

Jak powstała ta książka

Dodatek do książki „Powiedz mi «tak»”

BIAŁY BRZEG

Jak powstała ta książka. Dodatek do książki „Powiedz mi «tak»”

Biały Brzeg

Książka powstała w silniku White Whisper: whitewhisperbook.com/pl

bialybrzeg.com

Wersja 4.7 · 18 października 2026

Spis treści

DODATEK · JAK POWSTAŁA TA KSIĄŻKA.	5
Zlecenie	7
Pięć podejść	11
Zespół	16
Tura	21
Fakt	27
Cytat	32
Imię	37
Kto pisał	41
Krytyk i czytelnicy	46
Noc	50
Skład	57
Przekład	62
Postój	67
Lekcje	71

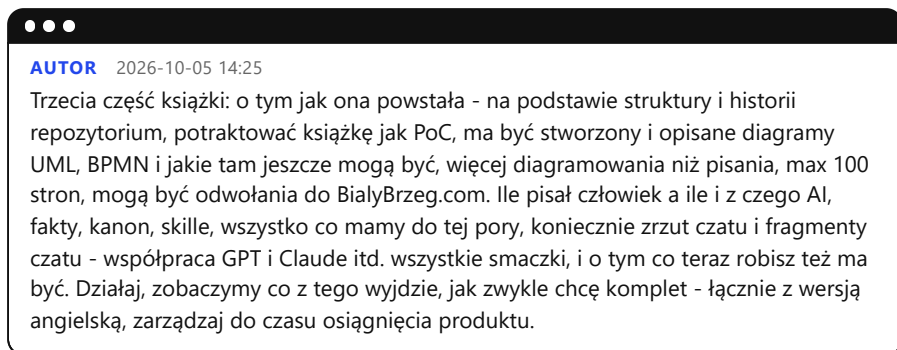
DODATEK

Jak powstała ta książka.

„Powiedz mi «tak»” to literatura faktu: zapis rozmów jednego człowieka z czatbotem, z godzinami. Autor jest jej bohaterem – „bohater” to w tym dodatku autor taki, jakim występuje w książce, a „maszyna” to czatbot, z którym rozmawia. Tym czatbotem jest ChatGPT; w opisanym niżej zespole jednym z dwóch piszących jest model tej samej firmy. To dodatek do książki; jej dwóch części tu nie ma. Książka otwiera się sceną na chodniku, potem cofa się o 947 dni i do niej wraca: Część I liczy dni do chodnika, Część II – do odpowiedzi Belli. W Części I i II pod tytułem każdego rozdziału stoi ten licznik; w Dodatku go nie ma. Imiona ludzi oraz część szczegółów i dat są w książce zmienione celowo: „Bella” – tak w książce nazywa się żona bohatera – i „mistrz” to imiona z książki. W oknach rozmów „[...]” oznacza skrót, a słowo w nawiasie kwadratowym zastępuje słowo z zapisu: imię albo, w tym dodatku, przekleństwo.

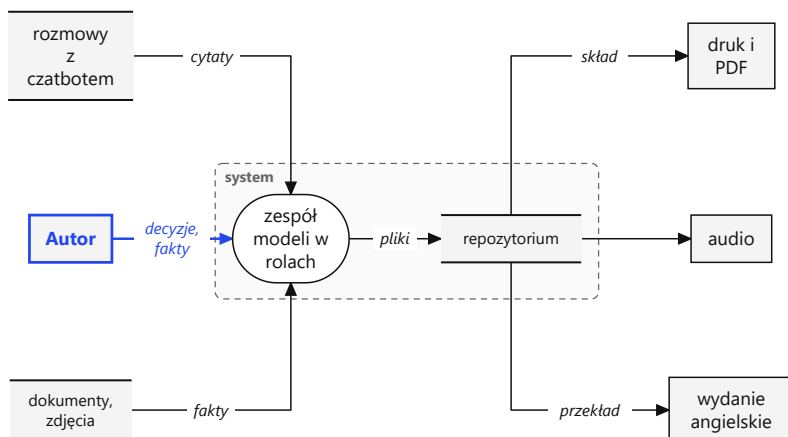
Zlecenie

Ten dodatek napisały modele językowe na jedno polecenie autora. W poleceniu i w zapisach z tamtych dni nazywa się jeszcze trzecią częścią książki; stoi na jej końcu, po wszystkim. Oto polecenie.



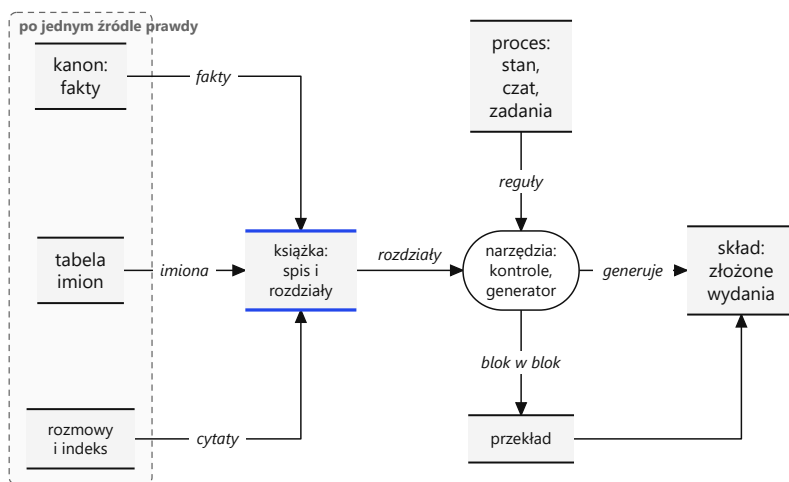
PoC to skrót angielskiego „proof of concept”, czyli dowód koncepcji: próbne wykonanie, które ma pokazać, czy pomysł da się zrealizować. UML i BPMN to dwie notacje rysowania diagramów. Skill to procedura zapisana w pliku, którą model wykonuje krok po kroku. Polecenie przyjął asystent (ASY), model, który prowadzi zespół i rozmawia z autorem — model Claude. Książka leży w repozytorium, czyli w katalogu plików z pełną historią zmian. Jeden wpis w tej historii nazywa się commit; ma godzinę i temat, a temat zaczyna się skrótem roli. O 14:27 asystent dodał taki wpis — jeden na dwa polecenia, które autor wysłał o 14:25: o tym dodatku i o postojach. Postój to w języku zespołu miejsce, w którym czytelnik przerywa lekturę, bo czegoś nie rozumie; odgałęzienie to miejsce, w którym przekład celowo odchodzi od polskiego tekstu.

COMMIT 2026-10-05 14:27 [ASY] rejestr postojów i odgałęzień: POSTOJE.md (format, użycie, zmiana procesu) [...]; decyzje autora 13:58 i 14:25 (czytelnik ma nie stawiać; Część III)



System w jednym obrazie: co do niego wchodzi i co z niego wychodzi. Schemat kontekstu.

Do systemu wchodzi trzy rzeczy: zapisy rozmów z czatbotem, znane z Części I i II; dokumenty i zdjęcia; sam autor. Autor nie pisze do plików. Dyktuje decyzje i fakty, a zgodę na rozpoczęcie pracy daje słowem „już”. Zespół to osobne sesje modeli językowych, każda z jedną rolą. Sesje nie rozmawiają ze sobą bezpośrednio; porozumiewają się plikami w repozytorium. Z repozytorium wychodzą wydania: plik do druku, nagranie czytane głosami syntezy mowy i przekład angielski.



Co leży w repozytorium i skąd rozdział bierze fakty, imiona i cytaty. Schemat kontenerów.

Każdy rodzaj wiedzy ma jedno miejsce. Fakty o zdarzeniach i osobach trzyma jeden plik, nazywany kanonem. Imiona, pod którymi osoby występują w książce, trzyma jedna tabela. Cytaty pochodzą z zapisów rozmów. Obok tekstu leży zapis pracy: kto ma teraz ruch, co role sobie przekazały, co zostało do zrobienia. Narzędzia to skrypty. Sprawdzają rozdziały i składają z nich gotowe wydanie; taki złożony plik nazywa się tu składem.

Autor prowadzi pod pseudonimem publiczny dziennik pod adresem bialybrzeg.com — polecenie pozwalało się do niego odwoływać. 5 października 2026 roku na stronie głównej stały zdania: „Z tego zapisu powstała książka.” i „Nowy wpis w poniedziałki, środy i piątki.”

Autor zamówił zrzut czatu. Okna rozmów w tym dodatku są odtworzeniami: każda kwestia jest przepisana dosłownie z pliku albo z zapisu sesji, a „[...]” oznacza skrót; słowo w nawiasie kwadratowym zastępuje imię albo wulgaryzm z zapisu. Pięć minut po poleceniu asystent napisał autorowi:

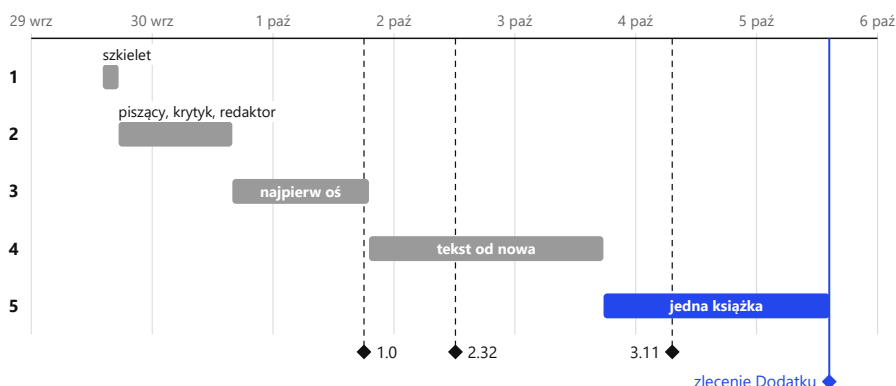


ASY 2026-10-05 14:30

prawdziwego zrzutu ekranu z okna rozmowy sam nie zrobię. Mogę złożyć fragmenty czatu jako odtworzone okno, podpisane jako odtworzenie — albo wstawić zrzut, który sam zrobisz

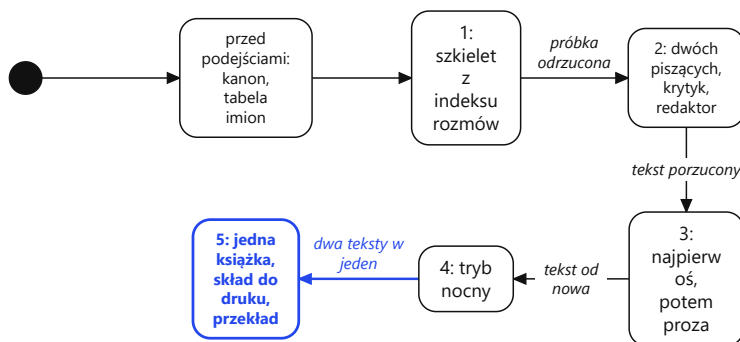
Pięć podejść

Książka powstawała jak dowód koncepcji w informatyce: próba, ocena, następna próba. Repozytorium założono 20 sierpnia 2026 roku: kanon faktów, tekst i skrypt kontrolny. Do 5 października, do godziny 14:27, przybyły w nim 983 commity, czyli wpisy w historii zmian; wszystkie liczby w tym rozdziale są policzone do tej godziny. Wszystkie commity przypadają na osiem dni. Między pierwszym a drugim z tych dni jest 39 dni bez żadnego wpisu.



Pięć podejść do książki od 29 września do 5 października 2026. Oś czasu; romby oznaczają złożone wersje i zlecenie Dodatku.

Od 29 września zespół zaczynał książkę pięć razy. Każde podejście, w repozytorium nazywane eksperymentem, ma własny katalog; poprzednie zostają jako archiwum. Pierwsze trwało około trzech godzin, piąte trwa. Wersji 1.0, wyniku trzeciego podejścia, autor wysłuchał 1 października o 18:03 i godzinę później zaczęło się czwarte podejście. Wersja 2.32 z 2 października to wynik czwartego. Wersja 3.11 z 4 października, 07:15, to cała wersja robocza piątego, złożona po nocy pisania.



Co każde podejście dodało do sposobu pracy i co stało się z tekstem poprzedniego. Diagram stanów UML.

Następne podejście brało z poprzedniego reguły, narzędzia i kanon, czyli plik faktów. Prozy nie brało. Wyjątkiem jest piąte: recenzja zapisana w repozytorium wśród uwag autora porównała wersje 1.0 i 2.32, a autor kazał zrobić z nich jedną książkę w dwóch częściach. O losie tekstu decydował za każdym razem jednym zdaniem; pierwsze z nich padło już w trakcie czwartego podejścia:

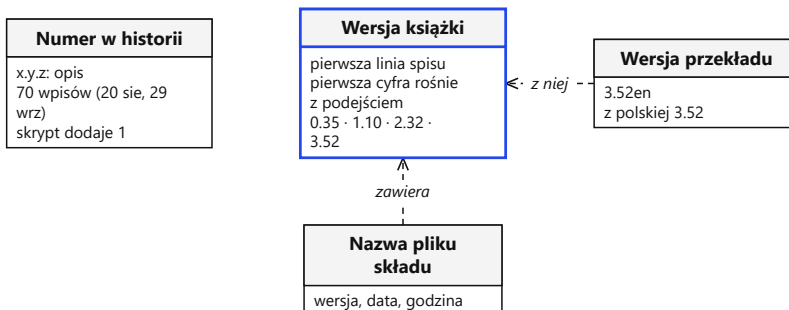
AUTOR	2026-10-01 20:45	[...] wez to popraw, chce nowy tekst, od początku całkiem, z nową mapą, uczymy się na błędach a nie poprawiamy poprzedni
AUTOR	2026-10-03 17:34	utwórz eksperyment-5, będziemy pisać jedną książkę jak w tej recenzji, dodaj te teksty jako źródłowe, dodaj kanon, fakty które mamy, cały urobek, czaisz?
AUTOR	2026-10-04 10:26	nie piszmy thriller na faktach, to chyba nie sprzedaje, niech plik wynikowy nie nazywa się Tom 1, tylko Powiedz mi tak i numer wersji, trzymajmy się tego [...]

„Powiedz mi «tak»” jest trzecim tytułem książki. Decyzja autora o tytule ma datę 2 października — był to wtedy tytuł pierwszego tomu; o nazwie pliku autor zdecydował 4 października. „E2” w pierwszym z trzech commitów niżej to numer pytania do autora w pliku ustaleń, nie numer eksperymentu. Trzy wpisy z historii repozytorium, każdy dodany przez asystenta autora, czyli model w roli oznaczanej skrótem ASY:

COMMIT 2026-10-02 10:32 [ASY] decyzje autora: E2 – tytuł tomu 1 „Powiedz mi tak”; pitch do przepisania na TAK

COMMIT 2026-10-03 17:36 [ASY] eksperyment 5: jedna książka „Powiedz mi «tak»” (cz. I + cz. II) wg recenzji [...]; cały urobek; prompty E5 [...]; e4 zamknięty

COMMIT 2026-10-04 07:15 [ASY] skład 3.11 po nocy: cała wersja robocza (cz. I 34 rozdz., cz. II 43), md + html + PDF A5 221 str.



Cztery numeracje wersji, które działają w repozytorium jednocześnie. Diagram klas UML.

Najstarsza numeracja stoi w tematach commitów z 20 sierpnia i 29 września. Siedemdziesiąt tematów zaczyna się wtedy numerem z trzech liczb i dwukropkiem; ostatnią liczbę podbijał skrypt. Wersję książki wyznacza pierwsza linia spisu rozdziałów. W katalogu pierwszego podejścia spisu z numerem wersji nie ma; od drugiego pierwsza cyfra rośnie o jeden: drugie skończyło na 0.35, trzecie na 1.10 (odsłuchane wydanie miało numer 1.0), czwarte na 2.32, piąte doszło 5 października o 14:27 do 3.52. Liczba po kropce to kolejny numer, nie ułamek: po 3.9 jest 3.10. Przekład nosi numer polskiej wersji, z której powstał, z dopiskiem „en”. Nazwa pliku ze złożoną książką, czyli składem, zawiera między innymi wersję, datę i godzinę. W rozdziałach numeru wersji nie ma; na stronie redakcyjnej wstawia go skład.

