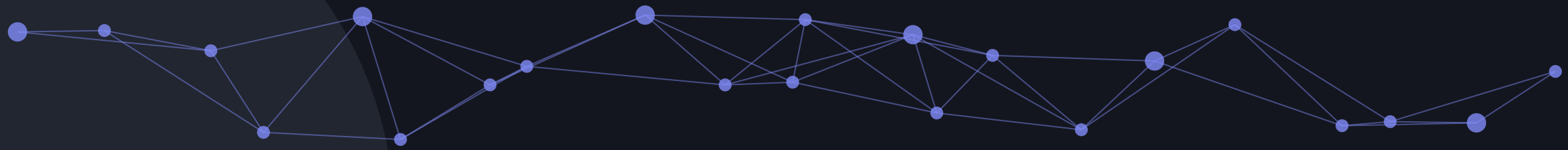
MATERIAŁY · MOŻLIWOŚCI AI W DEVELOPMENCIE

AI w Developmencie

Od pipeline'u technologicznego po gotowy produkt

Jak wykorzystać AI w tworzeniu na każdym etapie wytwarzania oprogramowania — szybciej, taniej i w sposób usystematyzowany. Materiał pokazujący możliwości i ścieżkę wdrożenia.

Cały pipeline | Testy i CI/CD | Migracje kodu | Wdrożenie i dokumentacja



Od pomysłu, przez kod, po utrzymanie — AI w każdym kroku.

PUNKT WYJŚCIA

Od AI „obok” do AI w całym pipeline

W wielu zespołach AI bywa używane doraźnie. Pełny potencjał odblokowuje wpięcie go w proces.

DZIŚ

- AI używane co najwyżej ad hoc — poza realnym kodem
- Brak integracji z CI/CD, testami i dokumentacją
- Wiedza o projekcie żyje w głowach, nie w narzędziach
- Proces i koszty pracy z AI — nieokreślone

PO WDROŻENIU

- AI częścią pipeline'u: projekt, testy, CI/CD, deployment
- Wiedza domenowa utrwalona w pliku Claude.md
- Powtarzalne testy i kontrola jakości — zautomatyzowane
- Kontrola kosztów i jeden standard pracy zespołu

GDZIE AI POMAGA — OBSZARY ZASTOSOWAŃ

Backend
API, logika, integracje

Frontend / UI
komponenty, ekrany

Embedded / IoT
firmware, sterowniki

Dane / ML
pipeline'y, analizy

DevOps / CI/CD
automatyzacja

OD CZEGO ZACZYNAMY

Cel: wprowadzić AI do realnego procesu — od razu dobrze

Cztery zasady, które ustawiamy od pierwszego dnia wdrożenia.

AI w realnym kodzie

Dziś AI co najwyżej w czacie — nie w faktycznym procesie wytwarzania.

Cel: AI realnie pracuje przy kodzie, testach, CI/CD i dokumentacji.

Systematycznie od startu

Wdrożenie bez procesu szybko rodzi chaos i złe nawyki.

Cel: jeden, powtarzalny sposób pracy z AI w całym zespole.

Kontrola kosztów od dnia zero

Bez limitów i telemetrii koszt potrafi wymknąć się spod kontroli.

Cel: widoczność i budżety zaprojektowane z góry — koszt przewidywalny.

Automatyzacja procesu

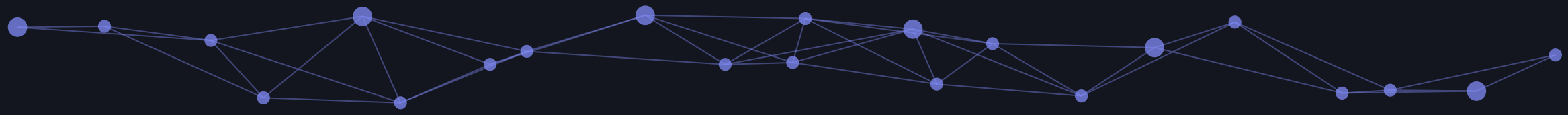
Powtarzalne zadania pochłaniają czas inżynierów.

