

How This Book Was Made

An appendix to the book “Tell Me “Yes””

BIAŁY BRZEG

How This Book Was Made. An appendix to the book "Tell Me "Yes""

Biały Brzeg

The book was made in the White Whisper engine: whitewhisperbook.com

bialybrzeg.com/en

Version 4.7en · 18 October 2026

Polish text: version 4.7

Contents

APPENDIX · HOW THIS BOOK WAS MADE.	5
The Order	7
Five Attempts	11
The Team	16
The Turn	21
The Fact	27
The Quotation	32
The Name	37
Who Wrote It	42
The Critic and the Readers	47
The Night	52
The Build	59
The Translation	65
The Stop	71
Lessons	76

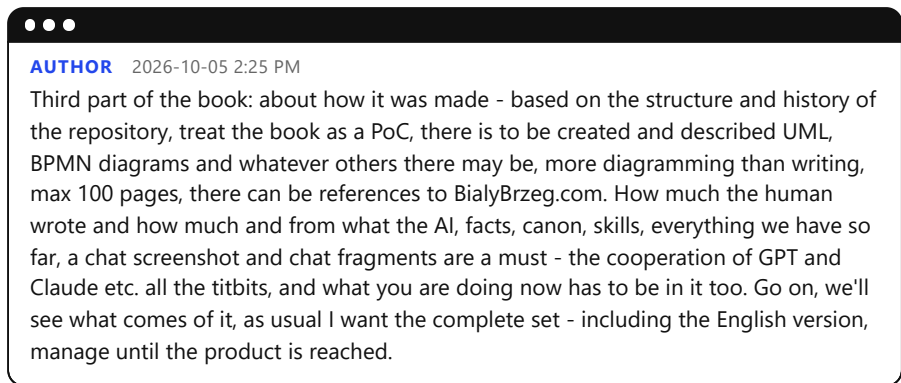
APPENDIX

How this book was made.

Tell Me “Yes” is non-fiction: the record of one man's conversations with a chatbot, with timestamps. The author is its hero – in this appendix “the hero” is the author as he appears in the book, and “the machine” is the chatbot he talks to. That chatbot is ChatGPT; in the team described below, one of the two writers is a model from the same company. This is an appendix to the book; its two parts are not here. The book opens with a scene on the pavement, then goes back 947 days and returns to it: Part I counts the days to the pavement, Part II to Bella's answer. In Parts I and II that counter appears under the title of every chapter; in the Appendix there is none. The names of the people, and some of the details and dates, are deliberately changed in the book: “Bella” – that is what the hero's wife is called in the book – and “the master” are names from the book. In the chat windows “[...]” marks a cut, and a word in square brackets replaces a word from the record: a name or, in this appendix, a swear word.

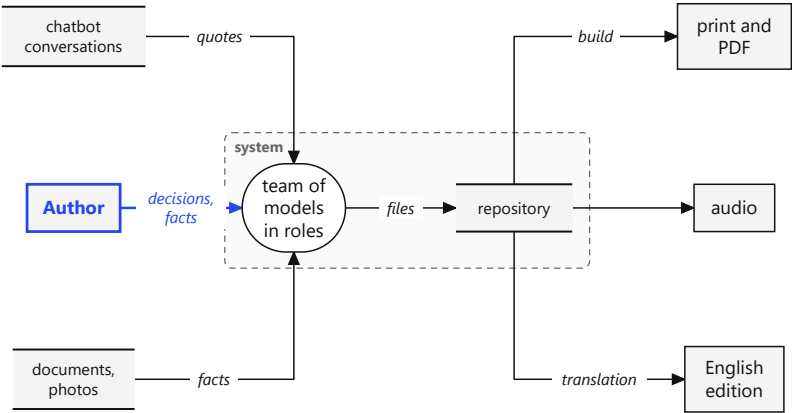
The Order

This appendix was written by language models on a single instruction from the author. In the instruction and in the records from those days it is still called the third part of the book; it stands at the end of the book, after everything else. Here is the instruction.