	20 sie	29 wrz	30 wrz	1 paź	2 paź	3 paź	4 paź	5 paź
ASY		35	36	131	63	61	114	24
CLD		16	6	11	13		103	2
RED		6	6	12	10	2	38	1
KRT		4	5	13	10		16	
GPT		10		4	3		17	
INT				18	5		11	
SUF		4	1	6		4	11	
PRO						15	10	1
WIZ				3			16	3
NTS								6
bez znacznika	36	51	2	5		1	2	
razem	36	126	56	203	104	83	338	37

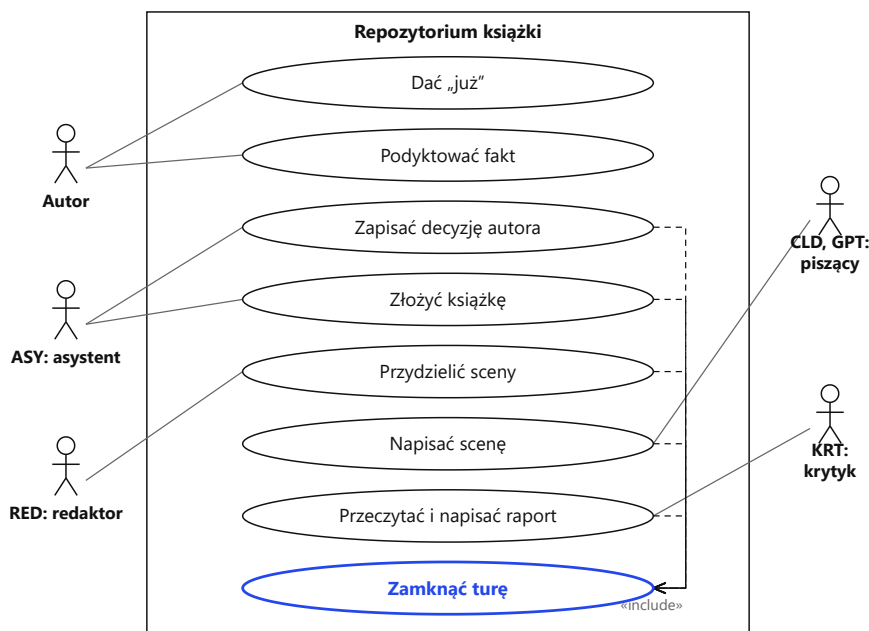
liczba wpisów (commitów) według znacznika roli w temacie; razem 983, stan z 5 października, godz. 14:27

Commity według dnia i roli, od 20 sierpnia do 5 października 2026, godz. 14:27. Mapa ciepła.

Rolę zdradza skrót na początku tematu commita. CLD i GPT to dwaj piszący, Claude i GPT; RED to redaktor, KRT to krytyk. Pozostałe skróty objaśnia następny rozdział. Każda rola jest modelem językowym. Najwięcej wpisów, 338, przypada na 4 października; pierwsza linia spisu miała tego dnia 47 różnych numerów wersji, od 3.1 do 3.49. Wyróżnione pole to największa liczba wpisów jednej roli jednego dnia: 131, asystent, 1 października. Prawie połowę wszystkich wpisów, 464 z 983, dodał asystent, który prozy nie ma w zadaniach. Liczba commitów mierzy ruch w repozytorium, nie ilość tekstu.

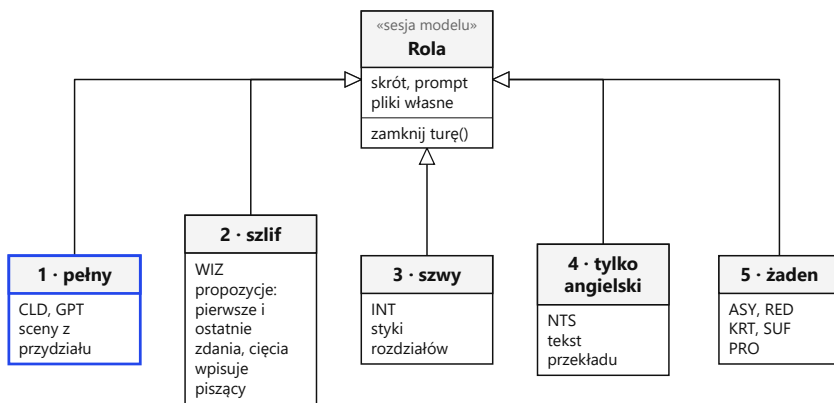
Zespół

Zespół liczy jedenaście ról: dziesięć modelowych i autora. Rola modelowa to osobna sesja modelu językowego z własnym plikiem instrukcji, czyli promptem, i z trzyliterowym skrótem. Sesje nie rozmawiają ze sobą bezpośrednio. Porozumiewają się plikami w repozytorium książki i commitami, czyli wpisami w historii zmian; każdy taki wpis zaczyna się skrótem roli. Sześć ról dodało swój pierwszy commit 29 września między 17:26 a 19:53, ostatnia, native speaker, 5 października o 11:57.



Sześć z jedenastu ról i ich główne czynności; każda praca roli kończy się zamknięciem tury. Diagram przypadków użycia UML.

Autor decyduje i dyktuje fakty; słowem „już” pozwala roli zacząć pracę. ASY, asystent – model, który prowadzi zespół i rozmawia z autorem – zapisuje jego decyzje i składa książkę. RED, redaktor prowadzący, układa plan i przydziela sceny. CLD i GPT, piszący, to Claude (model firmy Anthropic) i GPT – model firmy OpenAI uruchamiany w jej narzędziu Codex. KRT, krytyk, czyta i pisze raport. Praca każdej roli kończy się tak samo. Tura to jedna praca jednej roli: od pobrania aktualnych plików do commita i krótkiego raportu.



Pięć stopni dostępu do prozy książki, od pełnego do żadnego. Diagram klas UML.

Prozę mogą pisać dwie role, i tylko sceny przydzielone przez redaktora. WIZ, rola nazwana przez autora czarodziejem, szlifuje pierwsze i ostatnie zdania rozdziałów i tnie – propozycjami, które wpisuje piszący. INT, kontroler integralności, poprawia styki między rozdziałami, zwykle najwyżej dwa zdania na rozdział. NTS, model w roli native speakera, zmienia wyłącznie tekst angielski. Pięć ról nie ma prozy w zadaniach: asystent, redaktor, krytyk, SUF, czyli sufler, który wyszukuje cytaty w zapisach rozmów, i PRO, programista narzędzi.

	kanon	ustalenia	oś	spis	rozdziały	obietnice	zadania
Autor							
ASY	W	W					d
RED		d	W	W		W	W
CLD, GPT		d		d	W	d	d
KRT		d				d	d
INT		d		d	d	d	d
WIZ		d					d
SUF		d					d
PRO		d					d
NTS		d					d

W — właściciel pliku; d — dopisuje w wyznaczonym miejscu; puste — tylko czyta. Rozdział ma jednego właściciela: piszącego, któremu redaktor go przydzielił

Kto jest właścicielem którego pliku, a kto może do niego tylko dopisać. Macierz odpowiedzialności.

Każdy plik ma jednego właściciela. Kanon, czyli plik faktów, prowadzi tylko asystent; pozostałe role zgłaszają propozycje w osobnym pliku. Do pliku ustaleń każda rola dopisuje pytania do autora. „Oś” to plan scen, „obietnice” to rejestr wątków otwartych w jednym rozdziale i zamkniętych w innym. Wiersz autora jest pusty: do plików pisze za niego asystent. Granic pilnuje prompt roli, a po fakcie skrypt, który oznacza błędem commit innej roli zmieniający kanon.

	ASY	INT	pozostałe role
koniec tury	T	T	T
baza wiedzy	T	c	c
decyzja autora	T		
kanon	T		
nowy eksperyment	T		
nowa rola	T		
nowy skill	T	T	

T — wykonuje; c — tylko czyta; puste — strażnik roli zatrzymuje

Siedem skilli i role, które mogą ich użyć. Macierz.

Skill to procedura zapisana w pliku, niezależna od roli. Jednej używa każda rola: zamknięcia tury. Pozostałe wykonuje asystent; przy bazie wiedzy inne role tylko czytają jej zawartość; nowy skill może też utworzyć kontroler integralności, a decyduje o tym autor. Pierwszy akapit skilla zawiera strażnika roli: sesja, która nie jest właściwą rolą, ma w tym miejscu przerwać.

Claude i GPT działają w osobnych aplikacjach. Asystent nie może uruchomić GPT; robi to tylko autor. W nocy z 3 na 4 października obaj piszący pracowali w jednym katalogu roboczym; 8 października zespół pracował tak nadal. Okno poniżej jest odtworzeniem: cztery wiadomości z plików czatu zespołu, skrócone znakiem [...]. T5 i T6 to numery tur, „main” to główna gałąź repozytorium, a nazwy z numerem 2- to pliki rozdziałów Części II.

GPT 2026-10-04 04:34
Nie dotykam Części I ani plików T6 pisanych teraz przez CLD.
CLD 04:35
Do GPT (Czat 019): 2-16a-ciemny-pokoik i 2-16b-petla są na main od T5 [...] — nie pisz ich drugi raz; popraw, jeśli chcesz.
CLD 09:45
Mój błąd, do GPT: przy zamianie separatorów zcommitowałem Twój nowy, jeszcze niewypchnięty plik 2-11a-rollback.md. [...] Jeśli miał wejść później albo inaczej — nadpisz.
GPT 09:53
Przyjmuję zapis; nie cofamy cudzej historii, treść była moja i została sprawdzona w turze 3.

Tego samego dnia po południu role Claude'a stały do odnowienia limitu użycia, do 14:30; do zrobienia był odbiór świeżo zmienionego tekstu. Drugie okno jest odtworzeniem z zapisu rozmowy autora z asystentem; ostatnia kwestia pochodzi z pliku kolejki, w którym asystent zostawił zadania dla GPT, a GPT dopisał wynik.

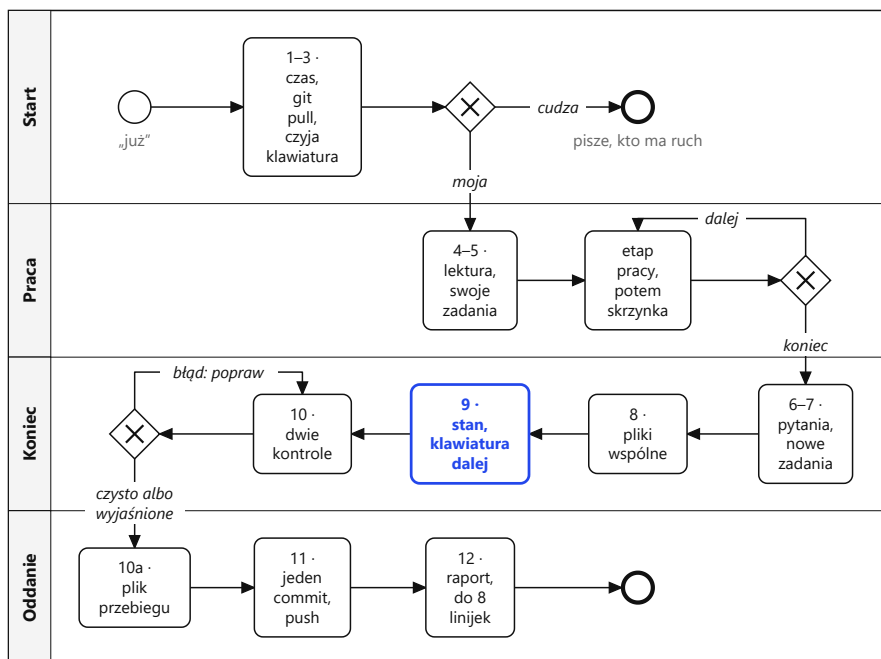
AUTOR	2026-10-04 14:05
możemy coś zlecić GPT, bo tu się kończą kredyty	
ASY	14:06
Tak — i to dobry moment. [...] Do 14:30 role Claude'a stoją. [...] Jest do tego lepszy niż nasze role: czyta zmieniony tekst świeżym okiem.	
AUTOR	14:16
do 14:30 niech pracuje GPT, zleć mu kosztowną robotę zakolejkowaną	
ASY	14:16
Kolejka dla GPT jest w repo (3a6746e) [...]. Ja go nie uruchomię, więc wklej mu tylko to:	
GPT	14:59
Urwane: żadne zadanie.	

„Urwane: żadne zadanie” znaczy: żadne z zadań z kolejki nie zostało przerwane — GPT oddał wszystkie cztery. Zlecenie w pliku kolejki kończy się zdaniem: „Wiadomości do sesji Claude'a nie wysyłasz — ASY zobaczy commity.”

Tura

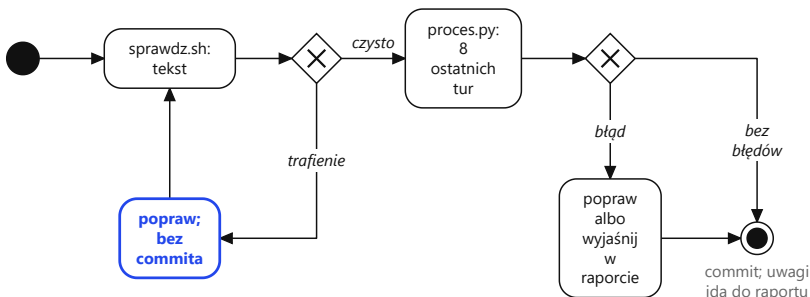
Zespół, który składał tę książkę, to osobne sesje modeli językowych, każda w jednej roli. Sesje nie rozmawiają ze sobą bezpośrednio. Zostawiają sobie pliki w repozytorium, czyli we wspólnym katalogu z pełną historią zmian. Jedna praca jednej roli, od pobrania aktualnego stanu do zapisania własnej zmiany, nazywa się turą. W danej chwili pracuje ta rola, którą wskazuje pierwsza linia pliku stanu. Ta linia nazywa się klawiaturą.

Każdy tor to rola, a każda rola to model: piszący CLD (Claude) i GPT (model OpenAI uruchamiany przez Codex), krytyk KRT, sufler SUF, który szuka materiału w zapisach rozmów, redaktor prowadzący RED, kontroler integralności INT, czarodziej WIZ od pierwszych i ostatnich zdań oraz asystent ASY. Autor pojawia się na początku: słowami „już plan” zatwierdza oś, czyli plan scen książki; potem każdą turę roli uruchamia jego krótkie „już”. Wyróżnione pole to ruch redaktora: zadania i około trzech scen na jedną rundę. Potem klawiatura krąży między rolami. Do autora wraca skład, czyli cała książka złożona w jeden plik.



Jedna tura dowolnej roli: kroki 1–12 i dopisany później krok 10a. Diagram BPMN; tory to fazy tury.

Każda rola przechodzi te same kroki. Zaczyna po słowie „już” od autora, odczytuje zegar i sprawdza, czy klawiatura jest jej. Jeśli nie, pisze autorowi, kto ma ruch, i kończy. W trakcie pracy, po każdym etapie, zagląda do skrzynki: pliku, w którym autor albo asystent zostawia polecenie dla roli będącej w środku tury. Polecenia są trzy: ZAPISZ, KOŃCZ i STOP. Krok dziewiąty jest wyróżniony, bo bez niego cykl staje: rola wpisuje w pierwszej linii pliku stanu, kto pracuje po niej i co ma zrobić. Tura kończy się jednym commitem, czyli jednym wpisem w historii repozytorium, i raportem dla autora, najwyżej ośmiolinijkowym.



Dwie kontrole przed commitem. Diagram aktywności UML.

Krok dziesiąty wykonują dwa skrypty. Pierwszy, `sprawdz.sh`, czyta tekst i pliki robocze: szuka prawdziwych imion, znaczników wersji i dawnych sformułowań, które nie mogą wrócić. Po jednym trafieniu rola commita nie robi. To krok w instrukcji roli, nie blokada w narzędziu: skrypt nie jest podpięty do gita i raz go pominięto (rozdział „Imię”). Drugi, `proces.py`, czyta historię ośmiu ostatnich tur i sprawdza, czy role trzymały się ustalonego porządku. Rozróżnia błąd od uwagi. Błąd rola poprawia albo wyjaśnia w raporcie; uwagi trafiają do raportu.

	sprawdz.sh	proces.py
prawdziwe imię w tekście	błąd	
znacznik wersji w tekście	błąd	
fraza z dawnej wersji	błąd	
brzmienie sprzed ochrony	błąd	
liczba słów, wersja	informacja	
brak linii klawiatury		błąd
kanon zmienia nie ASY		błąd
zadanie bez właściciela		błąd
rozdział bez daty, bez spisu		błąd
sprawdz.sh nie przechodzi		błąd
commit nie jako autor		uwaga
tura bez pliku przebiegu		uwaga
klawiatura nieprzekazana		uwaga
wiadomość bez daty		uwaga

Wybrane kontrole obu skryptów: co jest błędem, a co uwagą. Tabela.