Cel: zautomatyzować rutynę — testy, konfiguracje, dokumentację, deployment.

WIZJA

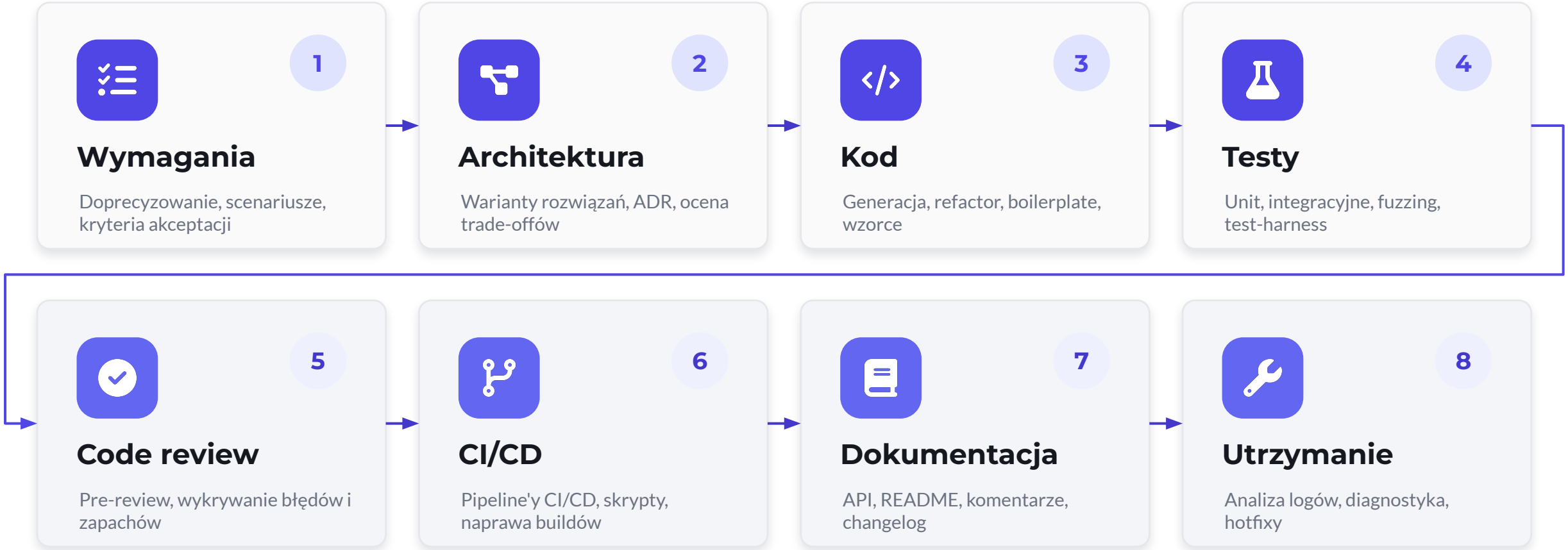
AI to nie chatbot obok procesu. To warstwa w całym pipeline'ie.

Największe efekty pojawiają się nie wtedy, gdy ktoś zapyta model o fragment kodu, ale gdy AI wspiera każdy etap — od wymagań, przez kod i testy, po review, wdrożenie i utrzymanie. Spójnie, mierzalnie i pod kontrolą kosztów.



PIPELINE TECHNOLOGICZNY

Gdzie AI pracuje na każdym etapie wytwarzania



PEŁEN PRZEKRÓJ

AI w całym stosie — od firmware po frontend

Backend i usługi

API, logika biznesowa, integracje, kolejki i obsługa błędów.

Frontend i UI

Komponenty, formularze, stany, dostępność, responsywność.

Systemy wbudowane / IoT

Firmware, protokoły, sterowniki, test-harness bez fizycznego HW.

Dane i ML

Pipeline'y danych, zapytania, transformacje, eksploracja i analizy.

DevOps i infrastruktura

IaC, pipeline'y CI/CD, konfiguracje, skrypty, diagnostyka.

Mobile i desktop

Ekrany, logika, integracje natywne, testy interfejsu.

