrat o krok zpět.

Špatný výsledek často nezačíná špatným modelem. Začíná nejasným zadáním.

Kde se AI workflow nejčastěji kazí

PŘÍLIŠ VÁGNÍ ZADÁNÍ

„Udělej onboarding lepší.“

PŘÍLIŠ VELKÝ ZÁBĚR

„Přepiš autentizaci, UI i analytics.“

PŘÍLIŠ MÁLO KONTEXTU

AI neví, co je kritické, co je zakázané a co se nesmí rozbít.

PŘÍLIŠ POZDNÍ REVIEW

Problém objevíte až ve chvíli, kdy je diff drahý na pochopení.

Zadání, které AI unese

Dobré zadání není dlouhé. Je rozhodnuté.

Musí být jasné:

- co se mění
- proč to děláme
- jak poznáme hotovo
- kde jsou hranice

MINIMUM, BEZ KTERÉHO NEZAČÍNAT

- cíl změny
- acceptance criteria
- omezení a rizika
- odkazy na relevantní části produktu

Každý task potřebuje tvar

KONTEXT

Kde problém vzniká a koho se dotýká.

VÝSLEDEK

Jaká změna má být po dokončení vidět.

DŮKAZ

Jak ověříme, že je výsledek správně.

Pokud neumíte vysvětlit task za minutu člověku, neumí ho dobře zpracovat ani AI.

Kontext, který šetří iterace

CO AI TYPICKY POTŘEBUJE

- relevantní soubory a moduly
- současné chování
- produktové omezení
- definici „co rozhodně ne“

CO DO KONTEXTU NEPATŘÍ

- všechnen obsah repozitáře
- neaktuální rozhodnutí
- duplicity
- detail, který nemá vliv na směr

Rozpad práce na malé změny

Malý task znamená malý diff.
Malý diff znamená levnější chybu.

PRAKTICKÉ PRAVIDLO

- jedna změna = jedna jasná odpověď
- nejdřív kostra, potom detail
- nejdřív bezpečné kroky, potom riskantní
- nejdřív vratitelné, potom nevratné

Branch a commit strategie

BRANCH PER ÚKOL

Jedna změna, jeden záměr, jedna stopa v historii.

COMMITY PO SMYSLU

Ne po minutách. Po logických krocích.

MAIN JAKO BEZPEČNÁ VĚTEV

Bez experimentů, bez rozpracovaných polotovarů.

Kdy nechat AI pracovat samostatně

ANO

- jasně ohraničený task
- známé prostředí
- silný feedback loop
- nízké produktové riziko

NE

- nejasné rozhodnutí
- vysoký dopad na architekturu
- neprobádaná část systému
- změna, kterou neumíme dobře vyhodnotit

Kdy má člověk vstoupit do řízení

PŘEDEM

Když je potřeba určit hranice, prioritu a trade-off.

V PRŮBĚHU

Když AI tápe, rozšiřuje scope nebo zavádí zbytečnou složitost.

NA KONCI

Když je potřeba potvrdit, že změna dává smysl i mimo kód.

Feedback loop je skutečný motor kvality

AI potřebuje rychlý signál, ne pozdní překvapení.

JAK MÁ VYPADAT ZDRAVÝ LOOP

- malý krok
- okamžitá kontrola
- oprava nebo potvrzení
- teprve potom další krok

Jak zastavit špatný směr včas

SIGNÁLY STOP

- scope roste bez dohody
- objevují se nové abstrakce
- diff je těžký na přečtení
- ztrácíme jistotu, co je vlastně hotovo

CO UDĚLAT

- vrátit se k cíli
- zmenšit task
- doplnit kontext
- rozdělit rozhodnutí a provedení

Produkt a vývoj musí být v jedné smyčce

PRODUKT PŘINÁŠÍ

hodnotu změny, prioritu, dopad na uživatele a hranice rozhodnutí

VÝVOJ PŘINÁŠÍ

proveditelnost, technická rizika, pořadí kroků a signály kvality

V AI workflow se rozhoduje častěji. Proto je důležité zkrátit vzdálenost mezi produktem a implementací.

Cvičení: rozpad feature

Vezmeme jednu reálnou feature a rozložíme ji na malé kroky, které může AI dodat bezpečně.

CO PŘI CVIČENÍ SLEDUJEME

- kde je první bezpečný krok
- co je už příliš velké
- kde chybí rozhodnutí
- jak by vypadal první task

Cvičení: z vágního zadání na říditelný task

START

„Ať je to chytřejší, rychlejší a víc AI.“

CÍL

jasný problém, jasný výsledek, jasné omezení, jasný první krok

Největší zrychlení často nevzniká lepším promptem. Vzniká lepší formulací práce.

První blok v jedné větě

SMĚR

Člověk drží cíl, hranice a prioritu.

TVAR

Task musí být malý, rozhodnutý a kontrolovatelný.

TEMPO

Postupujeme po malých změnách s rychlou zpětnou vazbou.

DRUHÝ BLOK · 13:30–16:00

Jak kontrolovat kvalitu, vybírat nástroje a pracovat s riziky

Od testů a review přes volbu modelů až po data, oprávnění, incidenty a týmové workflow.

Co se v druhém bloku učíme

1. KVALITA

Jak získat signály, že je změna opravdu bezpečná.

2. NÁSTROJE

Kdy sáhnout po jiném nástroji nebo modelu.

3. RIZIKA

Jak nevyrobiť rychlost za cenu chaosu a dluhu.

Kvalita není pocit. Je to sada signálů.

„Vypadá to dobře“ nestačí.

