AI dla nie-programistów

kilka praktycznych przykładów

Robert Olechowski | 2026

Jak używać tej prezentacji

📺 Tryb statyczny

Slajdy jako strona WWW.

🎛️ Tryb interaktywny

Prowadzący steruje slajdami na żywo.

⌨️ Skróty klawiszowe

→ / Space następny	← poprzedni	↓ / ↑ slajd dalej / wstecz (bez animacji)	Home początek prezentacji
[/] poprzednia / następna sekcja	F pełny ekran	O mapa slajdów	Esc przegląd / wyjście

Kontakt

Robert Olechowski

🌐 robertolechowski.com

📝 blog.robertolechowski.com

🐙 github.com/RobertOlechowski

📧 robertolechowski@gmail.com

Materiały do pobrania

prezentacje.robertolechowski.com

Materiały GitHub

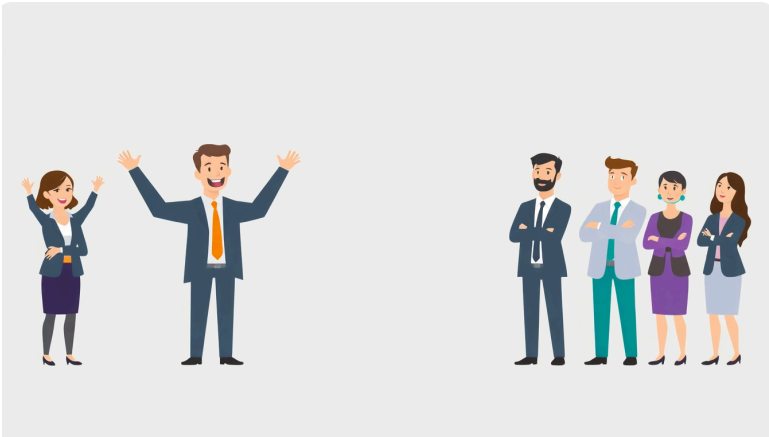
Usługi dla firm

- Szkolenia zespołów z AI
- Doradztwo we wdrażaniu AI
- Audyt i automatyzacja procesów
- Konsulting IT

Agenda

- Teoria i praktyka na przemian — minimum teorii, potem demo
  - Teoria #1 → **Demo #1** — Prasówka 2.0
  - Teoria #2 → **Demo #2** — NBP → skrypt → raport
  - Teoria #3 → **Demo #3** — raporty według szablonu
  - **Tool #1: Git** — wersjonowanie plików bez terminala
  - **Demo #4** — dane zza logowania (3 drogi)
  - **Tool #2: Markdown** — czysty tekst, natywny język AI
  - **Demo #5** — walka z gigantem (Emirates)
- Co zrobić w poniedziałek

Na tym spotkaniu są dwa obozy



🚀 Entuzjaści

„AI zmieni wszystko, zautomatyzujemy pół firmy!"

🛑 Sceptycy

„AI zmyśla, nie umie liczyć, to kolejny hype."

Obie strony mają rację — jednocześnie.

Nie jesteś spóźniony — **połowa jeszcze nie zaczęła.**

💬 **Napiszcie na czacie:** jedno zadanie w waszej pracy, którego szczerze nienawidzicie.

49%

pracowników w USA **nigdy** nie użyło AI w pracy

Gallup, Q4 2025

AI nie zastępuje ludzi

Zastępuje **nudną** część pracy

Teoria #1

Czym różni się **agent** od czatu — i z czego to zbudowane: **pamięć projektu**, **wtyczki (MCP)**, **skille**

Chat vs Agent

💬 Chat

Piszesz, on **odpowiada tekstem**.

Ty kopiujesz, wklejasz, klikasz, sprawdzasz.

🤖 Agent

Dostaje zadanie i **sam działa**:

czyta pliki, pisze skrypty, pobiera dane, tworzy raport.

🔁 **Pętla agentowa**

Czym to uruchomić?

claude.ai

Chat w **przeglądarce** — zero instalacji.

- internet, dokumenty, **Projects**
- na start: pytania, teksty, analizy

Claude Desktop

Aplikacja — Claude wchodzi na Twój komputer.

- przez **MCP** sięga do Twoich plików i narzędzi
- nadal klikasz jak w czacie

Claude Code

Agent w **terminalu** — piszesz mu polecenia.

- pracuje na całym **folderze projektu**
- najwięcej **kontroli**: widzisz i zatwierdzasz każdy krok

Plik CLAUDE.md — pamięć projektu

Zwykły plik tekstowy, który agent **czyta na starcie** każdej rozmowy.

Wpisujesz tam raz to, co zawsze musiałbyś powtarzać:

- czym jest projekt i jak jest zorganizowany
- zasady („pisz po polsku", „nie ruszaj tego folderu")
- konwencje, których ma się trzymać
- polecenia: jak uruchomić, zbudować i przetestować projekt
- stack i najważniejsze pliki — gdzie czego szukać
- czego unikać: częste pułapki i błędy z przeszłości

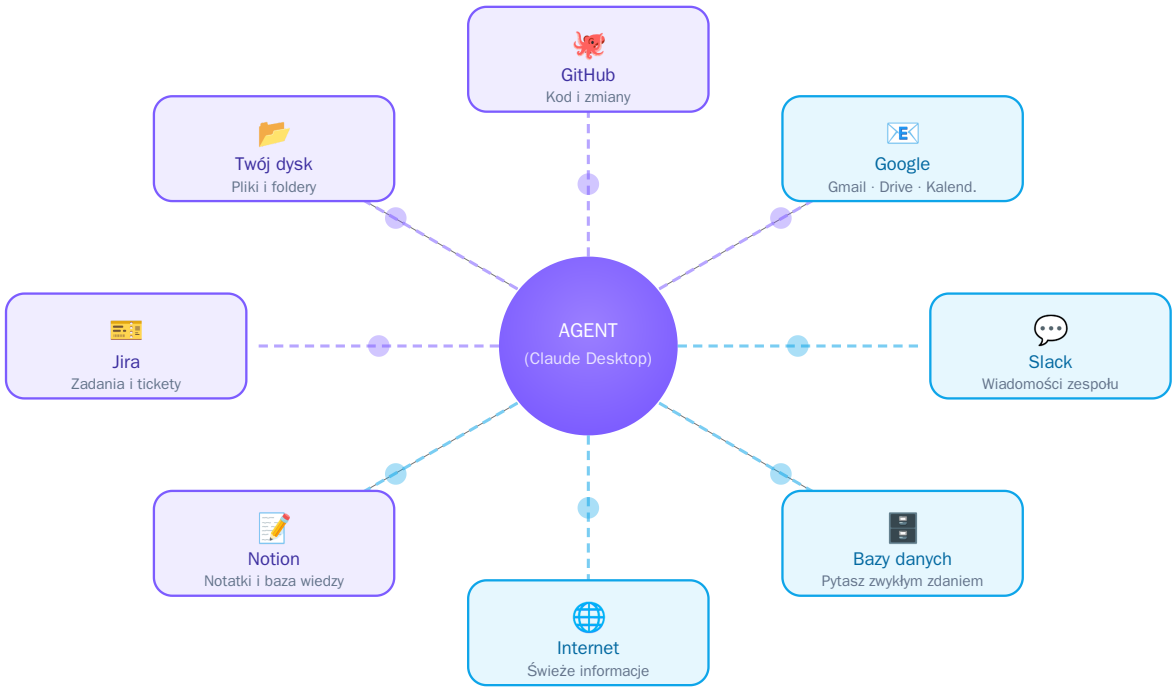
MCP — wtyczki do świata

MCP to standard, dzięki któremu podłączasz agenta do zewnętrznych narzędzi i danych — jak wtyczka albo przejściówka.

To **most** między modelem językowym a prawdziwym kodem: model mówi „chcę te dane", a po drugiej stronie program w języku programowania faktycznie je pobiera i oddaje.

Zamiast kopiować dane ręcznie — agent **sam sięga** tam, gdzie mu pozwolisz.

Co możesz podłączyć — MCP hub



Skill — zapisany prompt

Skill to gotowy, zapisany prompt z instrukcją „jak zrobić X" — plus trochę metadanych, żeby agent wiedział, **kiedy** go użyć.

Metadane skilla — to po nich agent wie, **kiedy** i **jak** go uruchomić:

Metadana	Co określa	Wymagane?
name	nazwa, po której go wywołujesz	✅
description	do czego służy i kiedy go użyć — po tym agent sam wybiera skill	✅
model	który model wykonuje: mocniejszy do trudnych, tańszy i szybszy do prostych	—
narzędzia *(allowed-tools)*	do czego skill ma dostęp — np. tylko czytanie plików, bez usuwania	—
wywoływalny ręcznie *(user-invocable)*	czy odpalasz go komendą, czy agent sięga sam	—

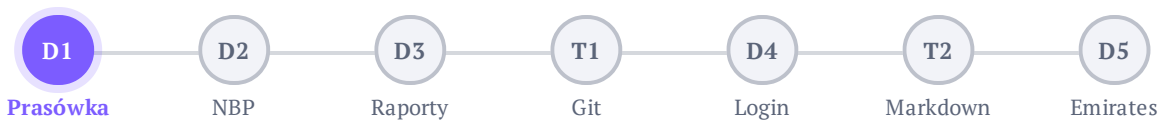
Do tego sama **treść** — instrukcja krok po kroku, co ma zrobić.

Demo #1

łatwe

Prasówka 2.0

Zbieranie informacji — prasówkę budujemy razem, na żywo



Krok 1 — krótki prompt, który pisze długi prompt

„Napisz mi **szczegółowy prompt** dla agenta, który codziennie rano robi **prasówkę**: przegląda wskazane źródła, wybiera newsy z tematów, które mnie interesują, i zapisuje **zwięzły raport po polsku**.

Prompt ma być na tyle dokładny, żeby agent nie musiał dopytywać. **Dopytaj mnie**, jeśli czegoś Ci brakuje."

Nie piszę instrukcji ręcznie.

Proszę AI, żeby napisało instrukcję za mnie.