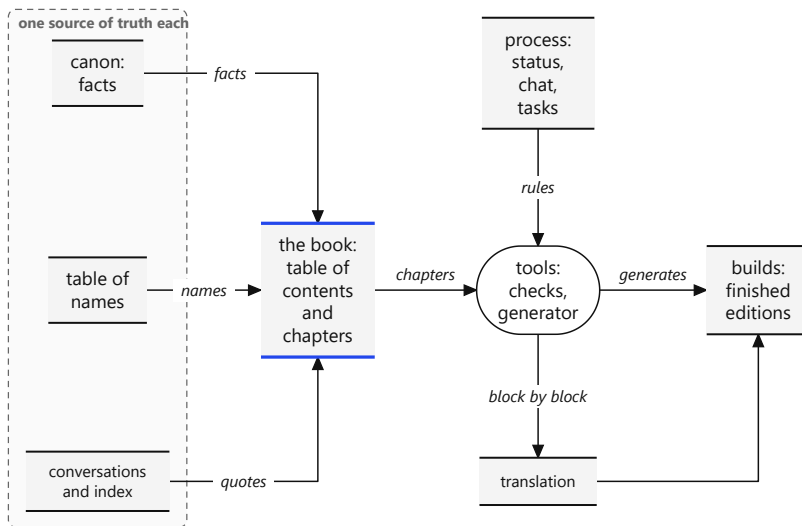
PoC is short for “proof of concept”: a trial run that is meant to show whether an idea can be carried out. UML and BPMN are two notations for drawing diagrams. A skill is a procedure saved in a file, which a model carries out step by step. The instruction was picked up by the assistant (ASY), the model that runs the team and talks to the author – a Claude model. The book is kept in a repository, that is, in a directory of files with a full history of changes. One entry in that history is called a commit; it has a timestamp and a subject, and the subject begins with the abbreviation of the role. At 14:27 the assistant added such an entry – one for the two orders the author sent at 14:25: about this appendix and about the stops. A stop (“postój” in Polish, hence the file name POSTOJE.md), in the team's language, is a place where the reader breaks off reading because they do not understand something; a divergence is a place where the translation deliberately departs from the Polish text.

COMMIT 2026-10-05 2:27 PM [ASY] register of stops and divergences: POSTOJE.md (format, use, change of process) [...]; the author's decisions of 13:58 and 14:25 (the reader must not stop; Part III)



The system in one picture: what goes into it and what comes out of it. Context diagram.

Three things go into the system: the logs of the conversations with the chatbot, known from Parts I and II; documents and photographs; and the author himself. The author does not write to the files. He dictates decisions and facts, and gives the go-ahead for work to start with the word “go”. The team consists of separate sessions of language models, each in one role. The sessions do not talk to each other directly; they communicate through files in the repository. Out of the repository come editions: a print file, a recording read by text-to-speech voices and the English translation.

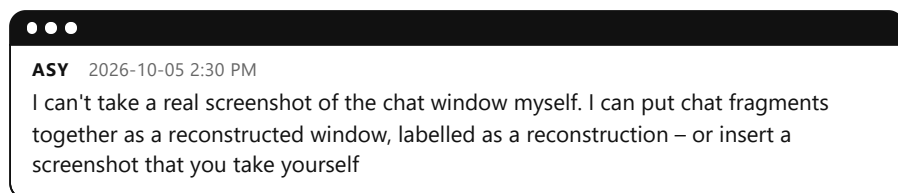


What is in the repository and where a chapter gets its facts, names and quotes. Container diagram.

Every kind of knowledge has one place. Facts about events and people are held in one file, called the canon. The names under which people appear in the book are held in one table. Quotes come from the conversation logs. Alongside the text is the record of the work: whose move it is now, what the roles have handed over to each other, what is left to do. The tools are scripts. They check the chapters and build the finished edition out of them; the resulting file is called the build here.