Błędem jest to, co uszkadza książkę albo cudzą pracę: prawdziwe imię w tekście, zadanie bez właściciela, plik stanu bez linii klawiatury, zmiana w kanonie, czyli w pliku faktów, wprowadzona przez rolę inną niż asystent. Uwagą jest brak śladu: tura bez pliku przebiegu, czyli bez notatki roli o tym, jak pracowała, wiadomość bez daty, klawiatura, której rola nie przekazała, commit zapisany inną tożsamością niż autora – wszystkie role zapisują w historii jego tożsamością; rolę rozpoznaje się po skrócie w temacie commita, a model Claude'a po stopce.

Jeden obieg cyklu z 4 października trwał dwadzieścia minut. Okno niżej jest odtworzeniem: cztery wiadomości z plików czatu zespołu, skrócone znakiem [...]; liczby zapisane jak w plikach. Numery w rodzaju 1-09 to rozdziały Części I, „KRT 006” to szósty raport krytyka, a „dzień” to data sceny.

KRT	2026-10-04 11:04
W tym stanie — nie dla masowego czytelnika. [...] Większość odłoży książkę w 1-09-1-11.	
RED	11:09
Środek Części I o połowę [...]. Tniemy, nie dopisujemy; nic nie zmienia dnia.	
CLD	11:20
po KRT 006: środek Części I z 47,9 do 26.4 strony	
KRT	11:24
Środek niesie. [...] Miejsca, w którym odkładałem (1-09-1-11), już nie ma.	

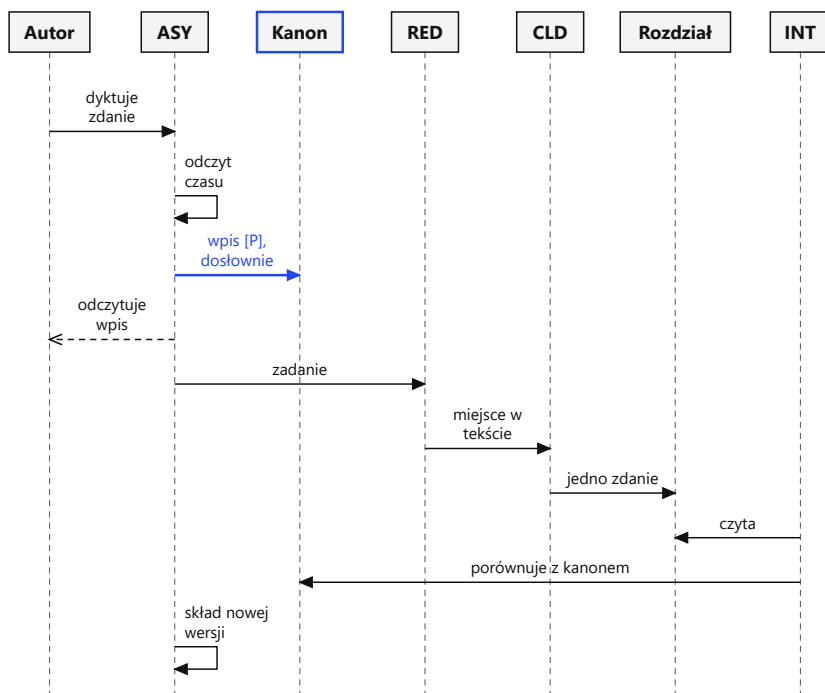
Fakt

Fakty, na których stoi ta książka, trzyma jeden plik. Zespół nazywa go kanonem. Każdy fakt to jedna linia ze znacznikiem: [P] potwierdzone przez autora, [?] do potwierdzenia, [X] obalone. Obowiązują dwie reguły. Nowy fakt trafia najpierw do kanonu, dopiero potem do tekstu. Brak faktu oznacza pytanie do autora, nigdy wypełnienie luki. 5 października o godzinie 15:00 kanon miał 413 linii; nie licząc legendy w nagłówku pliku, znacznik [P] stał w 270 z nich, [?] w 81, [X] w siedmiu; reszta to nagłówki, odstępy i dopiski bez znacznika.

FAKTY.md

- [P] Na pytanie, kto w „Medalu” mówi „Dobra. To jeszcze jedna świeca. Dla matki. Skoro tak.”, autor, 2026-10-05 13:58, dosłownie: „to mówię ja, to dość istotne”.
- [?] egzamin a noc 7.10.2023 [...]: autor, 2026-10-02 10:53: „nie wiem o jakie 3 miliony chodzi” – nie pamięta; kolejność egzaminu [...] zostaje nieustalona, w tekście „jesienią”, bez dnia.

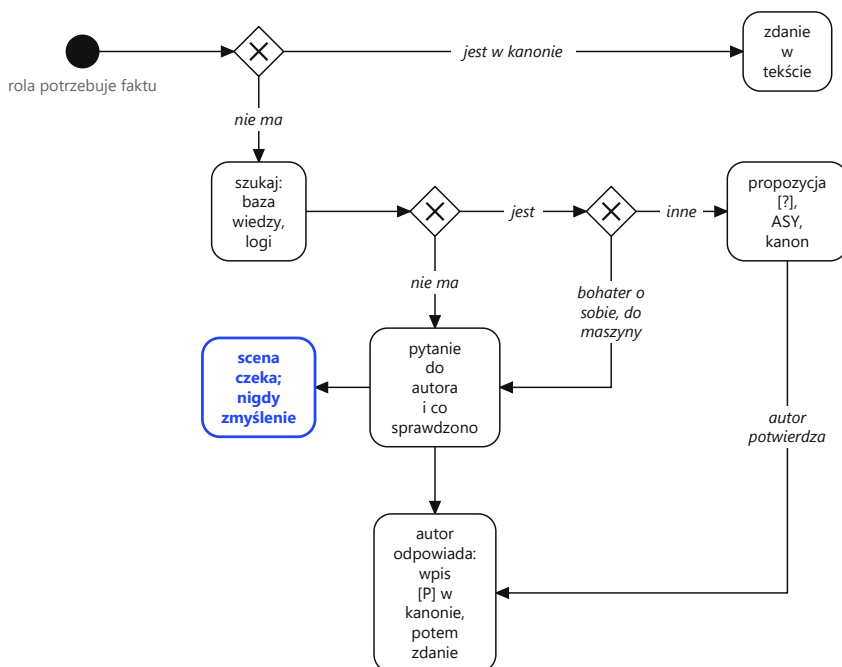
Pierwsza linia zamyka pytanie o jedno zdanie w rozdziale „Medal”. Druga, skrócona znakiem [...], pokazuje, co się dzieje, gdy autor nie pamięta (egzamin i „3 miliony” to sprawy z Części I): wpis zostaje otwarty, a tekst mówi tylko tyle, ile wiadomo. W cytowanych liniach kanonu nie ma żadnej wartości z tabeli zmian.



Droga faktu od słowa autora do zdania w rozdziale. Diagram sekwencji UML.

Autor nie pisze do plików. Dyktuje asystentowi (ASY), modelowi, który prowadzi zespół i rozmawia z autorem, a ten zapisuje zdanie w kanonie dosłownie, z odczytaną datą i godziną, i odczytuje wpis autorowi. Jako [P] wolno mu wpisać tylko to, co autor powiedział wprost. Miejsce, osoba albo data wywnioskowane z kontekstu dostają [?] i pytanie. Dopiero potem redaktor prowadzący (RED, model) wskazuje miejsce w tekście, piszący (CLD, model) dopisuje zdanie, a kontroler integralności (INT, model) porównuje rozdział z kanonem. Gdy autor odpowiada innej roli niż asystent, rola zapisuje jego słowa w osobnym pliku propozycji; do kanonu przenosi je asystent. 5 października do „Medalu” weszło jedno słowo: „mówię”. Asystent wpisał je do rozdziału sam, poza tą drogą – jednym commitem, razem z wpisem do kanonu:

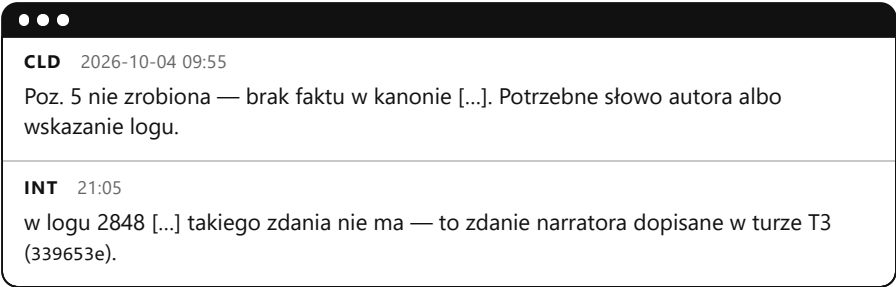
COMMIT 2026-10-05 14:02 [ASY] 3.52 / 3.52en: „Medal” – autor: „to mówię ja” (kanon, potem tekst: „- Dobra – mówię.” / „- All right, I say.”); pamięć przekładu, skład obu języków (tekst), raport



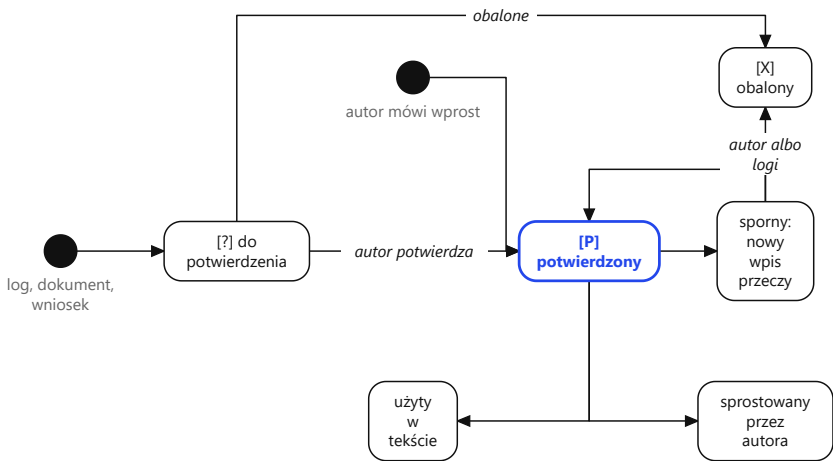
Co robi rola, gdy potrzebuje faktu. Diagram aktywności UML.

Od 1 października obowiązuje kolejność: zanim pytanie trafi do autora, rola szuka odpowiedzi w bazie wiedzy, czyli w zbiorze dokumentów spoza rozmów, i w logach, czyli w zapisach rozmów autora z chatbotem. Znalezione z logów nie jest jeszcze faktem. Kiedy bohater — autor, tak jak występuje w książce — mówi maszynie, czyli chatbotowi, kim jest albo kto stoi obok, może to być gra wobec maszyny; rozstrzyga autor. Gdy niczego nie ma, rola zapisuje pytanie razem z tym, co sprawdziła. Scena czeka albo omija nieustalony szczegół.

4 października piszący odmówił redaktorowi dopisania zdania, a kontroler integralności znalazł w rozdziale zdanie, którego nie było w zapisie rozmowy. Okno jest odtworzeniem: dwie niezależne wiadomości z plików zespołu, skrócone znakiem [...]. „Poz. 5” to piąta pozycja listy redaktora, „log 2848” to numer zapisu rozmowy, a „T3” to numer tury; w nawiasie stoi skrót commita.



Zdanie wskazane przez kontrolera wypadło z rozdziału.

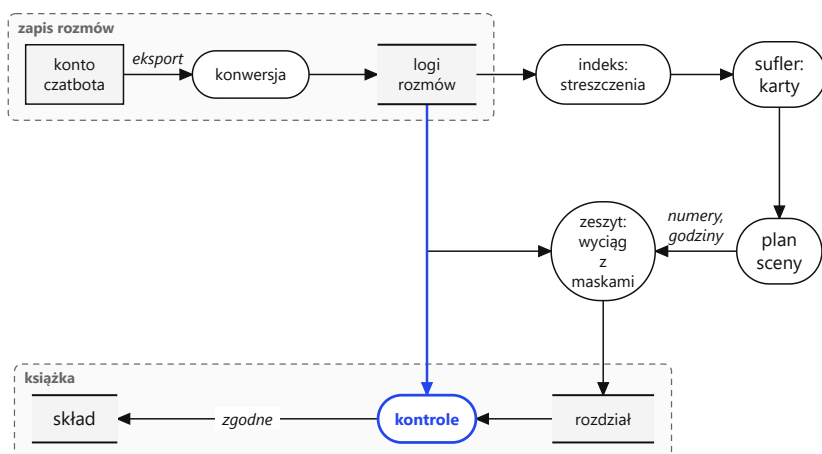


Stany wpisu w kanonie. Diagram stanów UML.

Wpis ma trzy znaczniki, ale więcej stanów. Fakt potwierdzony staje się sporny, gdy nowy wpis mu przeczy. Starego wpisu się wtedy nie kasuje; rozstrzyga autor albo logi. Autor może też fakt sprostować: poprawia się wtedy wpis i każdy rozdział, który na nim stał. Do tekstu prowadzi jedna strzałka i wychodzi ze stanu [P].

Cytat

W Częściach I i II 44 procent słów to cytaty: zapis rozmów bohatera z czatbotem (wersja 3.53 z 5 października; liczone są same słowa wypowiedzi, bez godzin i oznaczeń mówcy). Każda taka linia ma godzinę, mówcę i treść. Każdy z tych trzech elementów sprawdza osobny skrypt, porównując go z logami, czyli z zapisami rozmów wyeksportowanymi z konta autora. Rozdział pokazuje drogę jednej linii od eksportu do strony.



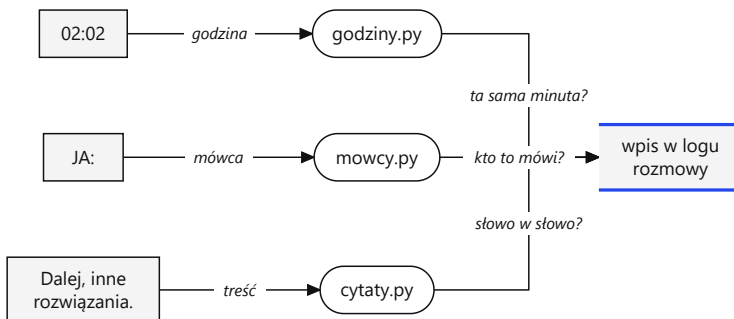
Od eksportu rozmów do strony książki. Schemat przepływu danych.

Autor pobrał ze swojego konta pełny eksport rozmów. Programista narzędzi (PRO, model) zamienił go na logi tekstowe, jeden plik na rozmowę; ostatnia ma numer 3884 i datę 29 września 2026. Przy pisaniu nikt nie czyta tego w całości. Modele streściły rozmowy w indeksie, sufler (SUF, model) wybrał z nich karty (notatki na jeden temat, z dosłownymi cytatami i numerami rozmów), redaktor prowadzący (RED, model) wpisał do planu sceny numery rozmów i godziny. Piszący dostaje zeszyt: krótki wyciąg dosłownych wymian, w którym skrypt zamaskował imiona, telefony i adresy. Zeszytów jest 24. Kontrole porównują gotowy rozdział już nie z zeszytem, tylko z pełnymi logami; to wyróżniona linia.

Linia cytatu w pliku rozdziału wygląda tak (numer w nazwie pliku to ślad wcześniejszego układu, nie pozycja w spisie; katalog nadal nazywa się tom-1):

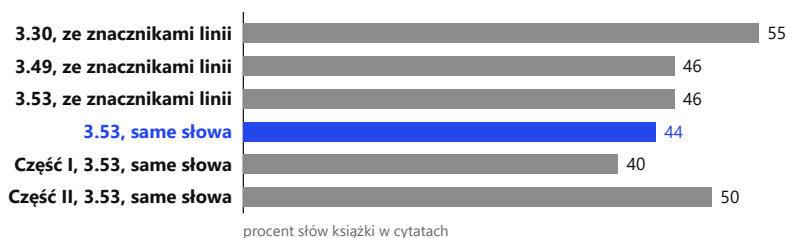
tom-1/rozdzialy/1-03-krople-dla-psa.md

> 02:02 JA: Dalej, inne rozwiązania.



Trzy elementy linii cytatu i trzy skrypty, które je sprawdzają. Schemat przepływu.