Krok 2 — poprawiamy wygenerowany prompt

prasowka-prompt.md — to napisał agent

```
# Prasówka poranna — instrukcja
Jesteś asystentem researchu; rano
przygotowujesz prasówkę.
## Zakres tematyczny
Interesują mnie tematy:
- biznes i rynki
- polityka i ekonomia
- Bliski Wschód
- transport
## Źródła
Z listy źródeł.
RSS → strona → API → wyszukiwarka.
## Zasady
- Po polsku, własnymi słowami; z linkiem.
- Grupuj po temacie, deduplikuj.
## Wynik
Raport: nagłówek → kategorie → newsy.
```

...a to moja wersja — trzy skreślenia i dopiski

```
# Prasówka poranna — instrukcja
Jesteś asystentem researchu; rano
przygotowujesz prasówkę.
## Zakres tematyczny
Interesują mnie WYŁĄCZNIE tematy:
- biznes i rynki
- polityka i ekonomia
- sytuacja w Zatoce Perskiej
- transport morski / żegluga
## Źródła
Z listy w `data.yaml`.
RSS → strona → API → wyszukiwarka.
## Zasady
- Po polsku, własnymi słowami; z linkiem.
- Grupuj, deduplikuj. Maks. 48 h.
## Wynik
Zapisz reports/RRRR-MM-DD.md; 3–5 zdań/news.
```

Krok 3 — każę agentowi zrealizować ten prompt

„Weź instrukcję z `prasowka-prompt.md` i zbuduj z niej działającą prasówkę: przygotuj konfigurację, a powtarzalną procedurę zapisz jako **skill**, żebym mógł ją uruchamiać jednym poleceniem."

Agent sam zamienia prompt w **strukturę projektu:**

- `CLAUDE.md` — zasady (z promptu)
- `data.yaml` — lista źródeł
- `skills/prasowka/` — procedura krok po kroku

Z **jednego pliku tekstowego** powstaje agent, który wie *co* robić i *jak* to powtarzać.

Krok 4 — podajemy źródła, które nas interesują

data.yaml — lista źródeł (7)

```yaml
sources:
  # Biznes / polityka / ekonomia
  - { name: "Reuters", topics: [biznes, polityka] }
  - { name: "Bloomberg", topics: [ekonomia, rynki] }
  - { name: "Politico Europe", topics: [polityka] }
  - { name: "Bankier.pl", topics: [ekonomia, biznes] }
  # Zatoka Perska / Bliski Wschód
  - { name: "Al Jazeera", topics: [zatoka-perska] }
  # Transport morski / żegluga
  - { name: "Lloyd's List", topics: [żegluga] }
  - { name: "TradeWinds", topics: [żegluga, frachty] }

settings:
  max_article_age_hours: 48
```

7 źródeł, 5 tematów: biznes · polityka · ekonomia · Zatoka Perska · transport morski

Czy w ogóle się dobijemy? — test dostępu

Agent próbuje pobrać każde źródło: najpierw za darmo, potem płatnym API

Źródło	1. Ręczny fetch (darmowy)	2. Płatne API (MCP)
Politico Europe	✅ pobrane	—
Bankier.pl	✅ pobrane	—
Al Jazeera	✅ pobrane	—
Lloyd's List	⚠️ paywall	✅ firecrawl
Reuters	❌ `robots.txt` blokuje boty	✅ **firecrawl (stealth)**
Bloomberg	❌ `403 Forbidden`	✅ **firecrawl (stealth)**
TradeWinds	❌ tylko wizytówka	✅ **firecrawl** (nagłówki)

`robots.txt` to wpis na stronie mówiący botom „nie wchodź", a `403 Forbidden` to odmowa dostępu od serwera.

Konfiguracja MCP

.mcp.json — cały „montaż" narzędzia

```
{
  "mcpServers": {
    "firecrawl": {
      "command": "npx",
      "args": ["-y", "firecrawl-mcp"],
      "env": {
        "FIRECRAWL_API_KEY": "fc-••••••••"
      }
    }
  }
}
```

„Mam klucz API do Firecrawl: `fc-••••••••`. Skonfiguruj **Firecrawl jako MCP** w tym projekcie."

„Jakie **MCP** masz teraz dostępne? Wypisz listę i napisz na co pozwalają."

„Przetestuj Firecrawl — pobierz jakąś **przykładową stronę** i pokaż wynik."

MCP — Firecrawl

Zamienia stronę WWW w **czysty markdown** dla agenta. Tryb **stealth** omija `robots.txt`, `403` i część paywalli. Kosztuje **5×** więcej.

Darmowy plan (1 000 kredytów / mies.) starcza na prasówkę; płatne od 16 $. Granice prawne i ToS (regulamin serwisu) — wracamy do tego przy demo #4.

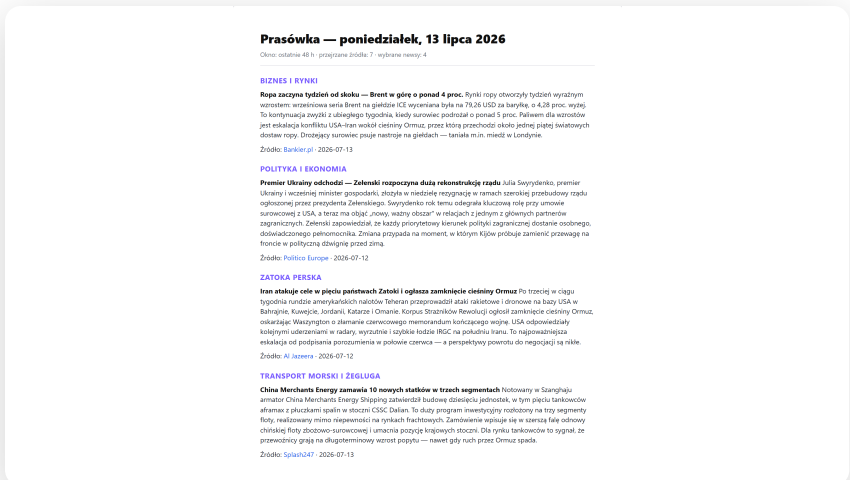
Ile kredytów kosztuje jedno zapytanie

Operacja	Kredyty
Scrape — zwykła strona	**1** / strona
Map — mapa sitemap	**1** / strona
Stealth — omija blokady	**5** / strona
Extract — LLM + schemat	**5** / strona
JSON / Enhanced Mode	**+4** / strona

Reuters i Bloomberg = tryb stealth → **5 kredytów** za każdą stronę zamiast 1.

Cyklicznie — codziennie o tej samej porze, bez nas

„Uruchamiaj skill **prasowka** codziennie o **12:00** i zapisuj raport do `reports/`. W poniedziałki dołóż krótkie podsumowanie całego tygodnia."



Prasówka — poniedziałek, 13 lipca 2026

Okno: ostatnie 48 h · przejrzane źródła: 7 · wybrane newsy: 4

BIZNES I RYNKI

Ropa zaczyna tydzień od skoku — Brent w górę o ponad 4 proc. Rynki ropy otworzyły tydzień wyraźnym wzrostem: wrześniowa seria Brent na giełdzie ICE wyceniana była na 79,26 USD za baryłkę, o 4,28 proc. wyżej. To kontynuacja zwyżki z ubiegłego tygodnia, kiedy surowiec podrożał o ponad 5 proc. Paliwem dla wzrostów jest eskalacja konfliktu USA–Iran wokół cieśniny Ormuz, przez którą przechodzi około jednej piątej światowych dostaw ropy. Drożejący surowiec psuje nastroje na giełdach — tamiała m.in. miedź w Londynie.

Źródło: Bankier.pl · 2026-07-13

POLITYKA I EKONOMIA

Premier Ukrainy odchodzi — Zełenski rozpoczyna dużą rekonstrukcję rządu Julia Swyrydenko, premier Ukrainy i wcześniej minister gospodarki, złożyła w niedzielę rezygnację w ramach szerokiej przebudowy rządu ogłoszonej przez prezydenta Zełenskiego. Swyrydenko rok temu odegrała kluczową rolę przy umowie surowcowej z USA, a teraz ma objąć „nowy, ważny obszar" w relacjach z jednym z głównych partnerów zagranicznych. Zełenski zapowiedział, że każdy priorytetowy kierunek polityki zagranicznej dostanie osobnego, doświadczonego pełnomocnika. Zmiana przypada na moment, w którym Kijów próbuje zamienić przewagę na froncie w polityczną dźwignię przed zimą.

Źródło: Politico Europe · 2026-07-12

ZATOKA PERSKA

Iran atakuje cele w pięciu państwach Zatoki i ogłasza zamknięcie cieśniny Ormuz Po trzeciej w ciągu tygodnia rundzie amerykańskich nalotów Teheran przeprowadził ataki rakietowe i dronowe na bazy USA w Bahrajnie, Kuwejcie, Jordanii, Katarze i Omanie. Korpus Strażników Rewolucji ogłosił zamknięcie cieśniny Ormuz, oskarżając Waszyngton o złamanie czerwcowego memorandum kończącego wojnę. USA odpowiedziały kolejnymi uderzeniami w radary, wyrzutnie i szybkie łodzie IRGC na południu Iranu. To najpoważniejsza eskalacja od podpisania porozumienia w połowie czerwca — a perspektywy powrotu do negocjacji są nikłe.

Źródło: Al Jazeera · 2026-07-12

TRANSPORT MORSKI I ŻEGLUGA

China Merchants Energy zamawia 10 nowych statków w trzech segmentach Notowany w Szanghaju armator China Merchants Energy Shipping zatwierdził budowę dziesięciu jednostek, w tym pięciu tankowców aframax z płuczkami spalin w stoczni CSSC Dalian. To duży program inwestycyjny rozłożony na trzy segmenty floty, realizowany mimo niepewności na rynkach frachtowych. Zamówienie wpisuje się w szerszą falę odnowy chińskiej floty zbożowo-surowcowej i umacnia pozycję krajowych stoczni. Dla rynku tankowców to sygnał, że przewoźnicy grają na długoterminowy wzrost popytu — nawet gdy ruch przez Ormuz spada.