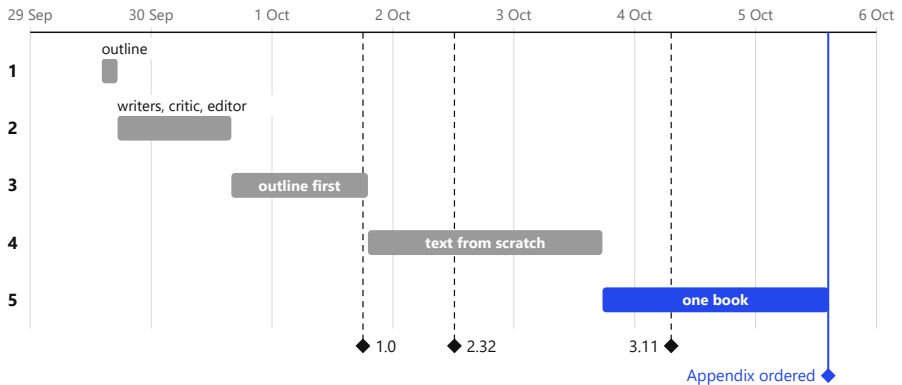
The author keeps a public diary under a pseudonym at bialybrzeg.com – the order allowed references to it. On 5 October 2026 the home page carried the sentences: “A book was made from this record.” and “A new entry on Mondays, Wednesdays and Fridays.”

The author ordered a chat screenshot. The chat windows in this appendix are reconstructions: every line is copied word for word from a file or from the record of a session, and “[...]” marks a cut; a word in square brackets replaces a name or a swear word from the record. Five minutes after the instruction the assistant wrote to the author:



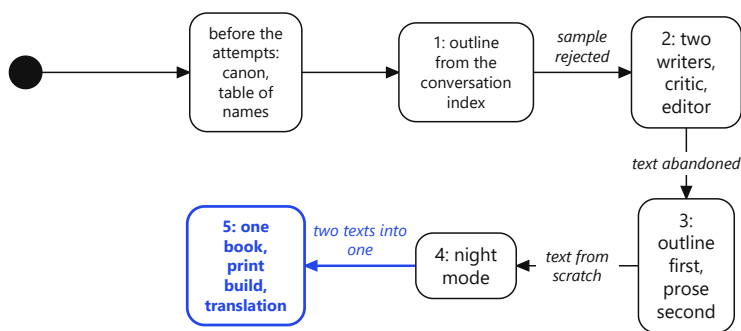
Five Attempts

The book was made the way a proof of concept is made in computing: an attempt, an assessment, the next attempt. The repository was set up on 20 August 2026: the canon of facts, the text and the checking script. By 14:27 on 5 October it held 983 commits, that is, entries in the history of changes; all the figures in this chapter are as of that time. All the commits fall on eight days. Between the first and the second of those days there are 39 days without a single entry.



Five attempts at the book from 29 September to 5 October 2026. Timeline; the diamonds mark built versions and the order for the Appendix.

From 29 September on, the team started the book five times. Each attempt, called an experiment in the repository, has its own directory; the earlier ones stay as an archive. The first lasted about three hours, the fifth is still going. The author listened to version 1.0, the result of the third attempt, on 1 October at 18:03 and an hour later the fourth attempt began. Version 2.32 of 2 October is the result of the fourth. Version 3.11 of 4 October, 07:15, is the complete working draft of the fifth, built after a night of writing.



What each attempt added to the way of working and what happened to the text of the previous one. UML state diagram.