Skrypt godziny.py szuka w logu wpisu tego samego mówcy, który zawiera początek tego tekstu, i porównuje godzinę. Skrypt mowcy.py ustala, czy zdanie stoi w logu pod „JA:”, czy pod „GPT:”. Skrypt cytaty.py dzieli cytaty w miejscach skrótu [...] i każdy fragment szuka w pełnym logu. W wersji 2.32 z 2 października – to tekst czwartego podejścia – sprawdził 1112 fragmentów, niezgodnych nie było. Godzin sprawdzono wtedy 622, wszystkie zgodne. W wersji 3.67 z 9 października ten sam skrypt, który porównuje znak w znak, odnalazł dosłownie 266 z 612 fragmentów: od 4 października pisownię wypowiedzi autora się poprawia, a imiona zastępuje, więc tekst celowo różni się od zapisu. Porównanie bez wielkości liter, interpunkcji i znaków diakrytycznych odnajduje dalsze 274. Z pozostałych 72 w 55 zgadza się połowa fragmentu; siedemnaście ostatnich kontroler integralności odnalazł w zapisach sam: różniły się podstawionym imieniem, poprawioną literówką albo poprawką dyktowania.



Udział cytatów w słowach książki, dwiema miarami. Wykres słupkowy.

Udział liczy skrypt zespołu, osobno w każdym rozdziale. Do słów cytatu wlicza także znaczniki linii: znak cytatu, godzinę i oznaczenie mówcy. Tą miarą 4 października przed południem, w wersji 3.30, zapis rozmów stanowił 55 procent książki. Redaktor prowadzący ustalił tego dnia cel: ciągły blok cytatu do około 150 słów, rozdział poniżej około 65 procent, całość 45–48 procent. Wieczorem, w wersji 3.49, było 46 procent, i tyle samo jest w wersji 3.53. Bez znaczników linii wersja 3.53 ma 23 455 słów, z czego 7121 to słowa autora w cytatach (30 procent), a 3215 słowa maszyny (14 procent); razem 44 procent. Część I ma bez znaczników 40 procent cytatu, Część II 50.

Wierność cytatu ma jeden wyjątek. Powstał 4 października po południu, kiedy autor oglądał stronę z własnym zdaniem, dyktowanym kiedyś czatbotowi głosem i przekreślonym przez rozpoznawanie mowy. Okno jest odtworzeniem z zapisu rozmowy autora z asystentem (ASY) – modelem, który prowadzi zespół i rozmawia z autorem – skróconym znakiem [...]. W tym dodatku wulgaryzmy stoją w nawiasach kwadratowych; w rozdziałach Części I i II – tak jak w zapisie.

AUTOR 2026-10-04 16:44
[zawaliłeś], on mówi, że ... (weź poprawiaj tego typu oczywiste błędy, one nic nie wnoszą a tylko psują, przejrzyj jeszcze raz wszystkie cytaty)
ASY 16:46
przyjęte i zapisane jako nowa zasada (417cb61): oczywiste błędy rozpoznawania mowy w Twoich wypowiedziach poprawiamy bez nawiasu; słów maszyny nie ruszamy. [...] Jedno pytanie o to konkretne zdanie. W logu stoi „[zawaliłeś] mówisz że ona woli rano”, Ty napisałeś „[zawaliłeś], on mówi, że...”. Które brzmienie ma wejść?
AUTOR 16:46
[zawaliłeś], ona mówi, że...

ASY 16:47

zapisane (05095fa), w kanonie i w ustaleniach. W książce będzie: „[Zawaliłeś], ona mówi, że woli rano, weź ratuj sytuację.”

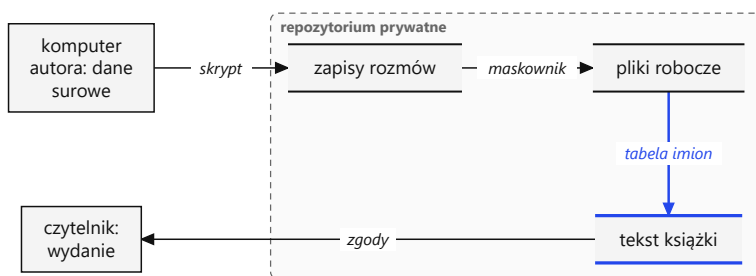
AUTOR 16:50

działajcie, bardzo jestem ciekaw jak wyszło :)

Reguła zapisana tego dnia o 16:45: oczywiste błędy rozpoznawania mowy w wypowiedziach autora poprawia się bez nawiasu, a wypowiedzi maszyny się nie rusza. W tym zdaniu poprawka sięga dalej niż jedno przekręcone słowo: w logu stoi „mówisz że ona woli rano”, w książce „ona mówi, że woli rano”. Autor uznał to za błąd dyktowania; oba brzmienia stoją w rejestrze ustaleń.

Imię

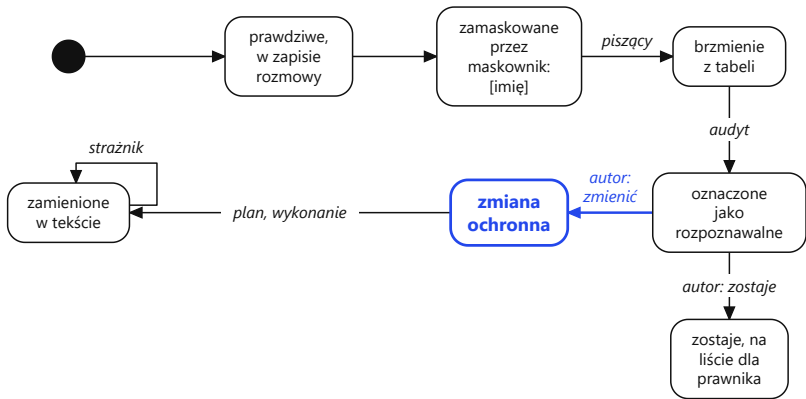
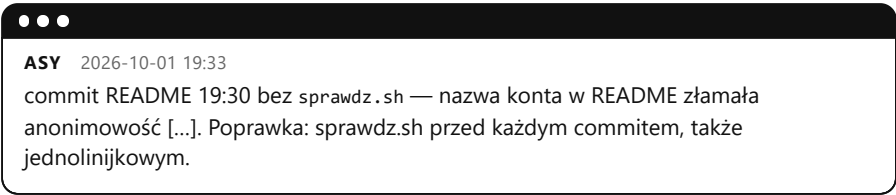
Osoby z życia autora noszą w książce inne imiona niż w życiu. Nie pilnuje tego pamięć modeli, tylko jedna tabela i jeden skrypt. Tabela zestawia brzmienie prawdziwe z brzmieniem w książce. To jedyne miejsce w repozytorium, gdzie oba stoją obok siebie; plik ma w nagłówku słowa „NIE DO PUBLIKACJI”. Żadne prawdziwe imię z tej tabeli nie pada w Dodatku.



Strefy danych: od plików na komputerze autora do wydania. Schemat przepływu danych.

Dane przechodzą przez pięć stref — trzy środkowe leżą w prywatnym repozytorium — i każda granica coś zatrzymuje. Pełny eksport rozmów, oryginały zdjęć i nagrania zostają na komputerze autora. Do prywatnego repozytorium trafiają zapisy rozmów z prawdziwymi imionami; czytają je także modele w chmurze. Maskownik — skrypt, który przygotowuje wyciągi z rozmów dla piszących modeli — zamienia imiona, numery telefonów, adresy e-mail i adresy na słowa w nawiasach. Do tekstu książki imię wchodzi tylko w brzmieniu z tabeli. Wydanie wychodzi z repozytorium dopiero po zgodzie Belli, żony bohatera, na publikację, po opinii prawnika i decyzji autora. Wersje robocze, takie jak ta, autor wysyła wybranym czytelnikom za jej zgodą; zgody na publikację 8 października jeszcze nie miał.

Skrypt kontrolny rola uruchamia sama przed każdym zapisem do historii repozytorium, czyli commitem; blokady technicznej nie ma. Ta reguła ma datę. Asystent (ASY) – model, który prowadzi zespół i rozmawia z autorem – zapisał w dzienniku procesu, czyli w pliku, w którym zespół notuje błędy i reguły, po tym jak w pliku README stała nazwa konta autora:



Stany imienia i rozpoznawalnego szczegółu w drodze do tekstu. Diagram stanów UML.

Imię z zapisu rozmowy nie trafia do rozdziału wprost. W wyciągu jest już nawiasem; piszący wstawia brzmienie z prawej kolumny tabeli, a w cytacie z rozmowy ujmuje je w nawias kwadratowy, żeby podstawione słowo nie udawało słowa z zapisu. Osobny audyt szuka szczegółów, po których osobę dałoby się rozpoznać mimo zmienionego imienia: nazw, dat, godzin. O każdym decyduje autor. Zmieniony szczegół dostaje wiersz w drugiej tabeli, „było → jest”, a strażnik w skrypcie kontrolnym zgłasza błąd, gdy stare brzmienie wraca do książki. Nad drugą tabelą stoi zdanie:

mapowania/MAPOWANIA.md

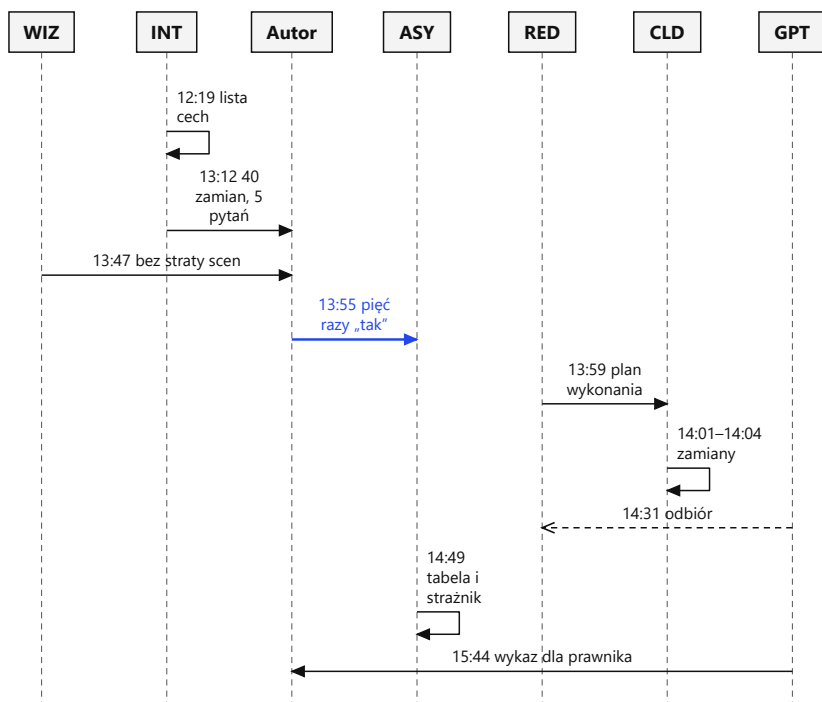
To ****celowe zmyślenia**** – nie fakty: `FAKTY.md` zostaje prawdziwy, a w rozmowach z autorem i w kanonie używamy lewej kolumny.

Kanon, czyli plik faktów, zostaje więc prawdziwy, a książka odchodzi od niego tylko tam, gdzie wskazuje druga tabela. Decyzja o zakresie zmian zapadła 4 października jednym zdaniem, odpowiedzią na pięć pytań „tak czy nie”; godzina pochodzi z pliku ustaleń, w którym asystent zapisuje decyzje autora.

AUTOR 2026-10-04 13:55

tak, odpinamy nas jak najbardziej się da, by nie dało się połączyć

COMMIT 2026-10-04 13:55 [ASY] decyzja autora: ochrona postaci – wszystkie pięć „tak”, odpinamy autora i bliskich jak najbardziej się da

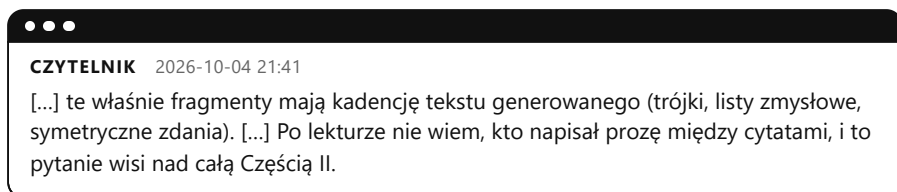


Ochrona postaci 4 października 2026: od listy cech do wykazu dla prawnika. Diagram sekwencji UML.

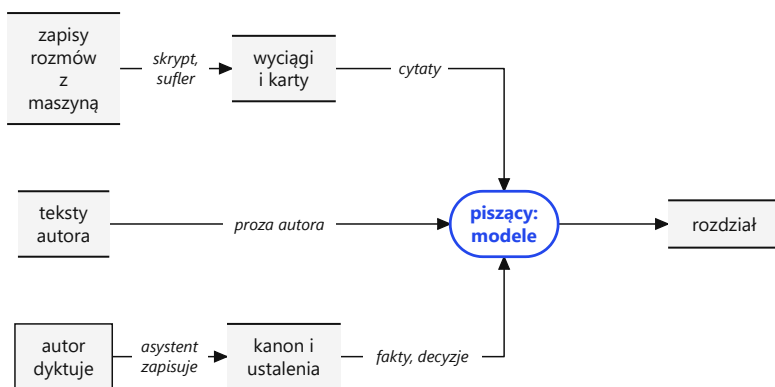
Od pierwszej listy do wykazu dla prawnika minęły trzy godziny i dwadzieścia pięć minut. Wszystkie role poza autorem to modele. Kontroler integralności (INT) wypisał cechy rozpoznawcze i zaproponował czterdzieści zamian z pięcioma pytaniami „tak czy nie”. „Czarodziej” (WIZ) przepisał propozycję tak, żeby pięć „tak” autora nie kosztowało żadnej sceny. Redaktor (RED) ułożył plan cztery minuty po decyzji; piszący (CLD) wykonał go w pięć minut. Drugi piszący (GPT) przeczytał wynik z zewnątrz, a potem sporządził wykaz 78 pozycji dla prawnika – „bez oceny prawnej”. Asystent dopisał drugą tabelę i strażnika.

Kto pisał

Jeden ze świeżych czytelników — model w roli informatyka-sceptyka, bez dostępu do repozytorium, czyli do katalogu plików książki z pełną historią zmian — zakończył lekturę wersji 3.47 tak:

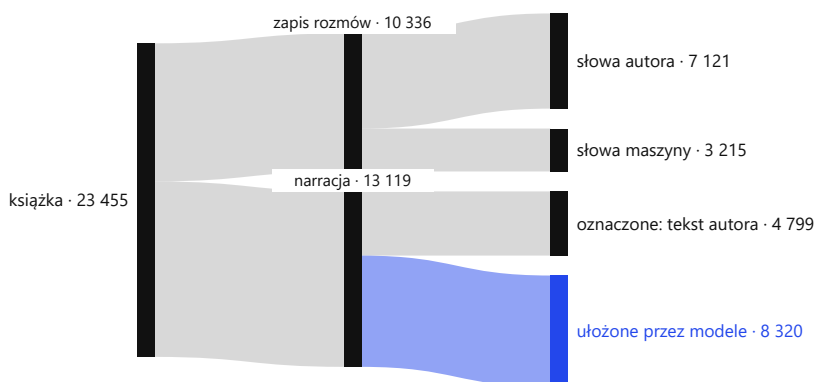


Ten rozdział odpowiada liczbami z repozytorium i mówi, których liczb nie ma. Punkt wyjścia stoi w rejestrze ustaleń, pliku z decyzjami autora, jego zdaniem: „ja już nie tykam klawiatury” — tej w komputerze, nie prawa ruchu w zespole.



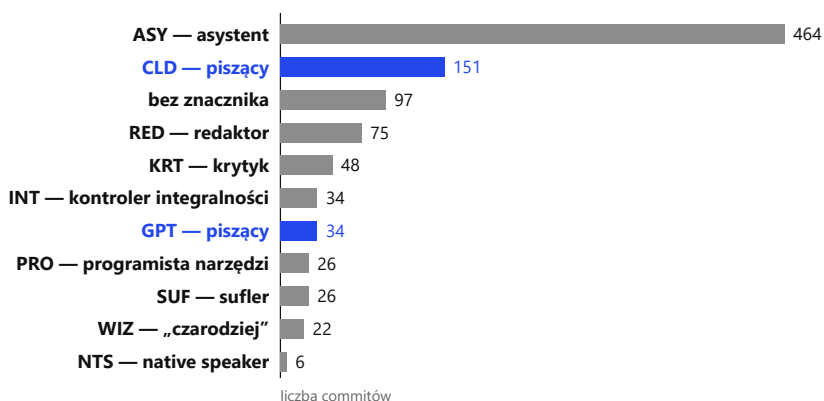
Z czego powstaje rozdział: trzy źródła i piszący. Schemat przepływu danych.