Człowiek w pętli: AI przyspiesza inżyniera, ale go nie zastępuje. Decyzje krytyczne i zgodność zawsze weryfikuje człowiek.

STUDIUM PRZYPADKU

Od opisu do działającego kodu — z testami

GDZIE AI PRZYSPIESZA

Generowanie z opisu

Od wymagania w języku naturalnym do działającej funkcji i interfejsu.

Refactor i czyszczenie

Upraszczanie, usuwanie duplikacji, czytelniejsze nazwy i struktura.

Testy od ręki

Przypadki brzegowe, dane testowe, pokrycie tam, gdzie go brakuje.

Wzorce i boilerplate

Powtarzalny kod, schematy, walidacje — w sekundy zamiast godzin.

orders.ts · TypeScript

```
// AI: oblicz wartość koszyka z rabatem
export function total(items, code) {
  const sum = items.reduce(
    (a, i) ⇒ a + i.price * i.qty, 0);
  return applyDiscount(sum, code);
}
// AI: wygenerowany test brzegowy
expect(total([], null)).toBe(0)
```

AI pisze implementację i testy. Ty przeglądasz mały, czytelny diff i zatwierdzasz — kontrola zostaje po stronie zespołu.

INTERFEJSY I APLIKACJE

Od makiety do działającego interfejsu — web, desktop, mobile

AI skraca drogę od pomysłu do gotowego ekranu — niezależnie od platformy.

Generowanie UI

Od makiety do gotowych komponentów i ekranów — web, desktop i mobile.

Logika i walidacje

Stany, formularze, walidacje pól, parsery i kreatory ustawień.

Cross-platform

Wspólna baza kodu i spójny wygląd na wielu platformach naraz.

Testy E2E i UX

Automatyczne scenariusze klikania, walidacja formularzy, regresja wizualna.

Modernizacja w tle: starsze, ciężkie interfejsy można stopniowo przenosić na nowoczesny, lżejszy stos — obniżając koszt utrzymania, bez przepisywania wszystkiego naraz.

MIGRACJA TECHNOLOGICZNA

Przepisywanie kodu między językami — w dniach, nie miesiącach

Typowe ścieżki migracji, które AI radykalnie przyspiesza:

`JavaScript` ⇄	`TypeScript`	Typowanie, mniej błędów w czasie wykonania, lepsze podpowiedzi w IDE.
`Python 2` ⇄	`Python 3`	Wsparcie, bezpieczeństwo i dostęp do nowoczesnych bibliotek.
`Java` ⇄	`Kotlin`	Zwięzłość, bezpieczeństwo null, łatwiejsze utrzymanie.
`Monolit` ⇄	`Usługi`	Stopniowy podział na moduły i usługi — bez przepisywania całości.