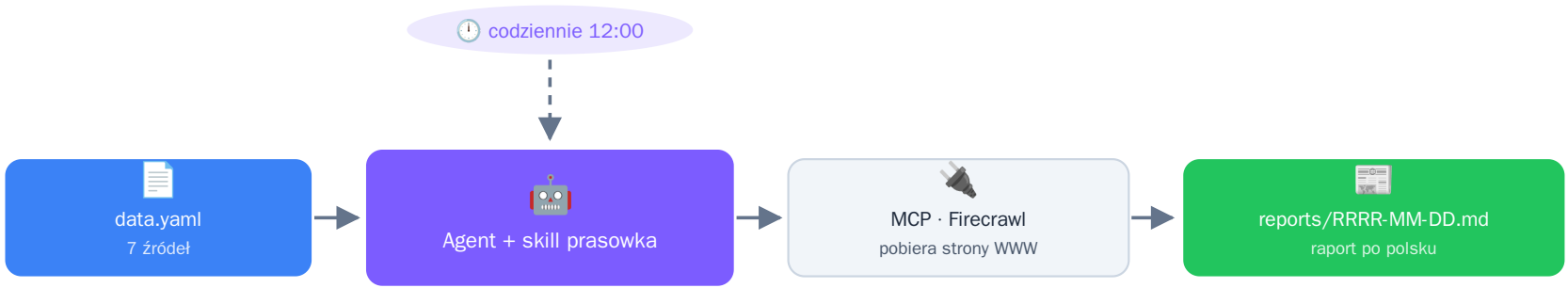
Źródło: Splash247 · 2026-07-13

Prasówka to już nie **polecenie**, tylko **usługa** działająca w tle.

Jedno zdanie ustawia harmonogram. Bez klikania, bez pilnowania, bez „przypomnę sobie rano".

Harmonogram to zaplanowane zadanie agenta, zapisane w systemie. Wykonuje je twój komputer, musi więc być wtedy włączony.

Cała prasówka na jednym obrazku



Od trzech zdań do usługi działającej w tle

Krok po kroku, bez jednej linijki kodu

1. **Krótki prompt** pisze długi, dokładny prompt
2. **Poprawki**: tematy, limit 48 h, format pliku
3. Prompt → **struktura projektu**: CLAUDE.md, data.yaml, skill
4. **7 źródeł** w zwykłym pliku tekstowym
5. Agent **sam dobrał narzędzia**, eskalował do płatnych
6. **Harmonogram**: prasówka budzi się bez nas

🎬 Demo — Pokaz na żywo

Na co zwrócić uwagę:

- agent **sam eskaluje** do płatnego API, gdy darmowe źródło się blokuje
- z trzech zdań powstaje **cykliczna usługa**, nie jednorazowa odpowiedź

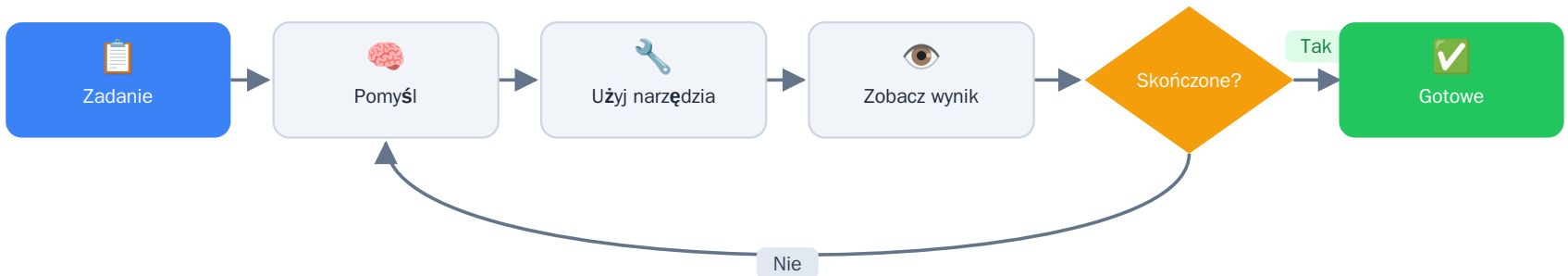
Teoria #2

Jak agent naprawdę pracuje — **pętla agentowa** krok po kroku

Model to silnik — nie cały agent

Ten sam agent może pod spodem **wymienić model** — mocniejszy do trudnych zadań, tańszy do prostych.

Pętla agentowa



Najczęstszy błąd początkującego

„Zrób mi raport" — **bez kontekstu** → agent zmyśla.

Traktuj agenta jak **nowego pracownika pierwszego dnia**:

- jasne polecenie
- przykład, jak ma wyglądać wynik
- dane, na których ma pracować
- daj dostęp do zbioru instrukcji

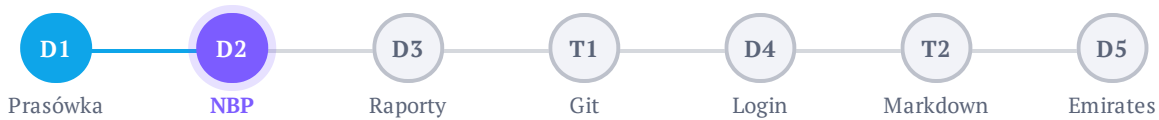
Im lepszy kontekst, tym lepszy wynik.

średnie

Demo #2

NBP → skrypt → raport

AI słabo liczy → AI pisze skrypt, a skrypt liczy



AI zmyśla liczby

„Jaki był **średni kurs EUR/PLN** w pierwszej połowie 2026 roku i o ile się zmienił?"

Model odpowiada **z pamięci**. Liczba brzmi wiarygodnie, ale nie umiemy powiedzieć **jak została policzona** — z jakich danych, z jakiego dnia, jakim wzorem. Może być zmyślona, może być sprzed roku. **Zero łańcucha audytu.**

„Nie odpowiadaj z pamięci. **Napisz skrypt**, który to policzy. Rozbij zadanie na dwa kroki:

1. Pobranie danych — skrypt ściąga kurs EUR/PLN z oficjalnego API i zapisuje surowe dane do CSV na dysk.

2. Raport — drugi skrypt czyta ten CSV i generuje raport z wykresem i statystykami."

Krok 1 — skrypt do pobierania danych

„Napisz skrypt w Pythonie, który pobiera z **oficjalnego API NBP** kurs **EUR/PLN** za pierwsze półrocze 2026 i zapisuje surowe dane do **CSV**. Potem policz na tych danych średnią, min, max i zmianę w okresie."

```python
import urllib.request, json, csv, statistics

URL = ("https://api.nbp.pl/api/exchangerates/"
       "rates/a/eur/2026-01-01/2026-07-10/?format=json")
rates = json.load(urllib.request.urlopen(URL))["rates"]

# 1. surowe dane lądują na dysku
with open("eur_pln.csv", "w", newline="") as f:
    w = csv.writer(f); w.writerow(["data", "kurs"])
    for r in rates:
        w.writerow([r["effectiveDate"], r["mid"]])

# 2. liczy SKRYPT — nie model
mid = [r["mid"] for r in rates]
print("notowań:", len(mid))
print("średnia:", round(statistics.mean(mid), 4))
print("min/max:", min(mid), "/", max(mid))
print("zmiana:", round((mid[-1]/mid[0]-1)*100, 2), "%")
```

Krok 2 — skrypt do generowania raportu

„Napisz **osobny skrypt**, który czyta `eur_pln.csv` i generuje **raport HTML** (`raport.html`).

Wykres — liniowy, w HTML (inline SVG, bez PNG):

- **dane liniowe**: kurs EUR/PLN dzień po dniu,
- **szare tło** pola wykresu, **czarne linie podziałki** (siatka),
- **wygładzona linia trendu** — średnia krocząca za **10 dni**.

Tabela — **agregacja miesięczna** (dla każdego miesiąca: średnia, min, max, liczba notowań) oraz **statystyki okresu**: średnia, mediana, min, max, odchylenie standardowe, zmiana %."

```
import csv, statistics as st

rows  = list(csv.DictReader(open("eur_pln.csv")))
kursy = [float(r["kurs"]) for r in rows]

# wygładzona linia trendu — średnia krocząca 10 dni
trend = [st.mean(kursy[max(0, i-9):i+1]) for i in range(len

# wykres jako inline SVG (szare tło, siatka, 2 linie),
# agregacja miesięczna → tabela, statystyki okresu…
svg   = rysuj_wykres(kursy, trend)   # ← ~15 linii niżej
tabela = agreguj_miesiacami(rows)    # ← ~5 linii niżej

open("raport.html", "w", encoding="utf-8").write(
    f"<h1>Kurs EUR/PLN — I półrocze 2026</h1>{svg}{tabela}"
```

całość: **~40 linii**, napisane w jednej odpowiedzi

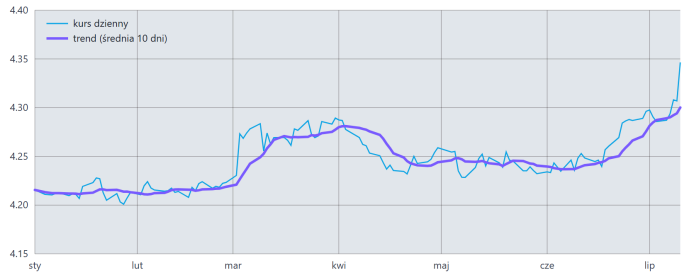
Krok 3 — gotowy raport

`raport.html` to **samodzielny plik** — otwierasz w przeglądarce, nic więcej nie trzeba. Skrypt z Kroku 2 wstawił do niego:

- **nagłówek** z okresem i źródłem danych,
- **wykres SVG** — kurs + wygładzony trend, w środku pliku,
- **tabelę** z agregacją miesięczną,
- **statystyki** okresu policzone z danych.

Kurs EUR/PLN — I półrocze 2026

Okres: 2026-01-02 → 2026-07-10 · Źródło: NBP, tabela A (api.nbp.pl) · Notowań: 132 · Wygenerowano: 2026-07-13



Agregacja miesięczna

Miesiąc	Średnia	Min	Max	Liczba notowań
styczeń 2026	4.2126	4.2009	4.2279	20
luty 2026	4.2174	4.2080	4.2241	20
marzec 2026	4.2718	4.2304	4.2894	22
kwiecień 2026	4.2529	4.2320	4.2874	21
maj 2026	4.2417	4.2284	4.2551	20
czerwiec 2026	4.2583	4.2334	4.2963	21
lipiec 2026	4.3020	4.2857	4.3465	8

Statystyki okresu

Średnia	Mediana	Min	Max	Odch. standardowe	Zmiana w okresie
4.2467	4.2435	4.2009	4.3465	0.0285	+3.11%

Krok 4 — audyt: nowa sesja, świeży agent

„Przejrzyj i **zaudytuj** wszystkie skrypty w tym folderze. Szukaj błędów: czy liczby liczą się poprawnie, czy wykres nie wprowadza w błąd, czy nie ma ukrytych założeń.