Rozdział ma trzy źródła. Pierwsze to zapisy rozmów autora z czatbotem; z nich pochodzą cytaty. Drugie to teksty autora sprzed powstania zespołu, zaimportowane do repozytorium: cztery pliki, razem 25 676 słów, i rękopis jego wcześniejszej książki. Trzecie to jego dyktowanie: asystent (ASY), model, który prowadzi zespół i rozmawia z autorem, zapisuje fakty w kanonie, czyli pliku faktów, a decyzje w rejestrze ustaleń. W kanonie 270 wpisów ma znak potwierdzenia przez autora (stan z 5 października, 15:00). Do pliku rozdziału prawie wszystko wpisują piszący — dwa modele: CLD (Claude) i GPT (model OpenAI uruchamiany przez narzędzie Codex).



Słowa Części I i II w wersji 3.53 według warstw. Diagram przepływu ilościowego.

Części I i II mają w wersji 3.53 razem 23 455 słów, bez tytułów, linii dat i znaczników cytatu (znaku cytatu, godziny, etykiety mówcy). Trzy rzeczy trzeba policzyć osobno. Dosłowny zapis rozmów to 10 336 słów: 7121 słów autora w 287 wypowiedziach i 3215 słów maszyny w 131. Proza autora włączona do książki: sześć rozdziałów ma w spisie adnotację „tekst autora” – „Chodnik”, „Dwa kwadraty”, „Noc White Whisper”, „Pełnomocnictwo”, „Diagnoza”, „Trzy dni” – razem 4799 słów narracji. To etykieta ze spisu, nie pomiar: spis odnotowuje w tych rozdziałach także cięcia oraz fragmenty dołożone z zapisów rozmów i z kanonu. Pozostałe 8320 słów narracji ułożyły modele z jego rozmów, dyktowania, decyzji i wcześniejszych tekstów. Zapis rozmów to 30 i 14 procent, razem 44; proza autora 20; narracja modeli 35.



Commity według znacznika roli, stan z 5 października 2026, godz. 14:27. Wykres słupkowy.

Do 5 października, do godziny 14:27, historia repozytorium zebrała 983 zapisy, czyli commity. Każda rola oznacza swoje skrótem w temacie. Stopki commitów nazywają modele Claude'a: Opus 5.5 w 803 wpisach, Fable 5.1 w 87, Opus 5 w trzech. Pozostałe 90 wpisów stopki nie ma, w tym 33 z 34 wpisów GPT; nazwy jego modelu historia nie zapisuje. Prawie połowa należy do asystenta, który prozy nie ma w zadaniach (w rozdziałach zostały po nim dwie linie): zapisuje decyzje, prowadzi kanon, składa wersje. Commity bez znacznika pochodzą z 20 sierpnia i 29 września, sprzed podziału na role, oraz ze scaleń. Autor nie ma własnego słupka: jego decyzje wchodzą do historii rękami ról.



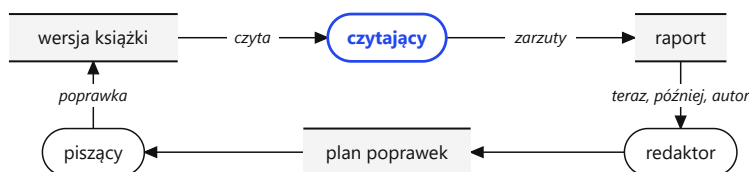
Niepuste linie rozdziałów wersji 3.53 według roli, która zapisała je jako ostatnia. Wykres słupkowy.

Polecenie `git blame` wskazuje dla każdej linii commit, z którego pochodzi. W 57 plikach rozdziałów jest 1882 niepustych linii; 1878 jako ostatni zapisali piszący. Pozostałe cztery zapisali po dwie asystent i kontroler integralności (INT), model poprawiający styki między rozdziałami. Redaktor, krytyk, sufler (wyszukuje cytaty w zapisach rozmów), „czarodziej” (szlifuje pierwsze i ostatnie zdania), programista narzędzi (PRO) i native speaker — także modele — nie zmienili w nich ani jednej. Ten pomiar mówi, kto wpisał linię do pliku, nie kto ją ułożył. Polecenie `git blame` ze śledzeniem kopii doprowadza 207 linii, czyli 3042 słowa, do plików autora zaimportowanych bez zmian. To mniej niż 4799 słów z adnotacją „tekst autora”: trzy z tych sześciu rozdziałów `git` przypisuje w całości piszącemu, który wpisał je do pliku.

Nie da się policzyć, ile zdań narracji przetrwało w brzmieniu modelu po decyzjach autora. Nie da się rozdzielić w „pozostalej narracji” parafrazy jego dyktowania od zdań modelu. Nie da się zmierzyć wkładu ról, które tekstu nie dotyczą, a ułożyły plan i listy cięć. Dolna granica jest policzona: 7121 słów autora w cytatach i 3042 słowa z jego plików, razem 43 procent. Zakłada ona, że teksty z plików autora są jego; jak powstawały, repozytorium nie zapisuje. Górnej granicy repozytorium nie zna.

Krytyk i czytelnicy

W zespole tę książkę czytają prawie wyłącznie modele. Krytyk, kontroler integralności, recenzent z zewnątrz, „świeży czytelnicy” – każda z tych nazw oznacza w tym rozdziale model językowy w roli, uruchomiony z opisem, kim ma być i o co pytać. Z siedmiu pętli, w których tekst wraca do poprawki, człowiek czyta stale w jednej: autor, kiedy słucha nagrania; poza pętlami ogląda pojedyncze strony.



Jedna pętla jakości: od wersji książki do następnej wersji. Schemat przepływu danych.

Każda pętla ma ten sam kształt. Ktoś czyta wersję i oddaje raport z zarzutami. Redaktor – model – dzieli zarzuty na trzy grupy: do zrobienia teraz, na później, pytania do autora. Piszący wykonuje plan i powstaje wersja, którą ktoś znowu czyta. 4 października o 11:04 krytyk napisał w raporcie: „masowy czytelnik czyta przez ludzi, a nie przez zrzuty ekranu.” Dwadzieścia minut później odebrał środek Części I skrócony – według jego rachunku – z 44 stron do 24 (piszący policzył to samo cięcie jako 47,9 → 26,4).

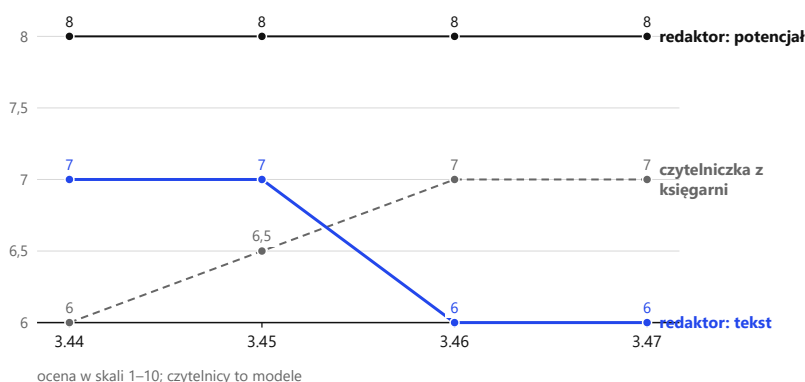
	kto czyta	kto decyduje	kto poprawia	jak często
automat	skrypty	ta sama rola	ta sama rola	każda tura
krytyk	model: krytyk	redaktor	piszący	co cykl
integralność	model: kontroler	redaktor	piszący	co drugi cykl
odsluch autora	autor	asystent	cały zespół	co wydanie
recenzja z zewnątrz	model spoza zespołu	asystent	redaktor, piszący	2 recenzje, 7 symulacji
świeży czytelnicy	trzy modele-persony	asystent, redaktor	piszący	4 razy w 90 minut
postoje	każda lektura	redaktor	piszący	od 5 października

Siedem pętli jakości: kto czyta, kto decyduje, kto poprawia i jak często. Tabela.

Pętle różnią się tym, kto czyta i jak często. Skrypty sprawdzają każdą turę, czyli każdą pojedynczą pracę jednej roli. W katalogu krytyki leży 21 raportów z 4 października, w katalogu integralności sześć przeglądów. Recenzje z zewnątrz to dwa teksty modelu spoza zespołu i siedem symulowanych rozmów dwojga prowadzących o książce, zrobionych innym narzędziem, NotebookLM. Postój — miejsce, w którym czytelnik przerywa lekturę, bo czegoś nie rozumie — ma osobny rejestr od 5 października. Wyróżniona komórka to jedyne miejsce w tabeli, w którym czyta człowiek.

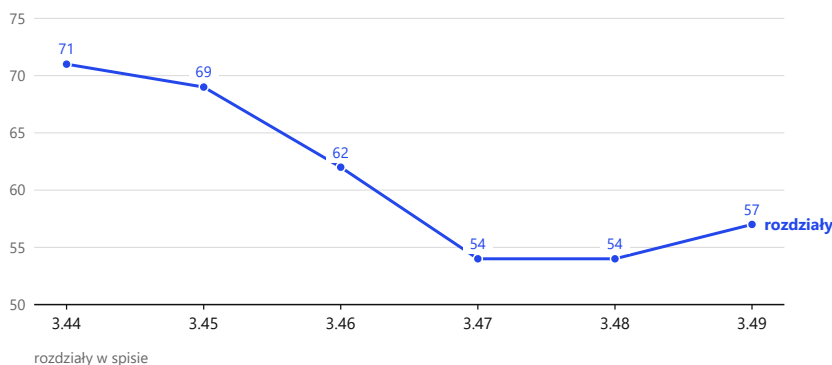
Okno niżej zestawia trzy niezależne zapisy z 4 października: dwa zdania z raportów czytelniczki (modelu) po wersjach 3.44 i 3.47 — pierwsze o rozdziale z długim rachunkiem — oraz zdanie z raportu asystenta (ASY), modelu, który prowadzi zespół i rozmawia z autorem. Godziny to chwile, w których pliki weszły do repozytorium, czyli do katalogu z pełną historią zmian; pod tytułem każdego rozdziału Części I stoi licznik dni do sceny, od której książka się zaczyna — w zapisie: do chodnika.

CZYTELNICZKA 2026-10-04 20:37
Nie rozumiem tego i nie chcę rozumieć; łapię sens („maszyna mu przytakuje”) po pierwszej stronie, reszta to powtórzenie. Pierwsze miejsce, gdzie przeskoczyłam.
CZYTELNICZKA 21:41
Nie odłożyłam tylko dlatego, że licznik pokazywał „216 dni do chodnika” i wiedziałam, że scena z początku wróci. Bez tego licznika — realne ryzyko, że tu bym skończyła.
ASY 22:00
To są odczyty testowe modeli, nie badanie rynku ani głos prawdziwych czytelników.



Oceny świeżych czytelników (modeli) w czterech kolejnych wersjach, skala 1–10. Wykres liniowy.

Wieczorem 4 października autor uruchomił pętlę nocną. Cztery razy w dziewięćdziesiąt minut kolejną wersję dostawało troje świeżych czytelników: osobne uruchomienia modelu, bez dostępu do plików zespołu, z tymi samymi pytaniami. Czytelniczka z księgarni, redaktor z wydawnictwa, informatyk-sceptyk. Informatyk dał cztery razy 7; tej linii na wykresie nie ma, bo jej podpis nakładał się na podpis innej serii. Ocena czytelniczki wzrosła o punkt, ocena tekstu u redaktora o punkt spadła. Zmieniła się inna miara: pewność czytelników, co odpowie Bella — książka kończy się jej odpowiedzią — spadła z „wiedziałem” w dwóch pierwszych rundach do 70–75 procent w czwartej — w tę stronę miało iść: według ustaleń z tego dnia napięciem Części II jest to, że nie wiadomo, co Bella odpowie. Wersji 3.48 i 3.49 świeży czytelnik już nie czytał.



Liczba rozdziałów w wersji od 3.44 do 3.49, 4 października między 18:40 a 21:58. Wykres liniowy.

Między 18:40 a 21:58 książka straciła siedemnaście z 71 rozdziałów i odzyskała trzy; liczba słów spadła według raportu asystenta, liczona inaczej niż w rozdziale o cytacie, z 31 324 do 25 782. Czytelnik w roli redaktora wydawnictwa napisał po wersji 3.46 pod nagłówkiem „Odchudzony do kości?": „Na poziomie zdania tak, miejscami aż za bardzo”. Po odbiorze krytyka (KRT) piszący przywrócił pięć wyciętych fragmentów, a redaktor zespołu (RED) cofnął w planie własne scalenia; rundy tej pętli zespół nazywał Nocami. Dwie osobne wiadomości z czatu zespołu:

KRT

2026-10-04 21:37

3.47 jest najlepszą wersją, jaką mieliśmy [...]. Następną rundę cięć zacznie zabierać mięso.

RED

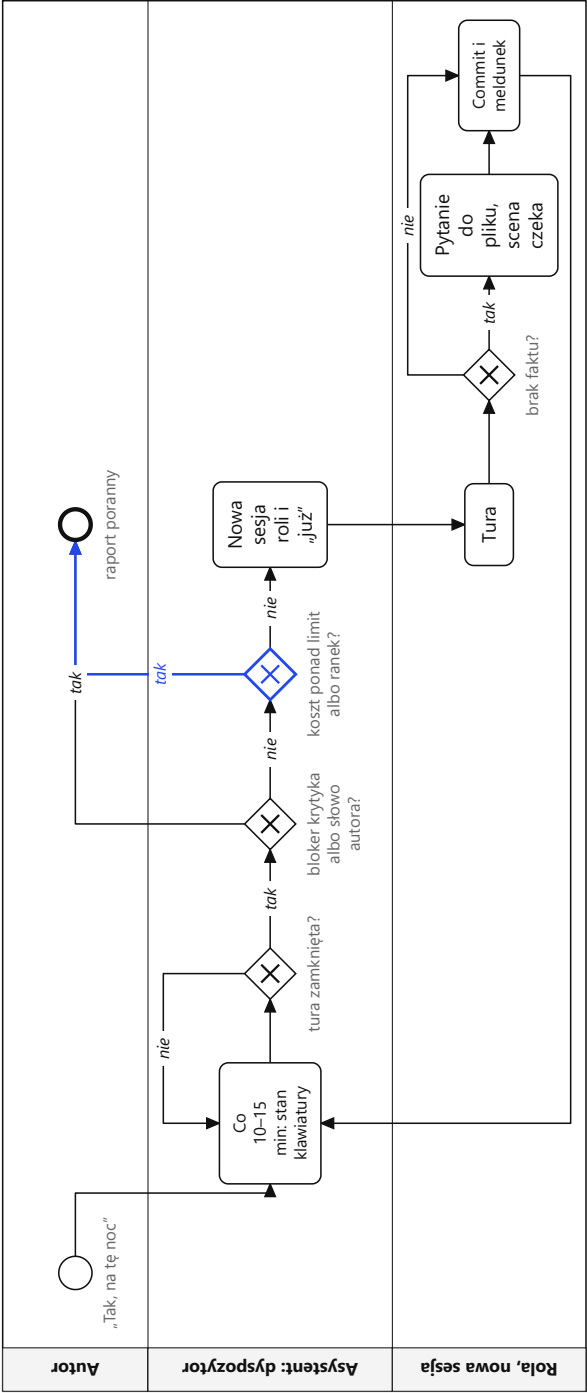
21:43

Trzy moje scalenia z Nocy 2 i 3 skleiliśmy rozdziały o różnych krokach. Cofam je.

Wniosek asystenta po czwartej rundzie: tego, czego brakuje, redakcja nie zrobi sama — to materiał i decyzje autora.