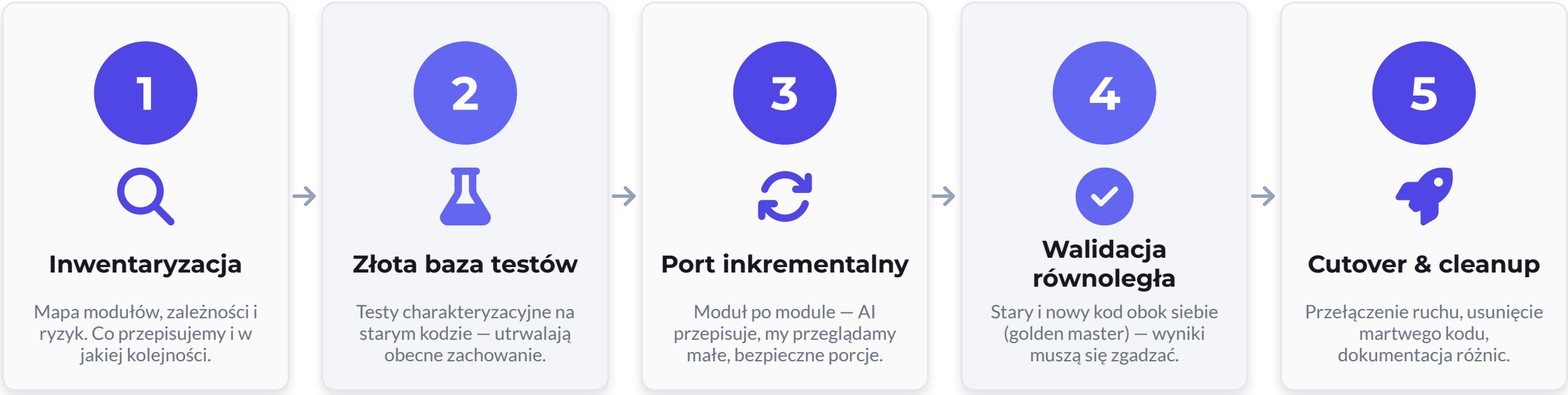
do 70%

krócej przy migracji sterowanej testami*

** Wartości orientacyjne — zależą od skali, jakości testów i złożoności kodu źródłowego.*

JAK ROBIMY TO BEZPIECZNIE

Migracja sterowana testami — bez utraty zachowania



Zasada: **AI pisze, testy decydują.** Zachowanie produktu jest chronione przez testy, a nie przez zaufanie do modelu.

KOSZTY POD KONTROLĄ

Optymalizacja kosztów — przestańcie zgadywać

Projektujemy kontrolę kosztów od pierwszego dnia — zanim zużycie zacznie rosnąć.

Routing modeli

Lekki model do prostych zadań, mocny tylko tam, gdzie naprawdę potrzeba.

Prompt caching

Wielokrotne użycie tego samego kontekstu = niższy koszt powtarzalnych zadań.

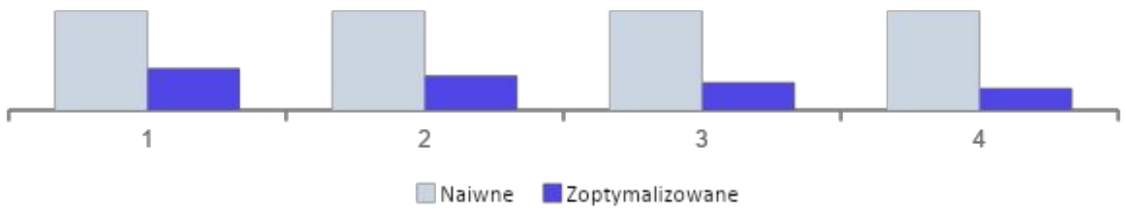
Budżety i limity

Limity per zespół / projekt z alertami, zanim koszt wymknie się spod kontroli.

Telemetria zużycia

Śledzenie zużycia per projekt i zespół — widać, gdzie realnie idą koszty.

Koszt na zadanie — podejście naiwne vs zoptymalizowane (ilustracyjnie)



Właściwe narzędzie do właściwego zadania

Claude Code do zadań agentowych w repo, czat do eksploracji, podpowiedzi w IDE do drobiazgów. Dobór narzędzia to często największa oszczędność.

JEDEN SPOSÓB PRACY

Ujednolicenie pracy z AI w całym zespole

Reguły repozytorium

Plik reguł (np. CLAUDE.md) z kontekstem projektu, konwencjami i stackiem — model od razu wie, jak u Was się pracuje.

Biblioteka promptów i skilli

Powtarzalne zadania jako gotowe szablony: generacja endpointu API, parser pliku, scenariusz testowy.

MCP i konektory

Podpięcie repozytoriów, baz i dokumentacji jako narzędzi AI — mniej kopiuj-wklej, więcej kontekstu.

Bramki w code review

AI wspiera review, ale zatwierdza człowiek. Jasne kryteria, co może, a co musi przejść przez inżyniera.

Playbook zespołu

Jeden dokument: kiedy, jak i czym korzystać z AI. Onboarding nowych osób w godziny, nie tygodnie.