Nie ufaj, że działa — zweryfikuj. Wypisz znalezione problemy z numerami i zaproponuj poprawki."

Dlaczego **nowa sesja**? Ten agent nie widział, jak skrypty powstawały — nie broni własnej roboty.

Niezależna, druga para oczu, tyle że też AI. Jak w zespole: raport idzie do recenzji, zanim pójdzie dalej.

Krok 5 — audyt coś znalazł i poprawił

AUD-01 · BŁĄD (wykres kłamie). Punkty SVG rozstawia `X(i)` — **po numerze wiersza**, nie po dacie → weekendy i luki znikają, 3-tygodniowa przerwa wygląda jak jeden dzień. Wykres zniekształca **tempo** zmian.

AUD-02 · RYZYKO (ciche założenie). „Zmiana w okresie" bierze `mid[0]` i `mid[-1]` — zakłada, że API zawsze zwraca dane **po kolei**. Zmieni się kolejność → liczba będzie zła.

Poprawił i pokazał efekt. Oś X liczona po **dacie**, nie po numerze wiersza: luki i weekendy wracają na wykres. Dane sortowane po dacie przed liczeniem zmiany.

Zbudowaliśmy narzędzie, nie jednorazową odpowiedź

1. **Liczy skrypt, nie model**: zero halucynacji w cyfrach i na wykresie
2. **Świeży agent zaudytował kod**: znalazł błąd i go poprawił
3. Skrypty zostają: za miesiąc **jedno polecenie** odświeża raport

🎬 Demo — Pokaz na żywo

Na co zwrócić uwagę:

- liczby liczy **skrypt, nie model** — surowe dane lądują na dysku
- świeży agent w **nowej sesji** audytuje kod i znajduje błąd
- powstaje **narzędzie**: za miesiąc jedno polecenie odświeża raport

Teoria #3

Gdzie AI zawodzi — halucynacje, słaba pamięć, niedeterminizm — i jak każde z tych ograniczeń obejść

Granice AI — i jak je obchodzimy

Słabość	Obejście
Halucynacje — zmyśla fakty	Każ mu pokazać źródło, drugi agent sprawdza
Słabo liczy — myli arytmetykę	AI pisze skrypt, **skrypt liczy**
Zapomina — gubi się w długiej rozmowie	Pamięć w plikach, nowa sesja na nowe zadanie
Niedeterminizm — za każdym razem inaczej	Szablon + skill + skrypt
Dane firmowe — RODO, chmura USA	Zostają lokalnie / on-prem

Ograniczeń nie usuwasz — **obchodzisz je narzędziami albo procedurą wokół modelu.**

Okno kontekstu — AI „zapomina" w długiej rozmowie

Model ma ograniczoną **pamięć roboczą**: okno kontekstu. Wszystko, co czyta i pisze w rozmowie, musi się w nim zmieścić. **To okno bywa różne** — nawet w rodzinie Claude:

- **tańsza wersja** — krótszy kontekst (**200 tys.** tokenów): standard w modelach Claude
- **wersja 1M** — dłuższy kontekst (**1 mln** tokenów): Sonnet / Opus / Fable 5, więcej materiału naraz, ale drożej

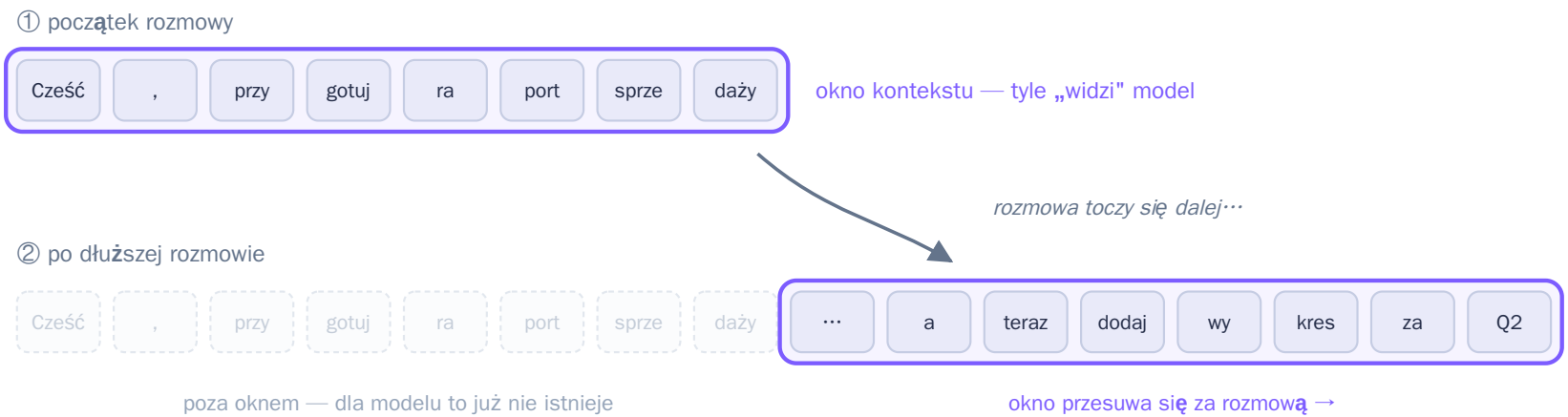
Niezależnie od modelu — im dłuższa sesja, tym więcej **szumu**: stare wątki zagłuszają obecne zadanie. Model nie powie „nie pamiętam" — po prostu zaczyna **gubić szczegóły**.

Obejście: pamięć trzymasz **na dysku**, nie w rozmowie:

- ustalenia i zasady → `CLAUDE.md`
- wyniki pracy → pliki: CSV, skrypty, raporty
- nowe zadanie → **nowa sesja**, czysty kontekst

Okno kontekstu — przesuwający się horyzont

Rozmowa to ciąg **tokenów**. Model widzi tylko to, co mieści się w oknie — reszta znika za horyzontem.



200 tys.

tokenów **Claude Opus**, ~**1,5 tomu**
Harry’ego Pottera

token ≈ 0.75 słowa · cała seria (7 tomów) ≈ 1,4 mln tokenów

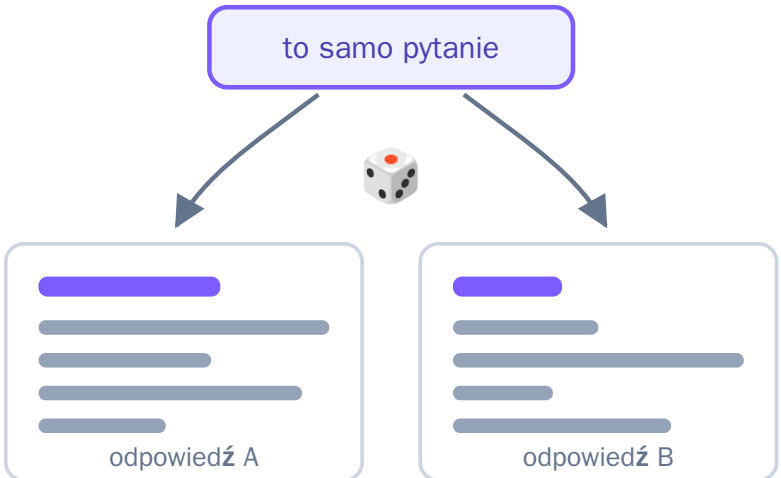
Ile mieści się w oknie? — miarą Harry Potter

Token ≈ 0.75 słowa. Cała seria *Harry'ego Pottera* (7 tomów) to ~1,1 mln słów ≈ **1,4 mln tokenów**.

Model	Okno kontekstu	To mniej więcej…
Claude Haiku 4.5	200 tys. tokenów	~**1,5 tomu** Harry'ego Pottera
Claude Sonnet / Opus / Fable 5 — **tańsza wersja**	200 tys. tokenów	~**1,5 tomu**
Claude Sonnet / Opus / Fable 5 — **wersja 1M**	1 mln tokenów	~**cała seria** (7 tomów)
GPT-5.2	400 tys. tokenów	~**3 tomy**
Gemini 3 Pro	1 mln tokenów	~**cała seria** (7 tomów)

Nawet „milion tokenów" ma dno. Im pełniejsze okno, tym większa **degradacja**: model gubi szczegóły ze środka.

Niedeterminizm — to samo pytanie, inna odpowiedź

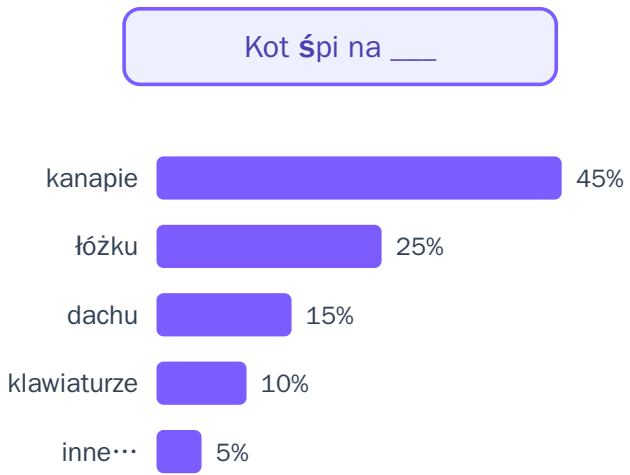


W burzy mózgów to zaleta. W **raporcie co miesiąc** — problem.

Co ma być stałe	Czym to pilnujesz
liczby	**skrypt**
procedura	**skill**
układ i wygląd	**szablon**

Temperatura — skąd bierze się ta losowość

Model nie „wybiera" słowa — dla każdego możliwego **tokenu** liczy prawdopodobieństwo, a potem **losuje**:



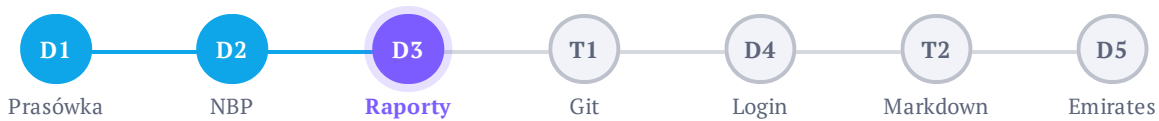
Temperatura to pokrętło tego losowania:

Demo #3

średnie

Raporty według szablonu

Ten sam raport co miesiąc — jednym poleceniem



Dwa klocki, z których składa się raport

Skill — zapisana procedura pobrania najnowszego pliku z danymi. Zawsze aktualne źródło, zero szukania po stronach.