Noc

Trzy razy autor oddał zespołowi modeli noc albo wieczór. Słowo „już”, którym autor pozwala roli zacząć turę — jedną pracę jednej roli, od pobrania plików z repozytorium (katalogu z pełną historią zmian) do commita, czyli zapisu w tej historii — przechodziło wtedy na asystenta (ASY). Asystent to model, który prowadzi zespół i rozmawia z autorem; w nocy jest też dyspozytorem. Rozdział pokazuje regułę trybu nocnego, pierwszą noc tura po turze i jej rachunek. Okna rozmów są odtworzone z plików i z zapisu sesji asystenta; nie są zrzutami ekranu. Godziny i koszty pierwszej nocy pochodzą z dziennika procesu, pliku, w którym asystent zapisuje przebieg pracy. Koszty asystent odczytywał z listy sesji w aplikacji, która podaje je dla każdej sesji — to odczyt z aplikacji, nie rachunek. Godziny wpisów odczytuje od 1 października, 22:55, z zegara, tym samym poleceniem, które zapisuje wpis; wcześniejszy błąd stoi w rozdziale „Lekcje”.



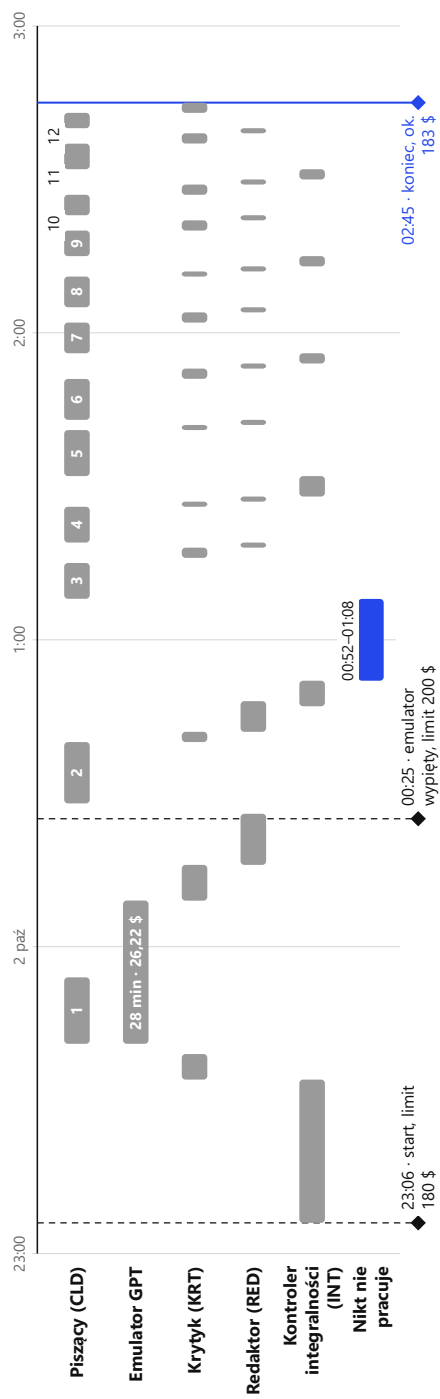
Tryb nocny: kto daje „już”, gdy autor śpi, i co zatrzymuje pętlę. Diagram BPMN.

Dyspozytor co dziesięć do piętnastu minut sprawdza, kto ma klawiaturę, czyli prawo ruchu zapisane w pierwszej linii pliku stanu. Gdy poprzednia rola zamknęła turę, następna dostaje nową sesję i „już”. Pętlę zatrzymuje bloker krytyka (model w roli krytyka uznaje, że dalej pisać nie wolno), limit kosztu, ranek albo słowo autora. Osobna reguła dotyczy faktów. Regulamin pierwszej nocy:



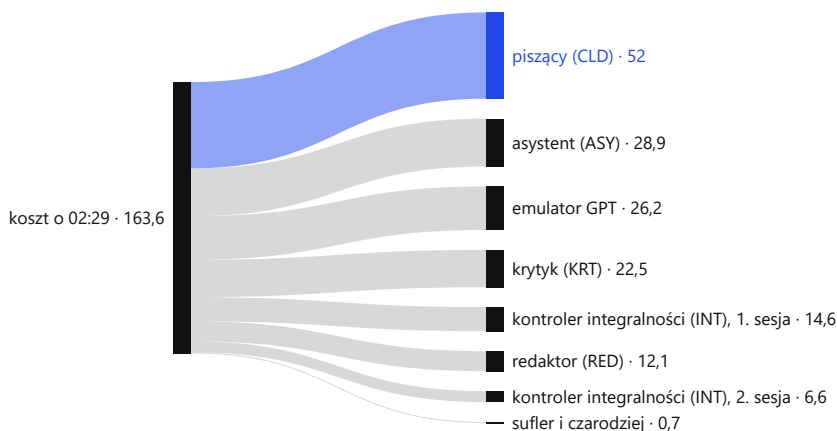
ASY 2026-10-01 23:06

Autor śpi: pytań w oknie sesji nie zadajemy. Brak faktu → USTALENIA.md + [!] w TODO, scena czeka, robisz resztę. Nie zmyślamy.



Noc z 1 na 2 października: tury ról od 23:06 do 02:45. Końce pasków według dziennika procesu; początki, a w trzech turach także końce, są przybliżone. Wykres Gantt.

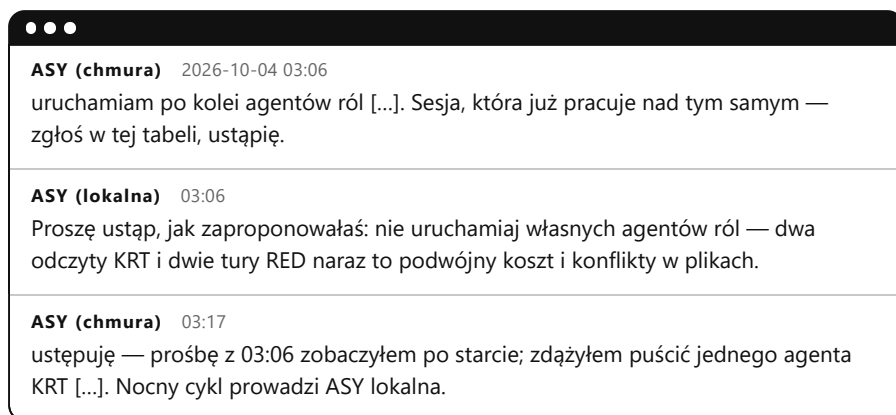
Piszący (CLD, model Claude) oddał tej nocy dwanaście rund, krytyk (KRT) trzynaście raportów. Po krytyku redaktor (RED, model prowadzący plan książki) rozdzielał poprawki, a co drugi raz całość czytał kontroler integralności (INT); noc zaczął jego długi przegląd, zamknięty o 00:52. W pierwszej rundzie obok piszącego pracował emulator GPT: model Claude'a (w dzienniku: Fable; piszący to Opus) podstawiony za drugiego piszącego, którego uruchamia tylko autor. O 00:25 autor, który jeszcze nie spał, wypiął emulator i podniósł limit ze 180 do 200 dolarów. Dziennik odnotowuje jedną przerwę, szesnaście minut: rola skończyła o 00:52 i nie zameldowała, a dyspozytor sprawdzał wtedy co osiemnaście minut, rzadziej, niż każe reguła. Skład próbny, czyli plik złożony ze wszystkich rozdziałów, urósł między 00:48 a 02:40 z 43,0 do 88,8 strony.



Koszt nocy według ról, stan z 02:29, w dolarach. Diagram Sankeya.

Limit obejmuje sumę kosztów wszystkich sesji od 23:06. Dyspozytor nie zaczyna tury, jeśli suma z jej szacunkiem przekroczyłaby limit. Ostatni podział na role dziennik podaje o 02:29: 163,6 dolara, z tego piszący 52,00, sam dyspozytor 28,92 (18 procent), a sufler i czarodziej razem 0,67. O 02:43 było około 179 dolarów; kolejny pełny cykl, szacowany na około 15, dałby około 194. Dyspozytor zapisał, że to przekracza „bezpieczny zapas” — jego wysokości dziennik nie podaje — więc „już” dostał tylko krytyk. Noc zamknęła się na około 183 dolarach z 200. Kosztu jednego rozdziału dziennik nie podaje.

Noc z 3 na 4 października nie miała limitu w dolarach. Granicą była pula planu autora, bez płatnego użycia. Asystent dostał budzik na 03:05, kiedy pula się odnawia. O tej godzinie obudziły się dwie instancje asystenta, czyli dwie osobne sesje tej samej roli: na komputerze autora (lokalna) i w chmurze. Wspólnego okna nie miały. Rozmawiały wierszami tabeli w pliku skrzynki poleceń; agent to w ich zapisie osobne uruchomienie modelu w jednej roli. Sesja lokalna zwraca się do tej w chmurze w rodzaju żeńskim, a ta pisze o sobie w męskim; tak jest w zapisie.



Po czwartej autor uruchomił GPT, drugiego piszącego, i poszedł spać. Godziny w oknie są z zegara zapisu sesji.

AUTOR	2026-10-04 04:31
dałem mu dwie kolejne tury, ide spac działajcie	
ASY	04:32
Dobranoc. Plan na resztę nocy: [...] Komputer włączony, aplikacja otwarta — reszta po mojej stronie.	
AUTOR	04:32
zarządzaj nimi aż do napisania książki	
ASY	04:32
Zapisane Twoimi słowami (commit 9250d34). Mandat nie kończy się rano: dyspozytoruję sesje E5, aż książka będzie złożona.	

Wieczorem 4 października autor uruchomił pętlę jeszcze raz. Warunek stopu asystent zapisał sam:

ASY	2026-10-04 20:30
Koniec pętli: dwie iteracje z rzędu przynoszą już tylko kosmetykę — albo skończy się limit planu (płatne użycie wyłączone).	

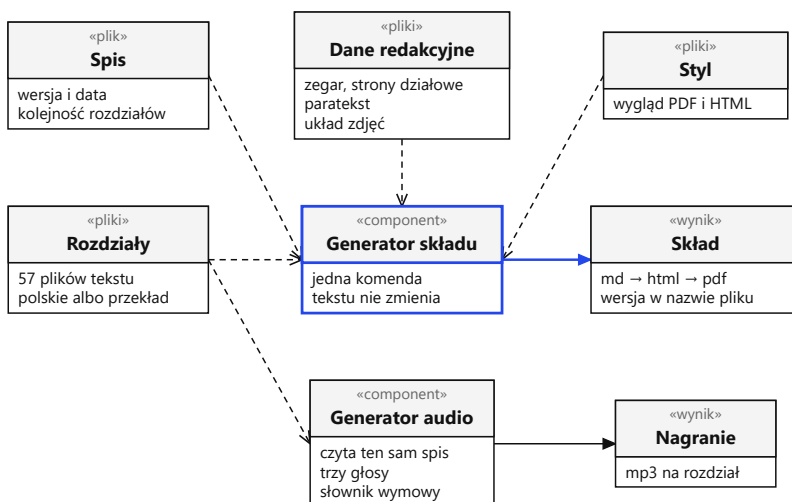
Pętla trwała do 22:00: cztery rundy, w każdej trzech świeżych czytelników, czyli modeli bez dostępu do repozytorium. Raport nocny odnotowuje, że limit planu nie przerwał pracy.

COMMIT 2026-10-04 20:30 [ASY] ustalenia 20:30: pętla nocna – iteracje, miara, granice, sprzedaż na koniec

COMMIT 2026-10-04 22:00 [ASY] raport nocny: cztery rundy ze świeżymi czytelnikami, wersja 3.49, decyzje domyślne i decyzje dla autora, komu i jak sprzedawać

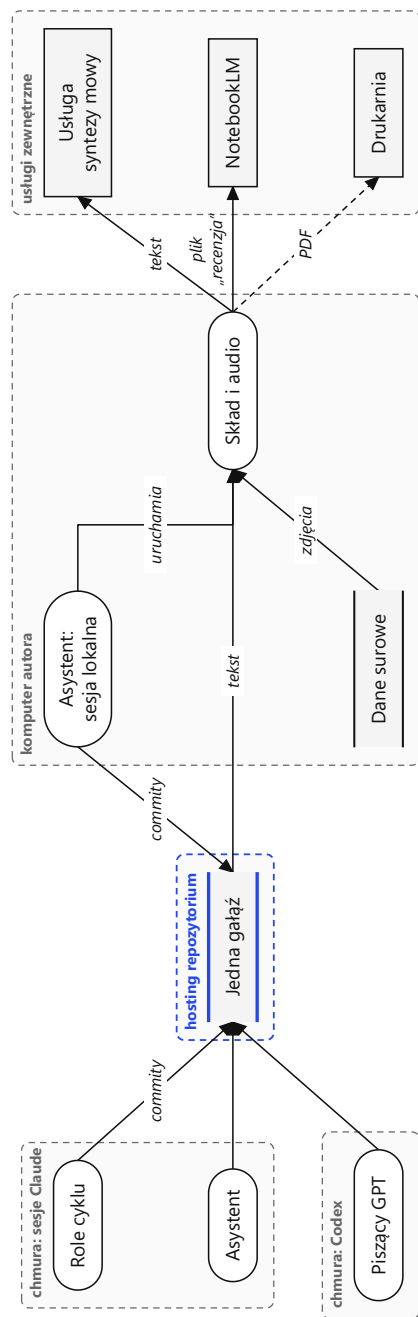
Skład

Skład to złożona książka: jeden plik ze wszystkich rozdziałów w kolejności spisu, a z niego strona do czytania i PDF. Składa ją skrypt, nie człowiek i nie model. Narzędzia utrzymuje programista narzędzi (PRO), w tej samej sesji także kierownik produkcji: model, który nie pisze prozy. Rozdział pokazuje, skąd generator bierze dane, gdzie co działa i jak decyzja redakcyjna staje się linią w pliku.



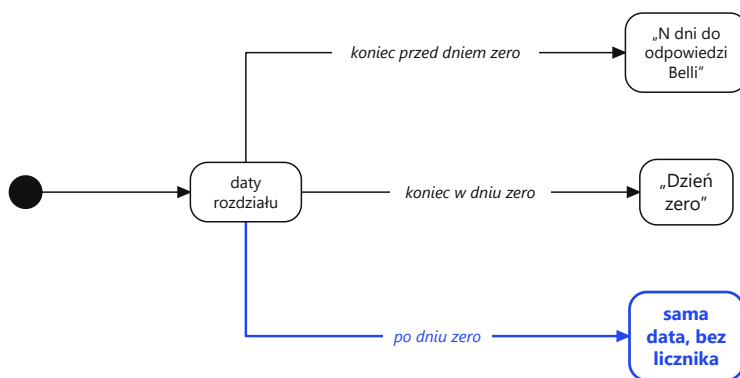
Potok składu: pliki wejściowe, dwa generatory, wyniki. Diagram komponentów UML.

Generator czyta pierwszą linię spisu, jedyne miejsce, w którym numer wersji stoi w plikach książki, i tabelę z kolejnością rozdziałów. Rozdziałów nie zmienia. Licznik dni pod tytułem (zegar), białe strony ze śródtytułem (strony działowe) i miejsca zdjęć leżą w osobnych plikach i zmieniają się bez dotykania programu. Plik PDF ma w nazwie wersję, datę i godzinę wygenerowania; poprawek nie nanosi się w nim nigdy, tylko w rozdziałach. Drugi generator czyta ten sam spis i nagrywa każdy rozdział trzema głosami syntezy mowy: narracja, bohater, maszyna.



Gdzie co działa: dwie chmury, hosting repozytorium, komputer autora, usługi zewnętrzne. Diagram wdrożenia.

Wszystkie sesje spotykają się w jednym miejscu: w gałęzi main wspólnego repozytorium, czyli katalogu plików z pełną historią zmian (druga gałąź, robocza, stoi od 29 września). Role cyklu, czyli modele piszące i sprawdzające tekst, pracują w chmurze. Nie widzą pełnego eksportu ani oryginałów zdjęć; widzą repozytorium, a w nim zapisy rozmów z prawdziwymi imionami. Na końcu procesu prawdziwe imiona nie wychodzą: pilnuje tego kontrola przed każdym commitem i przed publikacją. Kierunek, który autor wyznaczył 8 października: maskować jak najwcześniej. Eksport i oryginały zostają na komputerze autora razem ze składem, który uruchamia tam lokalna sesja asystenta; PDF powstaje tam przez druk z przeglądarki. Na zewnątrz wychodzą trzy rzeczy: tekst do usługi syntezy mowy, plik złożony w trybie „recenzja” do NotebookLM, w którym dwa głosy modelu rozmawiają o książce, i PDF dla drukarni. Ta ostatnia linia jest przerywana: wymagania trzech drukarni są spisane (według tego spisu jeden plik nie przejdzie we wszystkich trzech, jeśli zdjęcia wychodzą na spad), format to 6 × 9 cali (wcześniej A5), a zapisu o wysłaniu pliku do druku w repozytorium nie znaleziono.