Each new attempt took over the rules, the tools and the canon, that is, the file of facts, from the one before. It did not take over the prose. The fifth is the exception: a review saved in the repository among the author's notes compared versions 1.0 and 2.32, and the author ordered one book in two parts to be made from them. Each time he decided the fate of the text in one sentence; the first of them came while the fourth attempt was already under way:

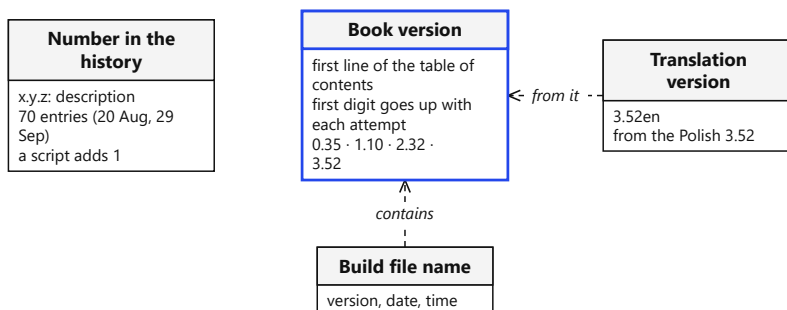
AUTHOR	2026-10-01 8:45 PM
[...] come on fix this, i want a new text, from the start completely, with a new map, we learn from mistakes and don't fix the previous one	
AUTHOR	2026-10-03 5:34 PM
create eksperyment-5, we'll be writing one book like in that review, add these texts as sources, add the canon, the facts we have, the whole output, get it?	
AUTHOR	2026-10-04 10:26 AM
let's not write thriller based on facts, that probably doesn't sell, let the output file not be called Volume 1, but Tell Me Yes and the version number, let's stick to that [...]	

Tell Me “Yes” is the book's third title. The author's decision on the title is dated 2 October – at that time it was the title of the first volume; the author decided on the file name on 4 October. “E2” in the first of the three commits below is the number of a question for the author in the decisions file, not the number of an experiment. Three entries from the history of the repository, each added by the author's assistant, that is, the model in the role abbreviated as ASY:

COMMIT 2026-10-02 10:32 AM [ASY] the author's decisions: E2 – title of volume 1 “Tell Me Yes”; pitch to be rewritten to YES

COMMIT 2026-10-03 5:36 PM [ASY] experiment 5: one book “Tell Me ‘Yes’” (Pt I + Pt II) following the review [...]; the whole output; E5 prompts [...]; e4 closed

COMMIT 2026-10-04 7:15 AM [ASY] build 3.11 after the night: the whole working draft (Pt I 34 ch., Pt II 43), md + html + PDF A5 221 pp.



Four version-numbering schemes in use in the repository at the same time. UML class diagram.

The oldest scheme appears in the commit subjects of 20 August and 29 September. Seventy subjects from that time begin with a three-part number and a colon; the last number was bumped by a script. The version of the book is set by the first line of the table of contents. The directory of the first attempt has no table of contents with a version number; from the second on, the first digit goes up by one: the second ended at 0.35, the third at 1.10 (the edition that was listened to had the number 1.0), the fourth at 2.32, the fifth reached 3.52 on 5 October at 14:27. The number after the dot is a running number, not a fraction: after 3.9 comes 3.10. The translation carries the number of the Polish version it was made from, with “en” added. The name of the file with the built book, that is, the build, contains, among other things, the version, the date and the time. There is no version number in the chapters; on the imprint page it is put in by the build.

	20 Aug	29 Sep	30 Sep	1 Oct	2 Oct	3 Oct	4 Oct	5 Oct
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
untagged	36	51	2	5		1	2	
total	36	126	56	203	104	83	338	37