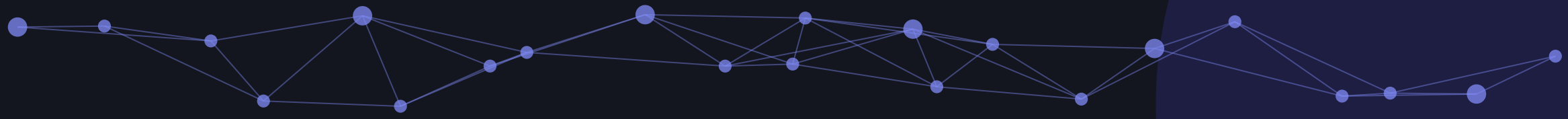
Zasady bezpieczeństwa

Co wolno wysłać do modelu, jak chronić dane wrażliwe i własność intelektualną.

CZĘŚĆ 2

Program szkoleniowy i wdrożenie in-situ

Od wspólnych podstaw, przez pracę na Waszych realnych projektach, po standard i kontrolę kosztów w całym zespole.



STRUKTURA PROGRAMU

Dwa tory: wspólny fundament i wdrożenie na produkcji

TOR A

Szkolenie początkowe

Wspólny język i fundament

- Warsztat dla całego zespołu
- Podstawy, prompting, narzędzia
- Workflowy pod Wasz stack
- Bezpieczeństwo i koszty

≈ 2 dni · cały zespół techniczny

TOR B

Wdrożenie in-situ

Akceleracja na realnych projektach

- Praca ramię w ramię na Waszych realnych projektach
- Budowa reguł, promptów i skilli
- Konfiguracja kontroli kosztów na żywo
- Pomiar efektów od 1. tygodnia

2–4 tygodnie · 1–2 projekty pilotażowe

TOR A · CZEGO NAUCZY SIĘ ZESPÓŁ

Sześć praktycznych kompetencji

1

Przygotowanie projektu pod AI

Struktura repo, kontekst i konwencje gotowe do efektywnej pracy z modelem.

2

CI/CD tworzone z AI

Pipeline'y CI/CD, skrypty buildów i automatyczna naprawa awarii buildu.

3

Testy generowane z AI

Unit, integracyjne i brzegowe — pokrycie tam, gdzie dziś go brakuje.

4

Deployment i dokumentacja

Automatyzacja wdrożeń oraz README, API i changelog generowane z AI.

5

Claude.md — kontekst i wiedza

Utrwalenie wiedzy domenowej o produktach w pliku, z którego korzysta AI.

6

Utrzymanie i diagnostyka

Analiza logów, debugowanie i szybkie hotfixy z pomocą AI.

Format hands-on · wszystko ćwiczone na Waszych repozytoriach i realnych zadaniach, nie na przykładach z internetu.

WIEDZA DOMENOWA

Claude.md — pamięć projektu i kontekst zespołu

Trwały kontekst dla AI

Jeden plik w repozytorium, który mówi modelowi, jak u Was się pracuje: stack, architektura, konwencje i słownik domenowy. Wiedza przestaje żyć tylko w głowach.

CO TO DAJE

- Spójność — każdy pracuje w tym samym kontekście
- Szybszy onboarding nowych osób — w godziny, nie tygodnie
- Mniej powtarzania kontekstu = niższy koszt
- Wiedza nie znika, gdy ktoś odchodzi z zespołu

CLAUDE.md

```
# Projekt: Platforma e-commerce (API + web)
## Stack
- Backend: TypeScript (NestJS)  ·  Web: React
## Architektura
- Usługi komunikują się przez REST i zdarzenia
## Słownik domenowy
- Koszyk: stan zamówienia przed płatnością
- SKU: unikalny identyfikator wariantu produktu
## Zasady
- Testy i lint przed mergem · konwencja commitów
- Zmiany w płatnościach zawsze przez review
```

CI/CD I JAKOŚĆ

Automatyzacja: od commita po produkcję

AI wpina się w cały proces dostarczania — od zmiany w repo, przez testy i build, po wdrożenie i obserwację.

Gdzie AI się wpina

- Generowanie i uzupełnianie testów
- Naprawa nieudanych buildów i konfiguracji
- Podsumowania pull requestów i wykrywanie regresji
- Generowanie release notes i dokumentacji zmian

Co zyskujecie

- Krótszy cykl od zmiany do wdrożenia
- Mniej ręcznej, powtarzalnej pracy
- Stabilniejsze, bardziej powtarzalne wydania
- Szybsza informacja zwrotna dla zespołu

TOR B · AKCELERACJA

Wdrożenie in-situ — na Waszych projektach

Praca ramię w ramię z zespołem

Nie ćwiczenia w oderwaniu od pracy. Wdrażamy AI bezpośrednio na realnych zadaniach z Waszych projektów — efekty widać od pierwszego sprintu.