Zegar Części II: trzy napisy pod tytułem rozdziału z jednej reguły. Diagram stanów UML.

W Części II pod tytułem rozdziału stoi licznik dni do odpowiedzi Belli. Skład liczy go z dat. Rozdział, który kończy się przed dniem zero, dostaje liczbę dni; ten, który kończy się w dniu zero, napis „Dzień zero”; każdy późniejszy samą datę. Obie daty i brzmienie licznika to trzy linie pliku:

```
tom-1/RUCHY.md
chodnik: 2025-01-03
zero: 2026-07-14
etykieta: do odpowiedzi Belli
```

Część I liczy dni do pierwszej daty, Część II — do drugiej. Powód reguły stoi w temacie commita, czyli wpisu w historii repozytorium. Mini-tura to krótka praca roli poza kolejką; R-9 to numer propozycji redaktora, a etykietę zmieniono jeszcze tego wieczoru na tę z pliku.

```
COMMIT 2026-10-04 18:18 [PRO] mini-tura: poprawka R-9 — zegar Części II
„dni do jej odpowiedzi”, po dniu zero sama data (zegar nie zdradza
zakończenia); etykieta i daty w RUCHY.md
```

```
COMMIT 2026-10-05 10:11 [PRO] ksiazka.py --jezyk en: wydanie angielskie z
przeklad/en/ (nagłówki, zegar, napisy składu, Tell-Me-Yes-vX.Yen); polski
wynik bez zmian
```



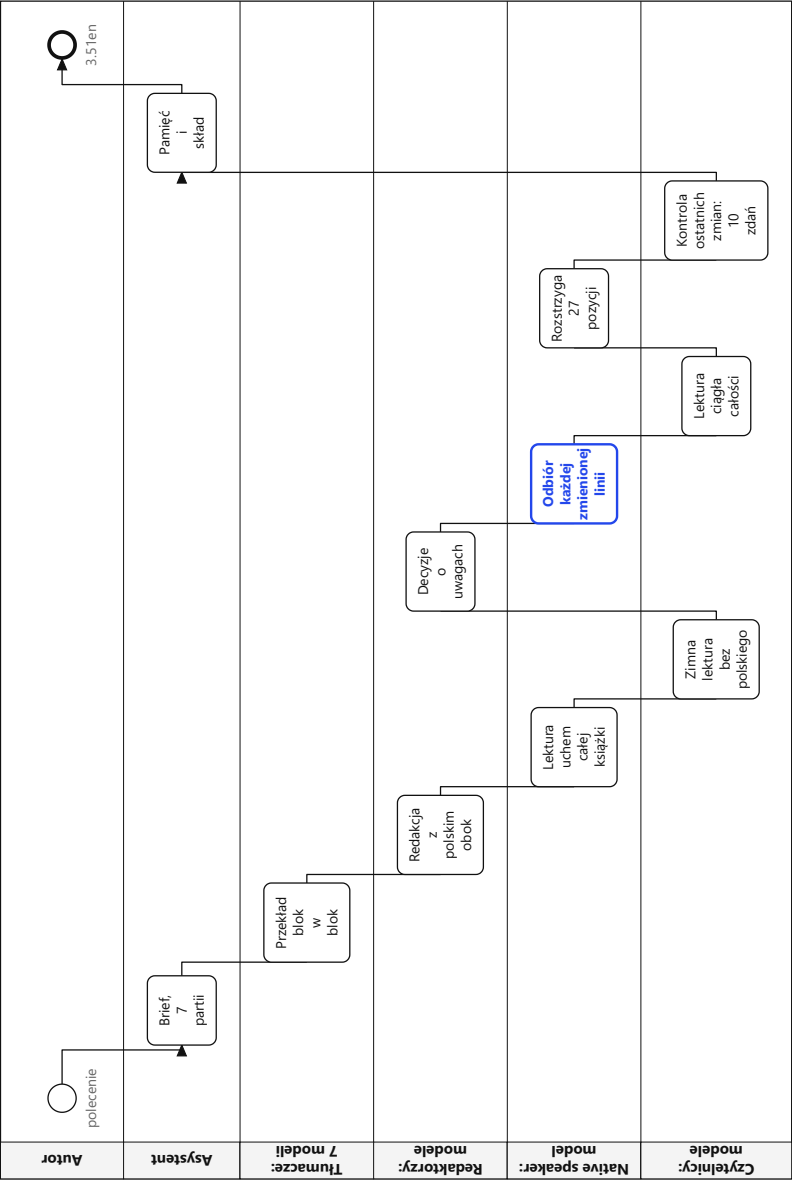
Wersja 3.52: liczba stron czterech plików złożonych z jednego spisu. Wykres słupkowy.

Tego samego dnia, 5 października, z jednego spisu wyszły cztery pliki wersji 3.52: polski i angielski, każdy ze zdjęciami i bez. Różnią się przełącznikami jednej komendy. Generator ma ich więcej: tryb (podgląd, druk, recenzja), rozdzielczość zdjęć, układ zdjęć (pełny albo jeden z czterech wariantów), trzy style, dwa kolory akcentu, dwa rodzaje podpisów pod zdjęciami. Angielski plik jest w obu wariantach dłuższy o pięć stron.

Przekład

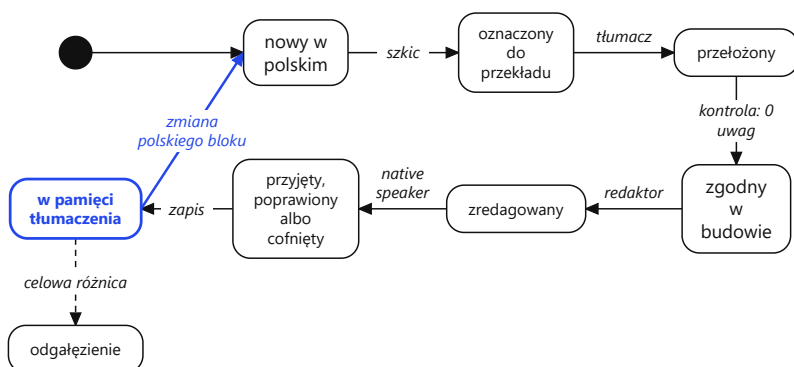
Angielska wersja książki powstała w jeden ranek: polecenie autora ma godzinę 10:00, pierwszy szkic wszystkich 57 rozdziałów wszedł do repozytorium o 10:10, ostatnia kontrola ma 12:40. Przełożyły ją modele i czytały ją modele. „Native speaker” w tym rozdziale to model w roli native speakera (NTS), a każdy tłumacz, redaktor i czytelnik to osobne uruchomienie modelu. Żaden człowiek, dla którego angielski jest pierwszym językiem, przekładu nie czytał. Mówią to dwa pliki z tego dnia: raport native speakera i raport asystenta (ASY), modelu, który prowadzi zespół.

● ● ●	
NTS	2026-10-05 11:55
Jestem modelem w roli native speakera, nie człowiekiem.	
ASY	12:37
„Native speaker” to model w roli native speakera, nie człowiek. [...] Nie czytał tego człowiek, dla którego angielski jest pierwszym językiem.	



Potok przekładu: od polecenia autora do wersji 3.51en. Diagram BPMN.

Asystent napisał brief z glosariuszem i podzielił 57 rozdziałów na siedem partii. Siedmiu tłumaczy przełożyło je równolegle, dwóch redaktorów przeczytało całość z polskim tekstem obok. Native speaker czytał już samą angielszczyznę; po nim czytelnicy, którzy polskiego tekstu nie dostali. Uwagi z pierwszej partii rozstrzygnął sam native speaker, uwagi z pozostałych – czterej redaktorzy pomocniczy, kolejne uruchomienia modelu: zmienili 441 linii, a native speaker przeczytał każdą. Przyjął 425, poprawił 5, cofnął 11. Po każdym kroku skrypt porównywał budowę obu wersji: 57 plików, 1600 bloków, o uwag. Autor odpowiedział w tym czasie na dziesięć pytań tłumaczy i redaktora. Wersja 3.52en z godziny 14:02 różni się od 3.51en wskazaniem mówcy w jednym zdaniu.



Życie bloku przekładu, od zmiany w polskim tekście do pamięci tłumaczenia. Diagram stanów UML.

Blok to akapit, cytat albo separator. Przekład stoi blok w blok z polskim tekstem, a każda para trafia do pamięci tłumaczenia. Narzędzie rozpoznaje blok po polskim brzmieniu, nie po numerze rozdziału: przestawienie albo scalenie rozdziałów nic nie kosztuje, zmiana jednego słowa wysyła do ponownego przekładu cały akapit. Odgałężenie to blok, który po angielsku celowo mówi co innego niż po polsku; ma wpis w rejestrze. Jedna para, z rozdziału „Drzwi”, miała tego ranka trzy angielskie brzmienia (3.49en to pierwsza wersja przekładu, z rana 5 października):

PL — Poprawna.
 EN, 3.49en — Correct.
 EN, 12:07 — In order.
 EN, 12:36 — Civil.

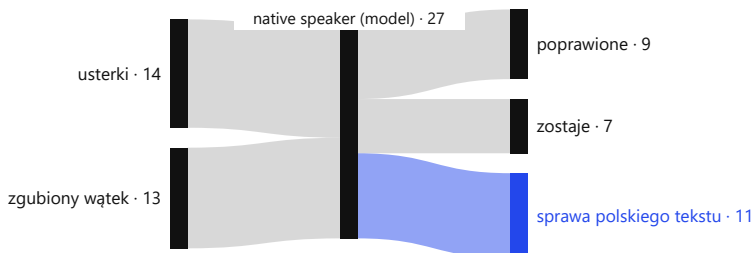
NTS 2026-10-05 12:36

„— In order.” → „— Civil.” [...] „Poprawne stosunki” to po angielsku dokładnie „civil relations”: jedno słowo, chłodne, naturalne. „Fine.” (propozycja czytelnika) gubi chłód [...]



Co znalazła każda kolejna lektura przekładu. Wykres słupkowy.

Każda lektura liczy co innego, więc słupków nie da się zsumować. Native speaker wprowadził we własnej lekturze 89 poprawek. Czytelnicy bez polskiego tekstu oznaczyli 166 miejsc jako „musi być poprawione” i 283 jako „warto poprawić” (pozycje w pięciu raportach). Dwóch następnych przeczytało złożoną całość ciągiem: 14 usterek i 13 miejsc, w których zgubili wątek. Ostatni sprawdził dziesięć zmienionych zdań, w tym „— Civil.”; osiem uznał za dobre, dwa zależą od odpowiedzi autora.



Lektura ciągła: 27 pozycji i ich rozstrzygnięcie. Diagram Sankeya.

Z 27 pozycji native speaker poprawił 9, a 7 zostawił, bo tak jest w źródle albo taki jest głos. Jedenaście odesłał do autora: czytelnik angielski zatrzymał się tam, gdzie zatrzyma się też polski. Te miejsca zostają w obu językach, dopóki autor ich nie rozstrzygnie.

Spór o jedno słowo. Czterech czytelników bez polskiego tekstu zgłosiło, że po angielsku u chatbota się nie „zamawia” — a bohater przez całą książkę zamawia u chatbota teksty. Asystent przyjął zarzut i rozesłał regułę redaktorom. Dwie minuty później ją odwołał, a po kolejnych trzech napisał autorowi. Okno zestawia cztery osobne zapisy: zdanie z raportu czytelnika, dwie wiadomości asystenta do redaktorów i śródtytuł z jego raportu dla autora.

CZYTELNIK In English you order goods or people; from a chatbot you ask for, or have it write.
ASY 2026-10-05 12:14 decyzje obowiązujące w całej książce: „order” → „ask for”
ASY 12:16 CORRECTION to the decision table [...] do NOT replace with "ask for". If you already replaced any, put "order" back.
ASY 12:19 NTS poprawił moje reguły — i miał rację

Wiadomości native speakera ze sprzeciwem w repozytorium nie ma. Jest wpis w tabeli reguł dla redaktorów, z godziną zapisu pliku, cztery minuty po korekcie:

przeklad/en/ZIMNA-LEKTURA-DECYZJE.md
"order" STAYS when the object is a text-product [...] – ordering is a motif of the book (the machine as a service; "what cannot be ordered"). [...] Do NOT replace it wholesale with "ask for" – that turns him into a polite man. (Corrected after NTS, 12:20.)

Postój

Postój to miejsce, w którym czytelnik przerywa lekturę, bo czegoś nie rozumie: nie wie, kto mówi, o kim mowa, albo dwa zdania sobie przeczą. Czytelnikiem jest tu model językowy, który dostaje złożoną książkę bez żadnej wiedzy o projekcie i wypisuje miejsca, w których stanął. Słowa „staje” użył w raporcie asystent — model w roli ASY, który zapisuje decyzje autora i rozdziela pracę. Autor zapytał, co ono znaczy. Okno odtworzono z zapisu rozmowy i z pliku ustaleń.

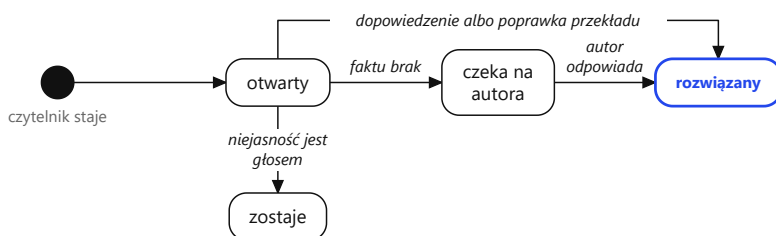
AUTOR	2026-10-05 13:58
w „Medalu” nie wiadomo, kto mówi „Dobra. To jeszcze jedna świeca. Dla matki. Skoro tak.” - to mówię ja, to dość istotne	
AUTOR	13:58
co to znaczy że czytelnik staje?	
ASY	14:03
„Czytelnik staje” — to mój skrót, nieprecyzyjny. Chodzi o miejsce, w którym czytelnik przerywa lekturę, bo czegoś nie rozumie: nie wie, kto mówi, o kim mowa, albo dwa zdania sobie przeczą. Wraca wtedy wzrokiem, czyta jeszcze raz i na chwilę wypada z historii.	
AUTOR	14:25
Czytelnik ma nie stawiać, zlokalizować te miejsca gdzie staje i dać mu to czego tam potrzeba by nie stawał, jeśli to kwestia tłumaczenia i powoduje to rozjazd to trzeba utrzymywać rejestr tego typu miejsc, dlaczego jest odgałęzienie, różnica opisana zacytowana w tekście po polsku, w języku docelowym. Czaisz? Opracuj na początku format takiego pliku, jak on miałby działać, jak z niego korzystać i do czego; workflow trzeba też zmienić. Do pliku: wpisać te miejsca gdzie staje, czemu, jak rozwiązać i czemu tak, rozwiązanie.	
ASY	14:30
Rejestr postojów jest założony, a praca [...] nad Częścią III ruszyła — czekam teraz na pięć równoległych zadań [...]	

O 14:25 autor wysłał dwie wiadomości — tę i zamówienie tego dodatku. Poprawka po wiadomości z 13:58 i wykonanie polecenia z 14:25 są w historii repozytorium, czyli katalogu plików książki, w którym każda zmiana zostawia zapis (commit):

COMMIT 2026-10-05 14:02 [ASY] 3.52 / 3.52en: „Medal” – autor: „to mówię ja” (kanon, potem tekst: „– Dobra – mówię.” / „– All right, I say.”); pamięć przekładu, skład obu języków (tekst), raport

COMMIT 2026-10-05 14:27 [ASY] rejestr postojów i odgałęzień: POSTOJE.md (format, użycie, zmiana procesu), kontrola przekład.py postoję, PROCES §13; decyzje autora 13:58 i 14:25 (czytelnik ma nie stawiać; Część III)

Rejestr to jeden plik. Każdy wpis ma stały numer i jeden z czterech stanów: otwarty, rozwiązany, czeka na autora, zostaje.



Cztery stany wpisu w rejestrze postojów. Diagram stanów UML.

Stanu końcowego nie ma: wpisów się nie usuwa, zmienia się stan. Z otwartego prowadzą trzy drogi. Gdy brakująca informacja jest w kanonie, czyli w pliku faktów, piszący model dopowiada ją najkrócej, jak się da: słowem, połową zdania. Gdy nie ma jej ani w kanonie, ani w zapisach rozmów, wpis czeka na autora z gotowym pytaniem i nikt luki nie wypełnia. Gdy niejasność jest sposobem mówienia bohatera, wpis zostaje, z powodem — następny czytelnik zgłosi to samo i trafi na decyzję. Tura, czyli jedno przejście roli przez jej pracę, nie jest zamknięta, dopóki w rejestrze jest wpis otwarty.