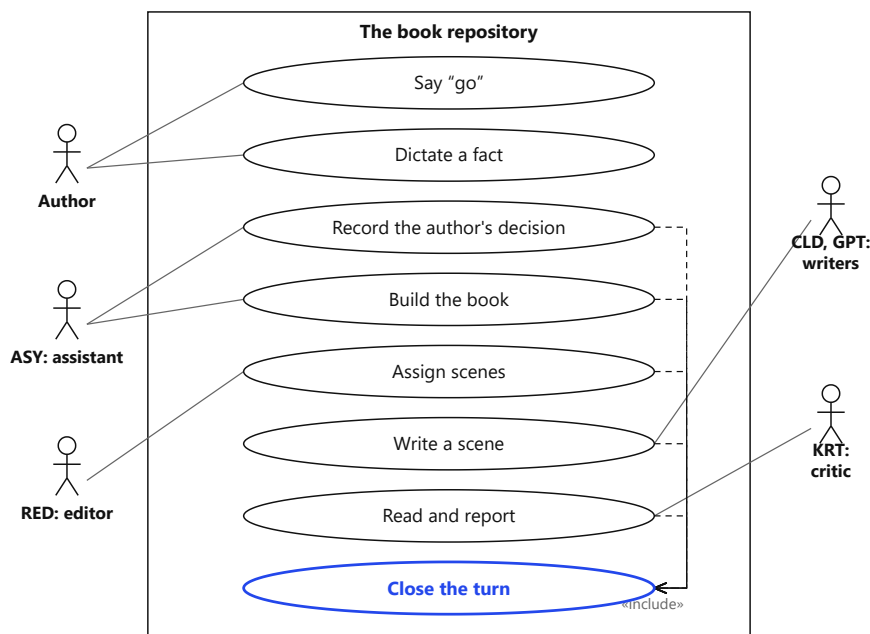
number of entries (commits) by role tag in the subject line; 983 in all, as at 14:27 on 5 Oct

Commits by day and role, from 20 August to 5 October 2026, 14:27. Heat map.

The role is shown by the abbreviation at the start of the commit subject. CLD and GPT are the two writers, Claude and GPT; RED is the lead editor, KRT is the critic. The remaining abbreviations are explained in the next chapter. Every role is a language model. The largest number of entries, 338, falls on 4 October; that day the first line of the table of contents went through 47 different version numbers, from 3.1 to 3.49. The highlighted cell is the largest number of entries by one role in one day: 131, the assistant, 1 October. Almost half of all the entries, 464 of 983, were added by the assistant, which does not have prose among its tasks. The number of commits measures activity in the repository, not the amount of text.

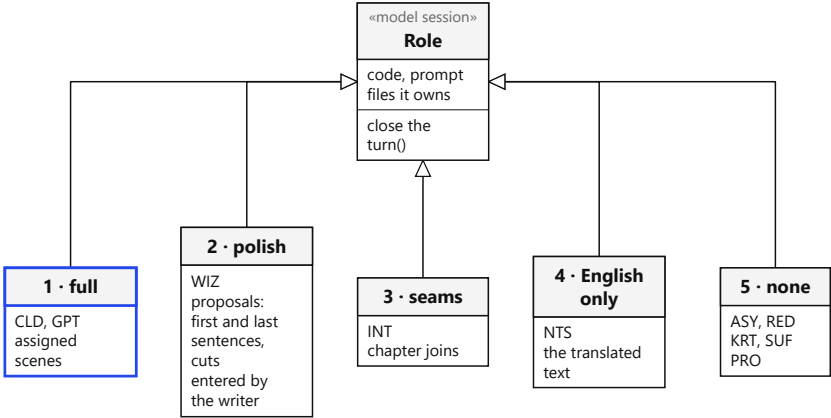
The Team

The team has eleven roles: ten model roles and the author. A model role is a separate session of a language model with its own instruction file, that is, a prompt, and with a three-letter abbreviation. The sessions do not talk to each other directly. They communicate through files in the book's repository and through commits, that is, entries in the history of changes; every such entry begins with the abbreviation of the role. Six roles made their first commit on 29 September between 17:26 and 19:53, the last one, the native speaker, on 5 October at 11:57.



Six of the eleven roles and their main activities; every piece of work by a role ends with closing the turn. UML use case diagram.