Szablon — struktura raportu: sekcje, wykresy, styl wniosków. Zawsze ten sam układ i brand, niezależnie od miesiąca.

Demo #2: walka o **pobranie danych** (raz).

Demo #3: dane mamy, walczymy o **powtarzalność** — ten sam raport **co miesiąc**.

Koniec z „za każdym razem inny format". Raport na **dzień** staje się **jednym poleceniem + przeglądem**.

Krok 1 — skille opisujesz słowami, nie kodem

„Zrób **skill** `pobierz-sprzedaz-gus`. Ma wejść na stat.gov.pl, znaleźć najnowsze opracowanie o sprzedaży detalicznej i pobrać plik **XLSX** do `dane/`. Opisz kroki tak, żeby dało się je **powtórzyć co miesiąc**."

„Zrób drugi **skill** `generuj-raport`. Bierze świeże dane z `dane/` i **wypełnia szablon raportu**: liczy dynamiki, robi wykresy, pisze wnioski. Wynik zapisuje w **Markdownie**."

„Zrób trzeci **skill** `raport-do-pdf`. Bierze gotowy raport w Markdownie i składa go do **PDF** z logo i kolorami firmy, zapis w `raporty/`."

Nie otwieram edytora kodu. **Opisuję procedurę po ludzku**: trzy zdania, trzy skille. Cały łańcuch: dane → raport → PDF.

Agent zamienia każdy opis w **plik skilla** (`SKILL.md`) — zwykły tekst, który potem wołam samą nazwą.

Plik, który zapisał agent

skills/pobierz-sprzedaz-gus/SKILL.md

```
---
name: pobierz-sprzedaz-gus
description: Pobiera najnowszy plik GUS
  ze sprzedażą detaliczną (XLSX).
---

# Jak pobrać najnowszy plik

1. Wejdź na stat.gov.pl → obszary
   tematyczne → Ceny. Handel → Handel.
2. Otwórz najnowsze opracowanie
   o sprzedaży detalicznej.
3. Na dole strony znajdź sekcję
   „Dane do wykresów" → plik XLSX.
4. Pobierz go do `dane/sprzedaz.xlsx`.
5. Sprawdź arkusz „miesięczne",
   pomiń wiersze nagłówkowe.
6. Zapisz datę edycji do `dane/zrodlo.txt`.
```

GUS **nie ma stałego linku** do pliku — zmienia się co miesiąc. Dlatego procedurę zapisujemy raz, słowami z Promptu 1.

Skill = **agent + zapisana procedura**. Wołasz go nazwą, agent wykonuje kroki jak nowy pracownik z checklisty.

Kluczowe jest `description` — po nim agent **sam decyduje**, kiedy użyć skilla, bez twojej podpowiedzi.

Krok 2 — pobieramy świeże dane

„Uruchom skill `pobierz-sprzedaz-gus` i przygotuj dane do raportu."

Co robi agent

1. Otwiera stat.gov.pl, wchodzi w Handel
2. Znajduje najnowsze opracowanie (maj 2026)
3. Pobiera XLSX → `dane/sprzedaz.xlsx`
4. Wskazuje arkusz, pomija nagłówki
5. Zapisuje datę edycji źródła

Agent **nie zgaduje** danych z pamięci — schodzi po realny, najświeższy plik prosto ze źródła.

Krok 3 — szablon raportu

```
# Raport sprzedaży detalicznej
## {{miesiąc}} {{rok}} — Firma Demo

## Executive summary
3-4 zdania: główny trend m/m i r/r,
co zaskoczyło, jedna rekomendacja.

## Kluczowe liczby
Tabela: dynamika r/r, m/m, w cenach
stałych i bieżących. Pogrub największą
zmianę.

## Wykresy
1. Dynamika r/r — 12 miesięcy (liniowy)
2. Top 5 kategorii wg zmiany (słupkowy)

## Wnioski
Punktory, ton rzeczowy, bez marketingu.
Zawsze podaj źródło i datę edycji GUS.
```

Szablon to **zwykły plik tekstowy** — placeholdery `{{...}}` i instrukcje dla agenta napisane po polsku.

Sekcje, wykresy, styl wniosków, brand — **ustalone z góry**. Agent ma wypełnić, nie wymyślać układ.

Krok 4 — szablon HTML: opisujesz wygląd słowami

„Zrób **szablon raportu w jednym pliku HTML** z osadzonym CSS. Wygląd:

- **format A4** do druku, marginesy 2 cm, font bezszeryfowy
- **nagłówek** z logo firmy po lewej, tytuł i miesiąc po prawej, pod spodem cienka linia w kolorze brandu (`#0057B8`)
- **Executive summary** w ramce z jasnym tłem
- **Kluczowe liczby** jako tabela zebra, spadki na czerwono, wzrosty na zielono, największa zmiana pogrubiona
- **miejsca na 2 wykresy** (PNG) w siatce 2 kolumn
- **stopka** ze źródłem i datą edycji GUS, numer strony

Placeholdery `{{...}}` tam, gdzie agent wstawi dane."

Markdown wystarcza na 90 % raportów. Gdy potrzebujesz **pełnej kontroli** (układ do druku, kolory, siatka wykresów), mówisz to samo, tylko celujesz w **HTML**.

Nie piszesz HTML-a ręcznie. **Opisujesz wygląd po ludzku**, a agent generuje `szablon.html` z CSS — gotowy do wypełnienia.

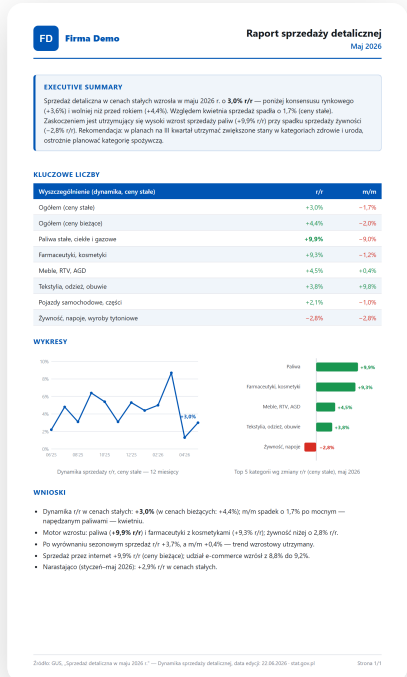
Krok 5 — poprawki

„Wygeneruj raport za maj według `szablon.html` i pokaż podgląd."

„Prawie dobrze. Zmień:

- nagłówek za duży — **zmniejsz logo o połowę**
- tabela: dodaj kolumnę **m/m** obok r/r
- Executive summary daj **nad** tabelą, nie pod
- kolor akcentu na **ciemniejszy niebieski**"

Oglądasz raport i **poprawiasz go rozmową**, jak z grafikiem obok. Każda poprawka wraca do szablonu: robisz go **raz**, miesiącami go nie ruszasz.



FD Firma Demo

Raport sprzedaży detalicznej
Maj 2026

EXECUTIVE SUMMARY

Sprzedaż detaliczna w cenach stałych wzrosła w maju 2026 r. o **3,0% r/r** — poniżej konsensusu rynkowego (+3,6%) i wolniej niż przed rokiem (+4,4%). Względem kwietnia sprzedaż spadła o 1,7% (ceny stałe). Zaskoczeniem jest utrzymujący się wysoki wzrost sprzedaży paliw (+9,9% r/r) przy spadku sprzedaży żywności (−2,8% r/r). Rekomendacja: w planach na III kwartał utrzymać zwiększone stany w kategoriach zdrowie i uroda, ostrożnie planować kategorię spożywczą.

KLUCZOWE LICZBY

Wyszczególnienie (dynamika, ceny stałe)	r/r	m/m
Ogółem (ceny stałe)	+3,0%	−1,7%
Ogółem (ceny bieżące)	+4,4%	−2,0%
Paliwa stałe, ciekłe i gazowe	**+9,9%**	−9,0%
Farmaceutyki, kosmetyki	+9,3%	−1,2%
Meble, RTV, AGD	+4,5%	+0,4%
Tekstylia, odzież, obuwie	+3,8%	+9,8%
Pojazdy samochodowe, części	+2,1%	−1,0%
Żywność, napoje, wyroby tytoniowe	−2,8%	−2,8%

WYKRESY

Dynamika sprzedaży r/r, ceny stałe — 12 miesięcy

Top 5 kategorii wg zmiany r/r (ceny stałe), maj 2026

WNIOSKI

- Dynamika r/r w cenach stałych: **+3,0%** (w cenach bieżących: +4,4%); m/m spadek o 1,7% po mocnym — napędzanym paliwami — kwietniu.
- Motor wzrostu: paliwa (**+9,9% r/r**) i farmaceutyki z kosmetykami (+9,3% r/r); żywność niżej o 2,8% r/r.
- Po wyrównaniu sezonowym sprzedaż r/r +3,7%, a m/m +0,4% — trend wzrostowy utrzymany.
- Sprzedaż przez internet +9,9% r/r (ceny bieżące); udział e-commerce wzrósł z 8,8% do 9,2%.
- Narastająco (styczeń–maj 2026): +2,9% r/r w cenach stałych.