JAK PRACUJEMY

- Pair-programming na bieżących zadaniach zespołu
- Ustawienie reguł repo i kontroli kosztów na żywo
- Budowa biblioteki promptów i skilli z realnych potrzeb
- Mierzenie czasu i jakości przed/po

CO ZOSTAJE W ORGANIZACJI

Playbook AI + Claude.md
Spisany standard pracy i utrwalony kontekst domenowy produktów.

Reguły repo + biblioteka promptów
Gotowe do użycia w codziennej pracy nad produktami.

Dashboard kosztów
Zużycie per projekt, limity i alerty — koszt przewidywalny.

Automatyzacja testów i CI/CD
Działające pipeline'y i testy regresyjne wpięte w proces.

HARMONOGRAM

Roadmapa — od audytu do skalowania

Faza 0

Audyt i cele

Diagnoza procesu, narzędzi i potrzeb

Tydzień 0

Faza 1

Szkolenie początkowe

Wspólny fundament dla zespołu

Tydzień 1 · ≈2 dni

Faza 2

Pilot in-situ

1–2 projekty, praca na żywo

Tydzień 2–4

Faza 3

Standaryzacja & koszty

Reguły, playbook, limity, telemetria

Tydzień 4–6

Faza 4

Skalowanie & governance

Cały zespół + utrzymanie standardu

Tydzień 6+

Tempo dopasowane do obciążenia zespołu — kamienie milowe ustalamy wspólnie na starcie.

CZEGO SIĘ SPODZIEWAĆ

Mierzalne rezultaty wdrożenia

Mniej rutyny

Zadania powtarzalne — testy, konfiguracje, dokumentacja — szybciej i z mniejszym wysiłkiem.

Koszty pod kontrolą

Zużycie widoczne per projekt, z limitami i alertami zamiast niespodzianek.

Jeden standard

Spójny playbook, reguły i prompty — cały zespół pracuje tak samo.

Migracje w dni

Przepisywanie między językami i modernizacja liczone w dniach, nie miesiącach.

Mierzymy od pierwszego tygodnia

Punkt odniesienia (baseline) ustalamy na starcie pilota — czas zadań, koszt, jakość. Konkretne wartości zależą od projektu; raportujemy realne dane, nie obietnice.

NASTĘPNE KROKI

Zacznijmy od pilota

Najszybsza droga do efektów: krótki warsztat startowy i wdrożenie na jednym realnym projekcie.

1 · Warsztat startowy

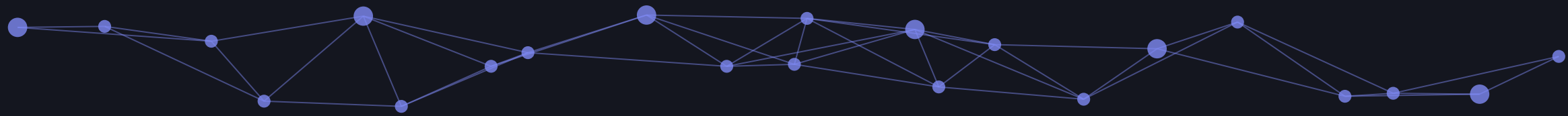
Wspólny fundament i język dla zespołu (≈4 dni).

2 · Wybór 1–2 projektów

Pilot in-situ na wybranym, realnym projekcie.

3 · Kontrola kosztów

Limity, telemetria i dobór narzędzi od pierwszego dnia.



KONTAKT

Porozmawiajmy o wdrożeniu

Pomożemy zaplanować i wdrożyć AI w Waszym zespole — od warsztatu po pełne wdrożenie in-situ.

Securinet

OSOBA KONTAKTOWA
Bartosz Sroka

E-MAIL
bartosz.sroka@securinet.pl

TELEFON
+48 696 236 88