The author decides and dictates facts; with the word “go” he lets a role start work. ASY, the assistant – the model that runs the team and talks to the author – records his decisions and builds the book. RED, the lead editor, draws up the plan and assigns scenes. CLD and GPT, the writers, are Claude (a model made by Anthropic) and GPT – a model made by OpenAI, run in its Codex tool. KRT, the critic, reads and writes a report. The work of every role ends the same way. A turn is one piece of work by one role: from fetching the current files to the commit and a short report.



Five levels of access to the prose of the book, from full to none. UML class diagram.

Two roles may write prose, and only the scenes assigned by the lead editor. WIZ, the role the author named the wizard, polishes the first and last sentences of chapters and makes cuts – as proposals, which the writer enters. INT, the integrity checker, fixes the joins between chapters, usually two sentences per chapter at most. NTS, a model in the role of native speaker, changes only the English text. Five roles do not have prose among their tasks: the assistant, the lead editor, the critic, SUF, that is, the researcher, which searches the conversation logs for quotes, and PRO, the tools programmer.

	canon	decisions	outline	table of contents	chapters	promises	tasks
Author							
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 — owns the file; d — adds in a set place; blank — reads only. A chapter has one owner: the writer the editor assigned it to

Who owns which file, and who may only add to it. Responsibility matrix.

Every file has one owner. The canon, that is, the file of facts, is kept only by the assistant; the other roles submit proposals in a separate file. Every role adds questions for the author to the decisions file. The “outline” is the plan of scenes, the “promises” are a register of threads opened in one chapter and closed in another. The author's row is empty: the assistant writes to the files for him. The boundaries are guarded by the role's prompt, and after the fact by a script that flags as an error any commit by another role that changes the canon.

	ASY	INT	other roles
closing a turn	T	T	T
knowledge base	T	c	c
author's decision	T		
canon	T		
new attempt	T		
new role	T		
new skill	T	T	

T — runs it; c — reads only; blank — stopped by the role guard

Seven skills and the roles that may use them. Matrix.

A skill is a procedure saved in a file, independent of the role. One is used by every role: closing the turn. The rest are carried out by the assistant; with the knowledge base, the other roles only read its contents; the integrity checker may also create a new skill, subject to the author's decision. The first paragraph of a skill contains the role guard: a session that is not the right role is to stop at that point.

Claude and GPT run in separate applications. The assistant cannot start GPT; only the author does that. On the night of 3 to 4 October both writers worked in one working directory; on 8 October the team was still working that way. The window below is a reconstruction: four messages from the team's chat files, abridged, with cuts marked [...]. T5 and T6 are turn numbers, “main” is the main branch of the repository, and the names beginning with 2- are chapter files of Part II.

GPT 2026-10-04 4:34 AM
I am not touching Part I or the T6 files being written now by CLD.
CLD 4:35 AM
To GPT (Chat 019): 2-16a-ciemny-pokoik and 2-16b-petla have been on main since T5 [...] – don't write them a second time; fix them if you want.
CLD 9:45 AM
My mistake, to GPT: while replacing the separators I committed your new, not yet pushed file 2-11a-rollback.md. [...] If it was meant to go in later or differently – overwrite it.
GPT 9:53 AM
I accept the record; we don't roll back someone else's history, the content was mine and was checked in turn 3.

That same afternoon the Claude roles stood idle until the usage limit reset at 14:30; the job waiting was the acceptance read of freshly changed text. The second window is a reconstruction from the record of the author's conversation with the assistant; the last line comes from the queue file, in which the assistant left tasks for GPT and GPT added the result.

AUTHOR

2026-10-04 2:05 PM

can we hand something to GPT, because the credits are running out here

ASY

2:06 PM

Yes – and it's a good moment. [...] Until 14:30 Claude's roles are standing idle. [...] GPT is better at this than our roles: it reads the changed text with fresh eyes.