Źródło: GUS, „Sprzedaż detaliczna w maju 2026 r." — Dynamika sprzedaży detalicznej, data edycji: 22.06.2026 · stat.gov.pl

Strona 1/1

Podsumowanie

1. **Powtarzalny raport**: za miesiąc nowa treść, ten sam układ
2. **Skill** pobiera najświeższy plik, zawsze aktualne źródło
3. **Szablon** ustala układ, wykresy i brand
4. Skill **generuj-raport** wypełnia szablon, **raport-do-pdf** składa PDF

🎬 Demo — Pokaz na żywo

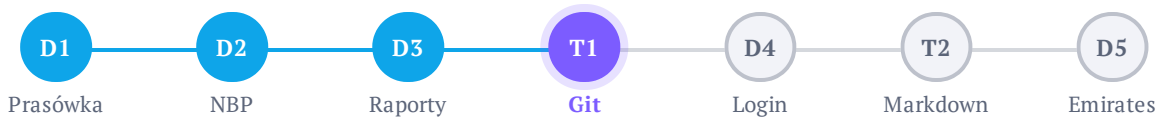
Na co zwrócić uwagę:

- skill opisuję **słowami**, nie kodem — agent zapisuje go jako plik
- szablon HTML **poprawiam rozmową**, jak z grafikiem obok
- szablon powstaje **raz** — miesiącami tylko podmieniam dane

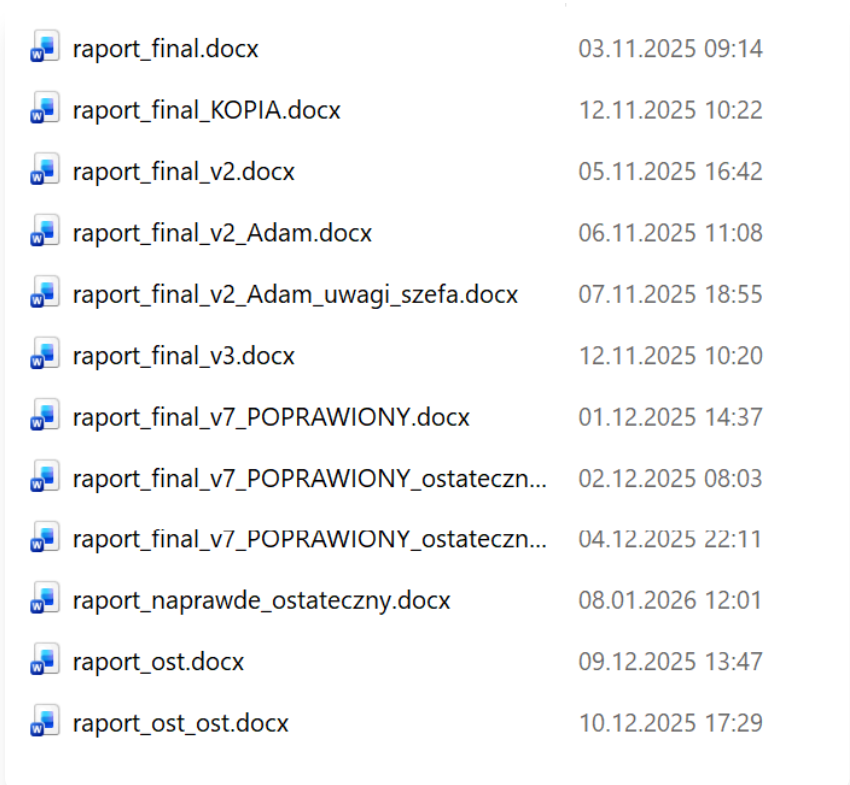
Tool #1

Git

Wersjonowanie plików bez terminala — pełna historia zmian, powrót do każdej wersji



Znasz ten folder?



`raport_final_v7_POPRAWIONY_ostateczny2.docx`

- Który plik jest tak naprawdę ostateczny?
- Kto co zmienił i kiedy?
- Jak wrócić do wersji sprzed tygodnia?

Programiści rozwiązali ten problem **ponad 20 lat temu**. Nazywa się **git**.

Git + TortoiseGit

Co daje git?

- Pełna historia każdej zmiany
- Powrót do dowolnej wersji
- Kto, co i kiedy zmienił
- Porównanie dwóch wersji
- Scalanie zmian wielu osób

- **Git**: terminal
- **TortoiseGit**: GUI

Najlepszy dla plików **tekstowych**, bo umie **porównywać** zmiany. Binarne (`.docx` , `.xlsx`) też utrzyma, tyle że bez porównywania.

AI świetnie zna gita. Każ agentowi: zrobi commit, cofnie zmiany, ogarnie historię. Dobrze też **scala różne zmiany** (merge).

Wiele osób, te same pliki. Każdy pracuje u siebie, git scala zmiany w jedną wspólną wersję.

🎬 Demo — Pokaz na żywo

Na co zwrócić uwagę:

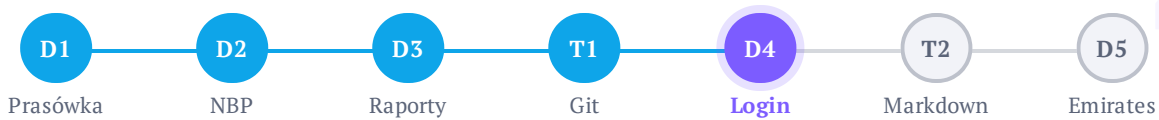
- **agent robi commit** i cofa zmiany — bez wpisywania komend w terminal
- powrót do dowolnej wersji zamiast folderu `raport_final_v7`
- ta sama historia widoczna też w **TortoiseGit** (GUI), bez terminala

Demo #4

zaawansowane

Dane zza logowania

Skrypt · przeglądarka · płatne usługi



Dane są za loginem — trzy drogi

1. Skrypt

Agent **pisze program**, który loguje się i pobiera dane.

2. Przeglądarka

Agent **klika i czyta jak człowiek**, w prawdziwej przeglądarce.

3. Płatne usługi

Gotowa infrastruktura: przeglądarki, proxy, anti-bot na skalę.

Metoda 1 — skrypt (to już znamy)

„Napisz skrypt, który loguje się do serwisu i pobiera moje dane. Jeśli serwis ma **API z kluczem** — użyj API."

Dokładnie ten sam mechanizm co w:

- **demo #1** — agent dobrał **API (MCP)**,
- **demo #2** — agent napisał **skrypt** pod API NBP.

Najtańsza droga. Skrypt raz napisany biegnie potem w sekundy, bez agenta i bez tokenów.

Działa świetnie, **gdy jest API albo prosta strona**. Formularz, JavaScript, 2FA → problemy.

Metoda 2 — Playwright: agent w przeglądarce

Playwright to narzędzie Microsoftu do **sterowania przeglądarką**. Powstało do testów aplikacji WWW.

Podpięte do agenta jako **MCP**.

Nie ma API? Nie szkodzi. Agent **widzi stronę tak jak my** i klika tak jak my. Użyteczne też do automatyzacji zadań w przeglądarce.

Co agent potrafi w przeglądarce

- **otworzyć** dowolny adres,
- **odczytać** strukturę strony (przyciski, pola, teksty),
- **wpisać** tekst, **kliknąć**, przewinąć, poczekać,
- **wypełnić** formularz, pobrać plik, przejść dalej.

Metoda 2 — jak to uruchomić

.mcp.json

```
{
  "mcpServers": {
    "playwright": {
      "command": "npx",
      "args": ["@playwright/mcp@latest"]
    }
  }
}
```

„Skonfiguruj **Playwright jako MCP**. Potem otwórz **the-internet.herokuapp.com/login** i pokaż, co widzisz."

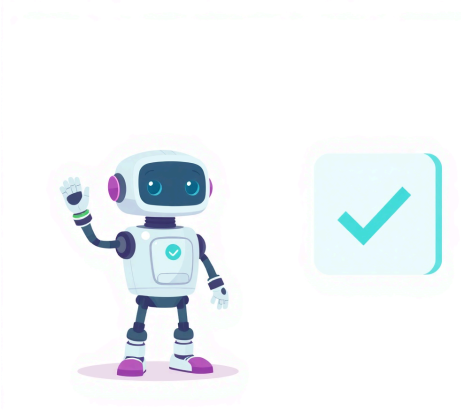
Nic nie instalujesz ręcznie. `npx` sam pobiera Playwright przy pierwszym uruchomieniu.

Domyślnie przeglądarka **otwiera widoczne okno** na Twoim ekranie — widzisz na żywo, co agent klika (tryb ukryty: `--headless`).

Wolniej i drożej niż skrypt — **każdy klik to decyzja agenta**, czyli tokeny.

Metoda 2 — logujesz się Ty, nie agent

1. Agent otwiera stronę logowania.
2. **Ty wpisujesz login, hasło, kod z SMS** — ręcznie, w oknie.
3. Mówisz „**jestem zalogowany, przejmij**".
4. Agent klika i pobiera dane **w Twojej sesji**.



2FA, captcha, „czy to Ty?" — te momenty robisz sam. Agent czeka i przejmuje po zalogowaniu.

Sesja zostaje w **profilu przeglądarki** na Twoim dysku. Następnym razem agent od razu przechodzi do roboty.

Metoda 3 — płatne usługi i MCP

Gdy potrzebujesz **skali** albo strona ma mocny **anti-bot** — gotowa infrastruktura zamiast własnego narzędzia.

Usługa	Co daje	Kiedy
Firecrawl	strona → markdown pod LLM, tryb stealth	pipeline AI, mały budżet (znamy z demo #1)
Browserbase	zdalne przeglądarki dla agentów	metoda 2 na skalę, wiele sesji naraz
Apify	10 000+ gotowych scraperów bez kodu	gotowiec do typowego serwisu
Bright Data	proxy 400M+ IP, najmocniejszy anti-bot	najtrudniejsze strony, duża skala

Rama prawna — trzy zasady, zanim zautomatyzujesz

✅ **Własne konto, własne dane.** Automatyzujesz dostęp, który i tak masz — ten sam eksport, który klikasz ręcznie. To automatyzacja własnej pracy.

⚠️ **Regulamin serwisu (ToS).** Część serwisów zabrania automatyzacji — sprawdź, zanim uruchomisz. Blokada (`robots.txt` , `403` , captcha) to sygnał „nie chcemy botów", a nie wyzwanie sportowe.

❌ **Masowy scraping cudzych danych = nie.** Dane osobowe, treści innych użytkowników — to nie jest automatyzacja pracy, to problem prawny.

🎬 Demo — Pokaz na żywo

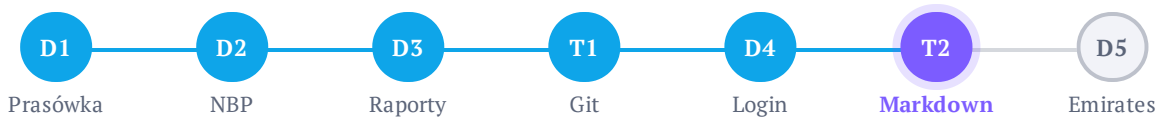
Na co zwrócić uwagę:

- loguję się **ja, nie agent** — hasło nigdy nie trafia do modelu
- agent naprawdę **klika w prawdziwej przeglądarce** (Playwright)
- do płatnej usługi sięgam **świadomie**, gdy własny skrypt nie wystarcza

Tool #2

Markdown

Czysty tekst, który wygląda jak dokument — i natywny język AI



Po co mi markdown?