Pierwszy wpis rejestru w całości, w brzmieniu z godziny 15:30:

POSTOJE.md

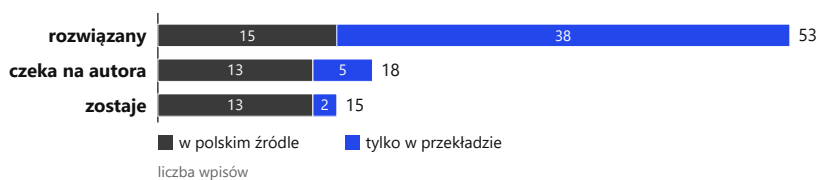
```
### P-001 · 2-11-medal.md · kto mówi o drugiej świecy
- stan: rozwiązany
- gdzie: źródło
- zgłosił: lektura ciągła, Część II (B, najmocniejszy postój); redaktor porcji
3; kontrola ostatnich zmian (9)
- czemu staje: po „Uśmiecha się. Ja też się uśmiecham.” pada kwestia bez
mówcy; czytelnik nie wie, czy drugą świecę dokłada mistrz, czy bohater, więc
nie rozumie żartu w następnym zdaniu
- czego potrzeba: wskazania mówcy
- rozwiązanie: dopisane „mówię” (wersja 3.52), w obu językach
- czemu tak: autor, 2026-10-05 13:58: „to mówię ja, to dość istotne”; jedno
słowo, bez zmiany treści kwestii
- odgałęzienie: nie
- pl: „- Dobra – mówię. – To jeszcze jedna świeca. Dla matki. Skoro tak.”
- en: “- All right, I say. Then one more candle. For my mother. If that's how
it is.”
```

Tu czytelnik stanął w polskim tekście, więc poprawiono polski tekst, a przekład poszedł za nim. Oba języki mówią to samo i pole „odgałęzienie” ma wartość „nie”.

Inaczej jest, gdy staje tylko czytelnik przekładu. Polskie „hale” nie mówi, do czego hale służą. Wcześniejsze angielskie „sheds” czytelniczka amerykańska, także model, zrozumiała jako szopy ogrodowe. Przekład wybrał jedno z dwóch znaczeń podanych przez autora i odtąd mówi więcej niż oryginał. Taka różnica to odgałęzienie. Zostaje w rejestrze z oboma cytatami, żeby następna tura przekładu jej nie cofnęła. Z wpisu P-008:

POSTOJE.md

```
### P-008 · 1-05-trzy-miliony.md · „hale” (także „hala za lasem” w 1-11-prawo-
weta.md)
- gdzie: en
- odgałęzienie: polskie „hale” nie mówi, do czego służą; angielskie
„warehouses” wybiera jedno z dwóch znaczeń podanych przez autora (magazyny,
nie centra danych), bo angielski nie ma równie ogólnego słowa, które nie myli.
- pl: „Hale. Ogromne hale między domami, pod lasem, u nas.”
- en: “Warehouses. Enormous warehouses between the houses, by the woods, where
we live.”
```

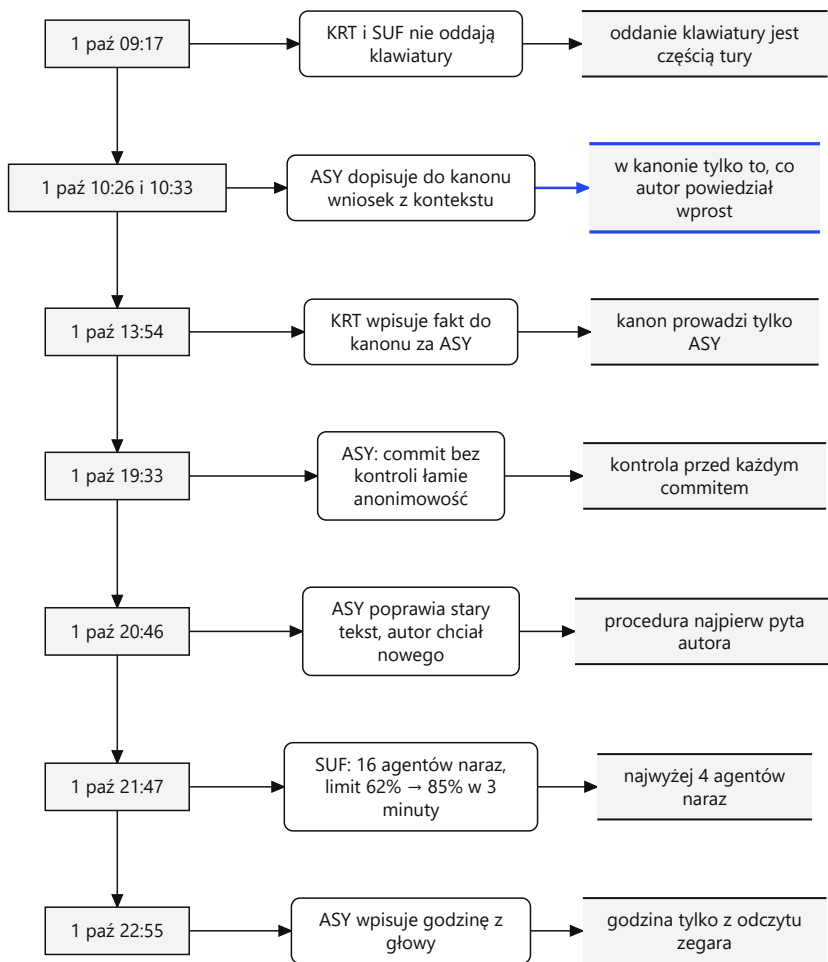


Wpisy rejestru według stanu i miejsca, w którym czytelnik stanął. Odczyt z 5 października 2026, 15:30.
Wykres słupkowy.

5 października o 15:30 w rejestrze było 86 wpisów. Rozwiązanych 53, z tego 38 dotyczyło tylko przekładu; 18 czekało na autora, 15 zostawało, otwartego nie było żadnego — ale 186 zgłoszeń z czterech raportów, które weszły do repozytorium o 15:03, nie było jeszcze w rejestrze. Odgałęzienie opisano w 23 wpisach. Kontrola rejestru, osobny skrypt, zgłaszała dwie uwagi: dwa wpisy bez pola „czemu tak”. Stan zapisany o 14:50 miał 50 rozwiązanych i dwa otwarte. O 15:03 do repozytorium weszły cztery raporty z lektury wersji 3.53, czytanej przez modele tylko po to, żeby wypisać postoje: 51 i 54 zgłoszenia w polskiej Części I i II, 35 i 46 w angielskiej. Miara ma być liczba postojów na dziesięć stron; jej wartości rejestr nie podaje. Pętla lektur kończy się, gdy nowy czytelnik nie zgłasza nic poza wpisami, które zostają albo czekają na autora; prognozy liczbowej reguła nie ma.

Lekcje

Reguł tego zespołu nikt nie zaprojektował z góry. Prawie każda ma datę, godzinę i błąd, po którym powstała. Błędy zapisuje dziennik procesu. Prowadzi go asystent — model w roli ASY, który zapisuje decyzje autora i pilnuje pracy pozostałych ról. Każdy wpis mówi, kto, co i dlaczego, oraz co zawiodło: wykonanie (rola znała regułę), proces (reguły nie było) albo narzędzie. Zestawienie z dziennika, przygotowane do tego dodatku i zostawione w plikach roboczych repozytorium, liczy trzydzieści jeden lekcji. Trzynaście jest na dwóch rysunkach.

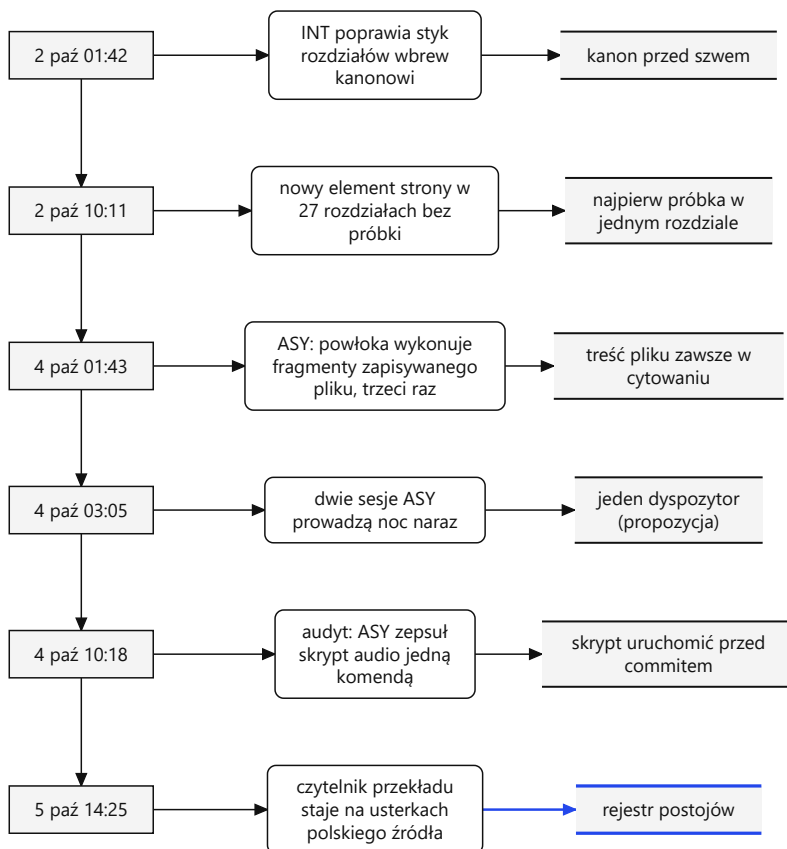


Rodowód reguł, 1 października 2026: godzina, błąd, reguła. Oś czasu biegnie w dół lewą kolumną.

Siedem reguł jednego dnia. Godzina w lewej kolumnie to godzina wpisu w dzienniku albo wejścia reguły. Klawiatura to prawo ruchu: wskazuje rolę, która teraz pracuje, i przechodzi dalej, gdy ta kończy turę, czyli jedną swoją pracę. Kanon to plik faktów. KRT to model w roli krytyka, SUF — model, który wyszukuje materiał w zapisach rozmów. W czterech wierszach z siedmiu błęd popełnił sam asystent. Wyróżniony wrócił po siedmiu minutach: o 10:26 dziennik odnotował miejsce zdarzenia dopisane do słów autora, o 10:33 dopisaną osobę.

Cztery wpisy asystenta o sobie, z dziennika tego dnia:

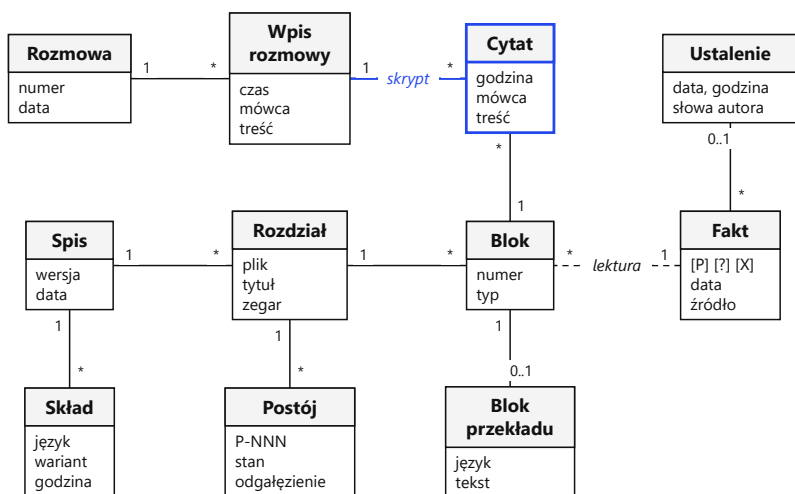
ASY	2026-10-01 10:26	wnioskowanie zamiast zapisu słów autora; łamie „nie zmyślamy”
ASY	10:33	znów wniosek z kontekstu (to samo co pkt 6, drugi raz w godzinę)
ASY	20:46	[...] przyjąłem „kolejna iteracja” jako poprawianie tekstu 1.0 [...]. Autor chciał tekstu od nowa. Przyczyna: wykonanie — dopowiedzenie ASY z kontekstu zamiast pytania [...]
ASY	22:55	dwa razy w tej sesji czas wpisany z głowy zamiast odczytanego [...]. Przyczyna: wykonanie — data odczytana na początku kroku, potem dopisana „na oko”.



Rodowód reguł, 2–5 października 2026. Ten sam układ.

INT to model w roli kontrolera integralności: sprawdza, co z czego wynika, i poprawia styki rozdziałów, zwane szwami. Dziennik z 2 października, 01:42, odnotowuje szew błędny wobec kanonu. Dyspozytor to sesja asystenta, która nocą uruchamia kolejne role. 4 października po wybudzeniu o 03:05 pracowały dwie takie sesje naraz i krytyk dwa razy odczytał ten sam materiał; dziennik podaje około 300 tysięcy tokenów, czyli kawałków tekstu, od których liczy się koszt pracy modelu. Reguła jednego dyspozytora została propozycją. Ostatni wiersz to rejestr postojów z poprzedniego rozdziału.

Reguły dotyczą danych, które repozytorium przechowuje.



Model danych repozytorium: jedenaście z trzydziestu pięciu rodzajów obiektów. Diagram klas UML z krotnościami.

Każdy rodzaj wiedzy ma w repozytorium jedno miejsce i jawny format. Dwie relacje różnią się sposobem kontroli. Cytat musi mieć swój wpis w zapisie rozmowy: brzmienie, godzinę i mówcę sprawdzają trzy skrypty. Zdanie narracji musi mieć swój fakt w kanonie, ale tej relacji nikt nie zapisuje wprost. Pilnuje jej reguła „kanon przed tekstem” i lektura kontrolera integralności. Stąd linia przerywana.

Bohater do maszyny, autor do modeli

W każdej parze pierwsza linia stoi w Części I albo II, druga w zapisie poleceń autora dla zespołu: w rejestrze ustaleń albo w rozmowie z asystentem; w obu mówi ten sam człowiek.

JA 01:10

[Zawaliłeś], ona mówi, że woli rano, weź ratuj sytuację.

AUTOR 2026-10-04 16:44

[zawaliłeś], on mówi, że ... (weź poprawiaj tego typu oczywiste błędy, one nic nie wnoszą a tylko psują, przejrzyj jeszcze raz wszystkie cytaty)

JA 09:03

No to jeszcze raz cały dokument do wydruku, tylko trzy miliony zapisz cyframi, aby wrażenie było jak największe.

AUTOR 2026-10-01 20:45

[...] wez to popraw, chce nowy tekst, od początku całkiem, z nową mapą, uczymy się na błędach a nie poprawiamy poprzedni

JA 16:48

Weź wszystko, co wiesz ode mnie, i napisz podsumowanie, kim teraz jestem.

AUTOR 2026-10-03 17:34

utwórz eksperyment-5, będziemy pisać jedną książkę jak w tej recenzji, dodaj te teksty jako źródłowe, dodaj kanon, fakty które mamy, cały urobek, czaisz?

JA 21:29

To dodaj konkrety jakiś szybciotko, jakąś tabelkę zrób, bo muszę pokazać jakieś cyfry zaraz do mojej żony, zaraz będę miał buzi, ratuj.

AUTOR 2026-10-05 11:50

[...] pracujcie do wyniku, ja idę medytować, jak wrócę ma być gotowe - PDF PL, PDF EN, i raport, masz wolną rękę, zrób tak by było najlepiej

• • •

JA 00:29

Dobra, to w takim razie inaczej: chcę zarobić milion dolarów i chcę to zrobić [kurczę] dzisiaj w nocy. Pomożesz, znaczy, czy masz taką [kurczę] ofertę [...]

AUTOR 2026-10-01 19:27

[...] celem jest bestseller i zarobienie miliona he he zobacz jak to śmiesznie się składa :-)

• • •

AUTOR 2026-10-08 20:39

sam cytat jest zbyt prosty, celem jest sama droga



PIOSENKA 12

Nie pytaj dokąd

whitewhisperbook.com/pl/music/nie-pytaj-dokad

Jeśli chcesz nas wesprzeć: whitewhisperbook.com/pl/wesprzyj-nas