Potřebujeme více zdrojů jistoty najednou.

TYPICKÉ SIGNÁLY

- typy a statické kontroly
- unit a integrační testy
- e2e scénáře
- review diffu a chování

Typy a statické kontroly

PROČ JSOU SILNÉ

Vrací rychlou zpětnou vazbu, jsou levné a chytají velkou třídu chyb dřív, než začne review.

PROČ NESTAČÍ

Nehlídí business logiku, UX rozhodnutí ani skutečné chování systému v toku uživatele.

Unit testy jako guardrail

CHOVÁNÍ

Ověřují konkrétní očekávání na malé ploše.

EDGE CASES

Nutí nás myslet na rohy, které AI ráda přeskochí.

REGRESE

Chrání to, co už jednou fungovalo.

Integrační a e2e testy

CO ODHALÍ

- rozbitý tok mezi vrstvami
- špatné předpoklady o datech
- problém v reálném scénáři
- křehká místa, která unit test nevidí

JAK S NIMI PRACOVAT

- držet klíčové scénáře krátké
- nezkoušet pokrýt všechno
- mít několik kritických cest
- používat je jako pojistku před merge

Review AI změn je jiný druh review

Neptáme se jen „je to hezky napsané?“
Ptáme se „je to správné a nutné?“

TĚŽIŠTĚ REVIEW

- smysl změny
- pokrytí rizik
- zbytečná komplexita
- dopad na existující systém

Co hledat v diffu

PŘEKVAPENÍ

Změny mimo domluvený scope.

ORNAMENTY

Abstrakce, které nepřinášejí hodnotu.

DŮKAZ

Signál, že změna ob stojí i mimo demo.

Volba nástroje podle typu práce

RYCHLÉ EXPLORACE

Menší úkoly, orientace v kódu, první návrhy, lokální úpravy.

DELŠÍ ŘÍZENÁ PRÁCE

Větší úkoly s kontextem, více kroků, opakované iterace a potřeba držet workflow.

Nástroj nevybíráme podle brandu. Vybíráme ho podle druhu rozhodnutí, délky úkolu a ceny chyby.

Volba modelu podle rizika

LEVNĚJŠÍ A RYCHLÉ

Když zkoumáme, třídíme a iterujeme.

SILNĚJŠÍ MODEL

Když je potřeba hlubší uvažování nebo citlivé rozhodnutí.

PŘEPNUTÍ

Když vidíme, že současný model naráží na strop.

Kdy změnit nástroj nebo model

SIGNÁLY KE ZMĚNĚ

- opakované stejné chyby
- povrchní reasoning
- ztráta kontextu
- neschopnost držet zadání

SIGNÁLY, ŽE STAČÍ ZMĚNIT ZADÁNÍ

- scope je moc široký
- chybí omezení
- není jasný důkaz hotova
- mícháme rozhodnutí a implementaci

Data a oprávnění nejsou právní detail

RIZIKO

Citlivý obsah teče do nástroje, který nemá správně nastavené hranice, retention ani přístupy.

PRINCIP

AI workflow musí respektovat stejné hranice jako zbytek systému: kdo smí co vidět, měnit a exportovat.

Rizika v době AI vývoje

TECHNICKÁ

regrese, neudržitelná abstrakce, skrytá závislost

PRODUKTOVÁ

špatná priorita, falešné „hotovo“, přehlédnutý dopad na uživatele

PROCESNÍ

velké diffy, nečitelná historie, žádný audit rozhodnutí

Incidenty, rollback a návrat do bezpečí

Když se něco pokazí, nechceme hrdinství.
Chceme vratitelnost.

PROTO DRŽÍME

- malé změny
- samostatné branche
- jasné commity
- rychlý rollback

Co měřit, aby AI pomáhala produktu

RYCHLOST

čas od zadání k review, velikost diffu, počet iterací na dodání

KVALITA

regrese, rework po review, incidenty, důvěra týmu ve výstup

Pokud roste jen rychlost, ale ne čitelnost a jistota, nekupujete produktivitu. Kupujete budoucí dluh.

Jak to zavádět do týmu

Pilot

Jeden tým, jeden typ úkolu,
jeden jasný cíl.

Pravidla

Společný jazyk pro tasky,
review a stop signály.

Signály

Co měříme, co považujeme
za bezpečné a kdy
eskalujeme.

Škálování

Rozšiřujeme až to, co už
umíme řídit.

Cvičení: review a rozhodnutí

SPOLEČNĚ PROJDEME SITUACI

Je změna připravená na merge, nebo potřebuje další krok?

SLEDUJEME

- dostatečný důkaz kvality
- skryté riziko v diffu
- správnost zvoleného nástroje
- vhodný další krok

Vlastní workflow blueprint

CO ZADÁVÁM AI

Jaké typy úkolů jí dávají smysl už zítra.

CO CHRÁNÍM

Kde potřebuji silnější kontrolu, review a guardrails.

CO MĚNÍM

Jak bude odteď vypadat jeden zdravý pracovní cyklus.

Druhý blok v jedné větě

KVALITA

Potřebuje více signálů než jen „funguje to u mě“.

NÁSTROJE

Volíme podle druhu práce, ne podle popularity.

RIZIKA

Neodstraňujeme je slibem. Řídíme je workflow.

Diskuse, dotazy a přenos do praxe

Přineste vlastní situaci:

- kde se vám AI workflow dnes láme
- kde je největší tření mezi rychlostí a kvalitou
- který jeden krok chcete po dnešku změnit jako první

vlastní repozitář

vlastní issue

vlastní workflow