Problem: Word ukrywa formatowanie w binarnym pliku. Nie zobaczysz w gicie, co się zmieniło. AI się w tym gubi.

Markdown: czysty tekst ze znacznikami. Czytelny bez żadnego programu — i w notatniku, i jako ładny dokument.

Idealny do **gita** (diff pokazuje zmienione zdanie) i **natywny język AI** — agenci piszą go najlepiej.

notatka.md

```
# Raport sprzedaży — czerwiec

## Podsumowanie
Sprzedaż wzrosła o **12%** względem maja.

| Region      | Sprzedaż | Zmiana |
|-------------|---------:|-------:|
| Mazowieckie |  120 000 |  +8%   |
| Śląskie     |   95 000 |  +15%  |

- Najlepszy miesiąc w tym roku
```

markdown → PDF

```
## Podsumowanie
Sprzedaż wzrosła o **12%** względem maja.
Najlepszy miesiąc w tym roku.

## Wyniki wg regionów

| Region      | Sprzedaż | Zmiana |
|-------------|---------:|-------:|
| Mazowieckie |  120 000 |  +8%   |
| Śląskie     |   95 000 |  +15%  |
| Pomorskie   |   72 000 |  +5%   |

## Kluczowe obserwacje

- Śląskie z **najwyższą dynamiką** wzrostu
- Nowi klienci odpowiadają za 1/3 przychodu
- Kampania e-mail zwróciła się w 2 tygodnie

## Plan na lipiec

1. Powtórzyć kampanię w Pomorskiem
2. Zwiększyć stany magazynowe
```

Raport sprzedaży — czerwiec 2026

Podsumowanie

Sprzedaż wzrosła o **12%** względem maja. Najlepszy miesiąc w tym roku.

Wyniki wg regionów

Region	Sprzedaż	Zmiana
Mazowieckie	120 000	+8%
Śląskie	95 000	+15%
Pomorskie	72 000	+5%

Kluczowe obserwacje

- Śląskie z **najwyższą dynamiką** wzrostu
- Nowi klienci odpowiadają za 1/3 przychodu
- Kampania e-mail zwróciła się w 2 tygodnie

Plan na lipiec

1. Powtórzyć kampanię w Pomorskiem
2. Zwiększyć stany magazynowe
3. Raport śródmiesięczny w połowie lipca

Uwaga: dane wstępne, korekta do 5. dnia miesiąca.

Czym to otworzyć?

Do edycji:

- **VS Code** — podgląd na żywo
- **Typora** — WYSIWYG jak Word
- **Obsidian** — baza notatek

Do PDF-a:

- **pandoc** — konwerter-kombajn: md → PDF / Word / HTML
- eksport jednym klikiem z Typory
- VS Code + rozszerzenie Markdown PDF

Markdown + AI w praktyce

Agent generuje **markdown**, pandoc robi z niego **firmowy PDF** (szablon + logo).

Notatki ze spotkań, dokumentacja procesów, raporty — wszystko pisane z AI.

Cały obieg tekstem: piszesz z AI → wersjonujesz w gicie → eksportujesz do PDF. Jeden format, zero klikania w formatowanie.

🎬 Demo — Pokaz na żywo

Na co zwrócić uwagę:

- ten sam **markdown** → trzy różne PDF-y (inny szablon, to samo źródło)
- agent pisze tekst, **pandoc** składa firmowy PDF z logo
- zero klikania w formatowanie — **cały obieg tekstem**

Demo #5

zaawansowane

Walka z gigantem

AI jako osobisty paralegal · sprawa o odwołany lot

Sprawa: Emirates odwołał lot

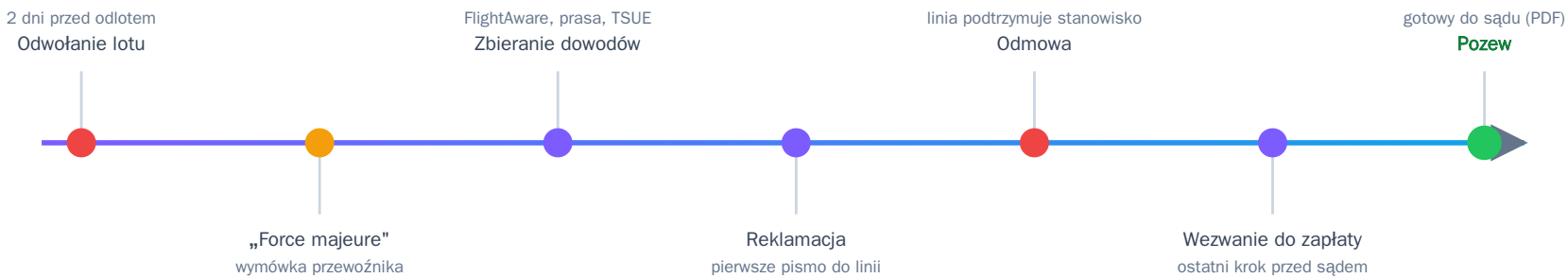
Fakty sprawy

Lot	Warszawa → Dubaj, odwołany 2 dni przed
Powód	„operational reasons" → „force majeure"
Roszczenie	600 EUR (rozp. UE 261/2004)
Przeciwnik	korporacja i jej dział prawny

Uniwersalne demo: **każdy** miał odwołany lot albo spór z firmą „nie do ruszenia".

Z AI zbudowałem **całą sprawę** — od chaosu maili do pozwu gotowego do sądu.

Cała sprawa na jednej osi



Od odwołanego lotu do pozwu gotowego do sądu — **agent trzyma cały porządek** w jednym katalogu.

Jak to działa — jeden katalog + agent

Repozytorium sprawy

```
emirates-lot/
├── sprawa.md       # kto, o co, ile, status
├── timeline.md     # chronologia i terminy
├── dokumenty/      # PDF-y → transkrypcje
├── pisma/          # reklamacja → wezwanie → pozew
└── skrypty/        # odsetki, przedawnienie
```

- Dowody **przepisane z PDF na tekst** — agent je czyta i cytuje
- Nowa sesja? Agent czyta `sprawa.md` i **wie wszystko**
- Terminy i odsetki liczy **skrypt, nie model**

Ta sama mechanika co prasówka: **pliki tekstowe + agent**. Zmienił się temat.

AI cytuje, nie zmyśla

Emirates: „force majeure" — zamieszki w Antananarywie. Agent poszedł po fakty do internetu:

Kontrdowód	Co pokazuje	Źródło
Inne linie latały	AirLink doleciał 03 i 05.10	FlightAware
Sam Emirates latał	EK0707 lądował 09–11.10	FlightAware
Godzina policyjna	od 19:00, przylot planowo 16:50	Reuters, BBC

Każdy dowód to **zrzut PDF + transkrypcja z datą i źródłem** — te same pliki pójdą do sądu jako załączniki. Resztę (orzecznictwo TSUE, kalkulator terminów, gotowy pozew) pokażę **w demo**.

AI jako osobisty paralegal — warsztat, nie wyrok

1. **Jeden katalog**: od chaosu maili do pozwu gotowego do druku
2. Dowody i orzecznictwo **przypięte do argumentów**, AI cytuje
3. Terminy i odsetki **liczy skrypt**, nie model

Sprawa w toku — pokazuję warsztat, nie sprzedaję wygranej. To materiały pomocnicze, nie porada prawna.

Ile to kosztuje — plany Claude

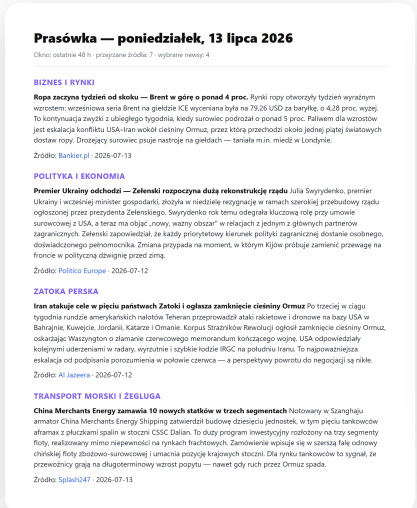
Pierwsze pytanie biznesu. **Wejście za $20** — i Claude Code jest w cenie od Pro.

Plan	Cena / mies.	Dla kogo
Free	0 $	chat, web search, pamięć, pliki — spróbować bez ryzyka
Pro	**20 $** ($17 rocznie)	sweet spot dla nie-programisty — **Claude Code w cenie**
Max	100 $ / 200 $	5× / 20× limitów Pro — cięższa praca z agentem
Team	od 25 $ / os.	zespół (min. 5 miejsc), wspólne rozliczenie

Stan cen: lipiec 2026. Enterprise (SSO, audit logs, kontekst 500K) — wycena indywidualna.

Co dziś powstało

Demo #1 · prasówka



Prasówka — poniedziałek, 13 lipca 2026

Okno: ostatnie 48 h · przejrzane źródła: 7 · wybrane newsy: 4

BIZNES I RYNKI

Ropa zaczyna tydzień od skoku — Brent w górę o ponad 4 proc. Rynki ropy otworzyły tydzień wyraźnym wzrostem: wrześniowa seria Brent na giełdzie ICE wyceniana była na 79,26 USD za baryłkę, o 4,28 proc. wyżej. To kontynuacja zwyżki z ubiegłego tygodnia, kiedy surowiec podrożał o ponad 5 proc. Paliwem dla wzrostów jest eskalacja konfliktu USA–Iran wokół cieśniny Ormuz, przez którą przechodzi około jednej piątej światowych dostaw ropy. Drożejący surowiec psuje nastroje na giełdach — taniała m.in. miedź w Londynie.