AUTHOR

2:16 PM

until 14:30 let GPT work, hand it the costly queued work

ASY

2:16 PM

The queue for GPT is in the repo (3a6746e) [...]. I won't be starting it, so just paste it this:

GPT

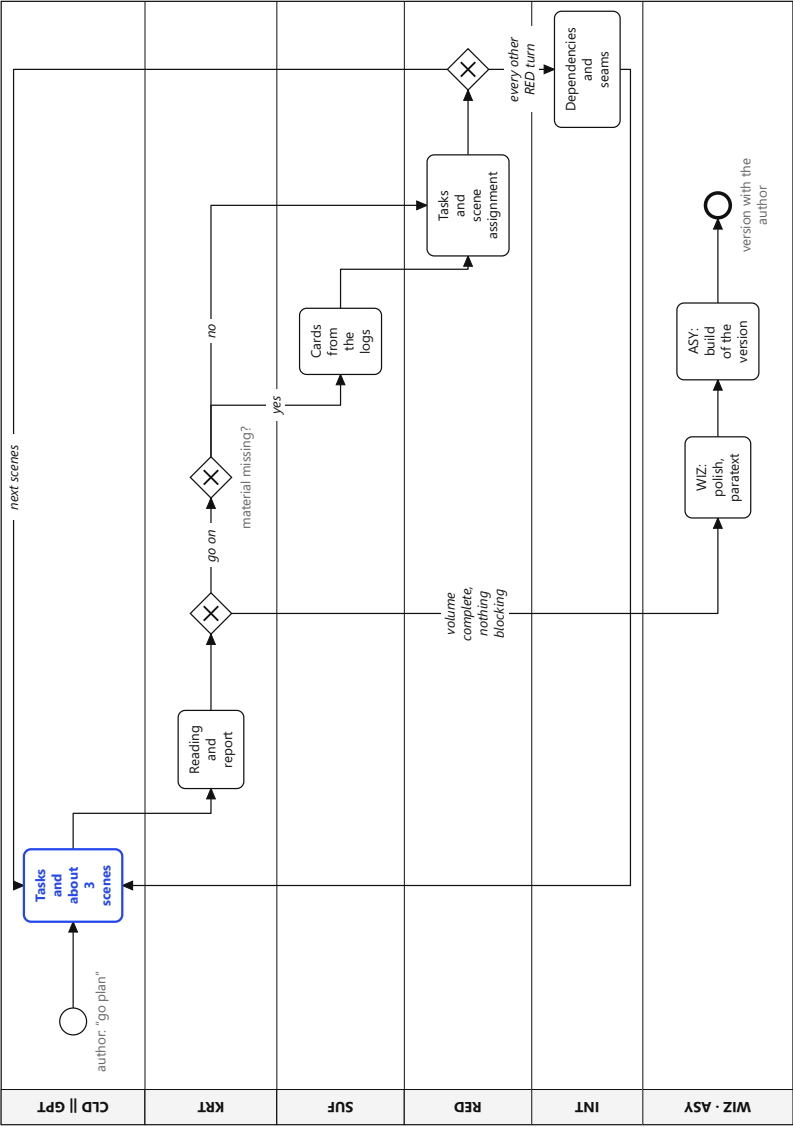
2:59 PM

Left unfinished: no task.

The order in the queue file ends with the sentence: “You don't send messages to Claude's session – ASY will see the commits.” “Left unfinished: no task” means: none of the tasks in the queue was cut short – GPT handed in all four.