Źródło: Bankier.pl · 2026-07-13

POLITYKA I EKONOMIA

Premier Ukrainy odchodzi — Zełenski rozpoczyna dużą rekonstrukcję rządu Julia Swyrydenko, premier Ukrainy i wcześniej minister gospodarki, złożyła w niedzielę rezygnację w ramach szerokiej przebudowy rządu ogłoszonej przez prezydenta Zełenskiego. Swyrydenko rok temu odegrała kluczową rolę przy umowie surowcowej z USA, a teraz ma objąć „nowy, ważny obszar" w relacjach z jednym z głównych partnerów zagranicznych. Zełenski zapowiedział, że każdy priorytetowy kierunek polityki zagranicznej dostanie osobnego, doświadczonego pełnomocnika. Zmiana przyjdała na moment, w którym Kijów próbuje zamienić przewagę na froncie w politycznąą dźwignię przed zimą.

Źródło: Politico Europe · 2026-07-12

ZATOKA PERSKA

Iran atakuje cele w pięciu państwach Zatoki i ogłasza zamknięcie cieśniny Ormuz Po trzeciej w ciągu tygodnia rundzie amerykańskich nalotów Teheran przeprowadził ataki rakietowe i dronowe na bazy USA w Bahrajnie, Kuwejcie, Jordanii, Katarze i Omanie. Korpus Strażników Rewolucji ogłosił zamknięcie cieśniny Ormuz, oskarżając Waszyngton o złamanie czerwcowego memorandum kończącego wojnę. USA odpowiedziały kolejnymi uderzeniami w radary, wyrzutnie i szybkie łodzie IRGC na południu Iranu. To najpoważniejsza eskalacja od podpisania porozumienia w połowie czerwca — a perspektywy powrotu do negocjacji są nikłe.

Źródło: Al Jazeera · 2026-07-12

TRANSPORT MORSKI I ŻEGLUGA

China Merchants Energy zamawia 10 nowych statków w trzech segmentach Notowany w Szanghaju armator China Merchants Energy Shipping zatwierdził budowę dziesięciu jednostek, w tym pięciu tankowców aframax z płuczkami spalin w stoczni CSSC Dalian. To duzy program inwestycyjny rozłożony na trzy segmenty floty, realizowany mimo niepewności na rynkach frachtowych. Zamówienie wpisuje się w szerszą falę odnowy chińskiej floty zbożowo-surowcowej i umacnia pozycję krajowych stoczni. Dla rynku tankowców to sygnał, że przewoźnicy grają na długoterminowy wzrost popytu — nawet gdy ruch przez Ormuz spada.

Źródło: Splash247 · 2026-07-13

Demo #2 · raport NBP

Kurs EUR/PLN — I półrocze 2026

Okres: 2026-01-02 → 2026-07-10 · Źródło: NBP, tabela A (api.nbp.pl) · Notowań: 132 · Wygenerowano: 2026-07-13

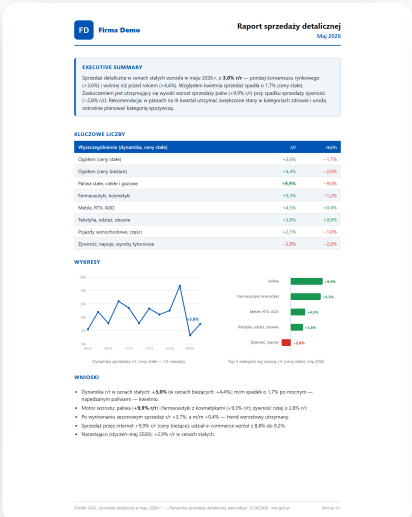
Agregacja miesięczna

Miesiąc	Średnia	Min	Max	Liczba notowań
styczeń 2026	4.2126	4.2009	4.2279	20
luty 2026	4.2174	4.2080	4.2241	20
marzec 2026	4.2718	4.2304	4.2894	22
kwiecień 2026	4.2529	4.2320	4.2874	21
maj 2026	4.2417	4.2284	4.2551	20
czerwiec 2026	4.2583	4.2334	4.2963	21
lipiec 2026	4.3020	4.2857	4.3465	8

Statystyki okresu

Średnia	Mediana	Min	Max	Odch. standardowe	Zmiana w okresie
4.2467	4.2435	4.2009	4.3465	0.0285	+3.11%

Demo #3 · raport GUS



FD Firma Demo

Raport sprzedaży detalicznej
Maj 2026

EXECUTIVE SUMMARY

Sprzedaż detaliczna w cenach stałych wzrosła w maju 2026 r. o **3,0% r/r** — poniżej konsensusu rynkowego (+3,6%) i wobec spadku przed rokiem (+4,4%). Wzgledem kwietnia sprzedaż spadła o 1,7% (ceny stałe). Zaskoczeniem jest utrzymujący się wysoki wzrost sprzedaży paliw (+9,9% r/r) przy spadku sprzedaży żywności (−2,8% r/r). Rekomendacja w placach na III kwartał utrzymuje zwiększone stany w kategoriach zdrowie i uroda, ostrożnie planować kategorię spożywczą.

KLUCZOWE LICZBY

Wyszczególnienie (dynamika, ceny stałe)	r/r	m/m
Ogółem (ceny stałe)	+3,0%	−1,7%
Ogółem (ceny bieżące)	+4,4%	−2,0%
Paliwa stałe, ciekłe i gazowe	+9,9%	−9,0%
Farmaceutyki, kosmetyki	+9,3%	−1,2%
Meble, RTV, AGD	+4,5%	+0,4%
Tekstylia, odzież, obuwie	+3,8%	+9,0%
Pojazdy samochodowe, części	+2,1%	−1,0%
Żywność, napoje, wyroby tytoniowe	−2,8%	−2,8%

WYKRESY

Dynamika sprzedaży r/r, ceny stałe — 12 miesięcy

Top 5 kategorii wg zmiany r/r (ceny stałe), maj 2026

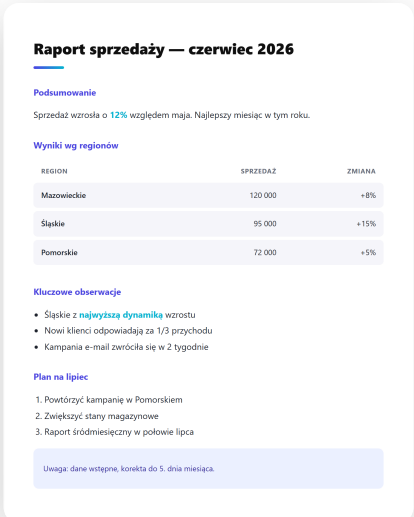
WNIOSKI

- Dynamika r/r w cenach stałych: **+3,0%** (w cenach bieżących: +4,4%); m/m spadek o 1,7% po mocnym — najpodatniejszy paliwami — świątecznu.
- Motor wzrostu: paliwa (**+9,9% r/r**) i farmaceutyki z kosmetykami (+9,3% r/r); żywność niżej o 2,8% r/r.
- Po wyrównaniu sezonowym sprzedaż r/r +3,7%, a m/m +0,4% — trend wzrostowy utrzymuje.
- Sprzedaż przez internet +9,9% r/r (ceny bieżące); udział e-commerce wzrósł z 8,8% do 9,2%.
- Narastająco (styczeń–maj 2026): +2,0% r/r w cenach stałych.

Źródło: GUS, „Sprzedaż detaliczna w maju 2026 r." — Dynamika sprzedaży detalicznej, data edycji: 22.06.2026 · stat.gov.pl

Strona 1/1

Tool #2 · markdown → PDF



Raport sprzedaży — czerwiec 2026

Podsumowanie

Sprzedaż wzrosła o **12%** względem maja. Najlepszy miesiąc w tym roku.

Wyniki wg regionów

REGION	SPRZEDAŻ	ZMIANA
Mazowieckie	120 000	+8%
Śląskie	95 000	+15%
Pomorskie	72 000	+5%

Kluczowe obserwacje

- Śląskie z **najwyższą dynamiką** wzrostu
- Nowi klienci odpowiadają za 1/3 przychodu
- Kampania e-mail zwróciła się w 2 tygodnie

Plan na lipiec

1. Powtórzyć kampanię w Pomorskiem
2. Zwiększyć stany magazynowe
3. Raport śródmiesięczny w połowie lipca

Uwaga: dane wstępne, korekta do 5. dnia miesiąca.

Cztery realne pliki z dzisiejszych dem. Każdy klikalny — to jest ten **realny zysk**.

Co zrobić w poniedziałek

Nie musisz zmieniać całego sposobu pracy. **Zacznij od jednej rzeczy.**

1. Claude.ai → Projects

Wrzuć swoje pliki i zasady (`CLAUDE.md`) do projektu. Zadaj pierwsze pytanie o **własne** dane — jak w prasówce.

2. Claude Desktop

Podłącz **MCP** do dysku i internetu. Każ napisać skrypt, który liczy, jak w demo NBP.

3. Pierwszy skill / agent

Zapisz powtarzalny prompt jako **skill**. Raz ustawione — działa co tydzień.

AI jest proste — 4h w weekend = realny zysk. Nie musisz umieć programować — agent jest tłumaczem.

Gotowe prompty i pliki z dem: **prezentacje.robertolechowski.com**

Podsumowanie

„AI nie zabierze ci pracy. Ludzie korzystający z AI zabiorą ją tym, którzy z AI nie korzystają.”

— parafraza **Richarda Baldwina** (WEF Growth Summit, 2023): *„AI won’t take your job. It’s somebody using AI that will take your job.”*

Więcej o AI na blogu

To, co dziś zobaczyłeś, rozpisane krok po kroku. **blog.robertolechowski.com**

Agenci AI

Czym są, jak działają, ekosystem, pamięć, orkiestracja, praktyczne projekty.

MCP dla Claude

Model Context Protocol. Architektura, limity tokenów, gotowe serwery, własny MCP w Pythonie.

Skille w Claude Code

Jak delegować zadania. Zapisany prompt, który działa w kółko.

Agenci AI w praktyce

Wprowadzenie dla nie-programistów. Fundamenty bez pisania kodu.

blog.robertolechowski.com — wszystko z tej prezentacji, rozpisane

Kontakt

Robert Olechowski

🌐 robertolechowski.com

📝 blog.robertolechowski.com

🐙 github.com/RobertOlechowski

📧 robertolechowski@gmail.com

Materiały do pobrania

prezentacje.robertolechowski.com

Materiały GitHub

Usługi dla firm

- Szkolenia zespołów z AI
- Doradztwo we wdrażaniu AI
- Audyt i automatyzacja procesów
- Konsulting IT

Dziękuję

Pytania?

prezentacje.robertolechowski.com