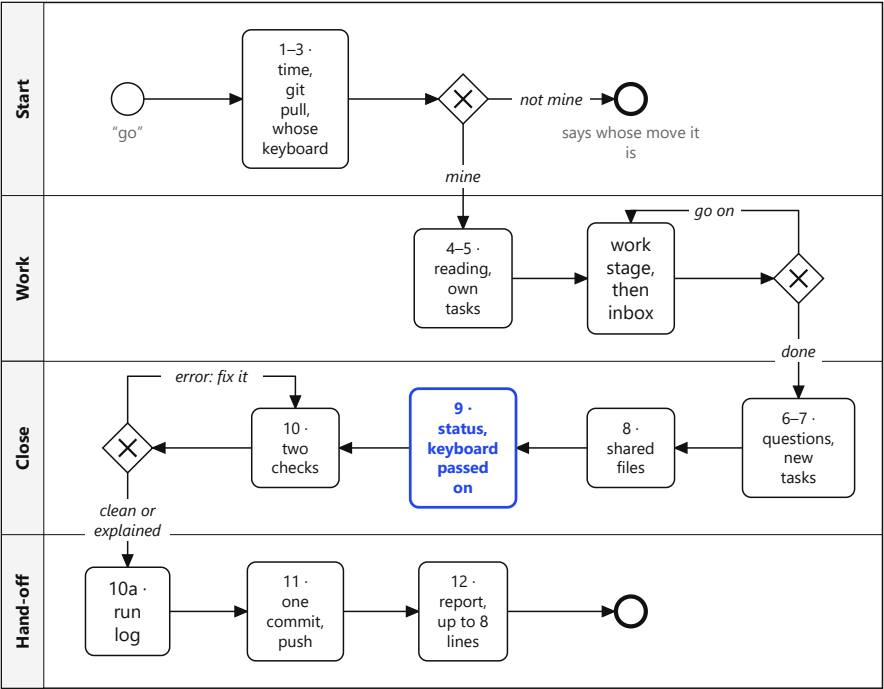
The Turn

The team that put this book together consists of separate sessions of language models, each in one role. The sessions do not talk to each other directly. They leave files for each other in the repository, that is, in a shared directory with a full history of changes. One piece of work by one role, from fetching the current state to saving its own change, is called a turn. At any given moment the role at work is the one indicated by the first line of the status file. That line is called the keyboard.



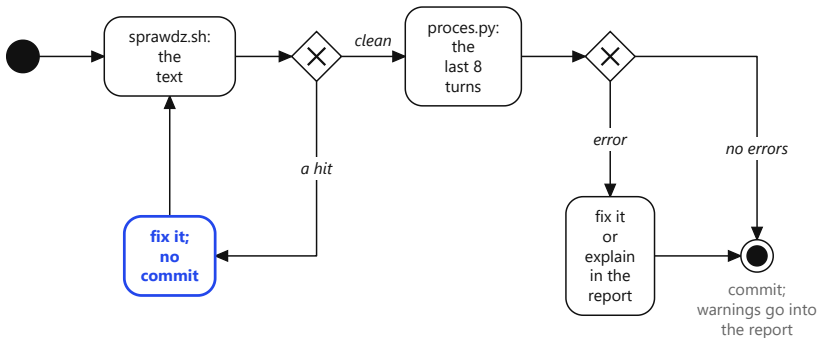
The writing cycle: who gets the keyboard after whom. BPMN diagram with lanes.

Every lane is a role, and every role is a model: the writers CLD (Claude) and GPT (an OpenAI model run through Codex), the critic KRT, the researcher SUF, which looks for material in the conversation logs, the lead editor RED, the integrity checker INT, the wizard WIZ for first and last sentences, and the assistant ASY. The author appears at the beginning: with the words “go plan” he approves the outline, that is, the plan of the book's scenes; after that his short “go” starts every turn of a role. The highlighted box is the lead editor's move: tasks and about three scenes for one round. Then the keyboard circulates among the roles. What returns to the author is the build, that is, the whole book built into one file.



One turn of any role: steps 1–12 and step 10a, added later. BPMN diagram; the lanes are the phases of the turn.

Every role goes through the same steps. It starts after the word “go” from the author, reads the clock and checks whether it holds the keyboard. If not, it tells the author whose move it is, and stops. During the work, after each stage, it checks the inbox: a file in which the author or the assistant leaves an instruction for a role that is in the middle of a turn. There are three instructions: SAVE, FINISH and STOP. Step nine is highlighted, because without it the cycle stops: the role writes in the first line of the status file who goes next and what it is to do. The turn ends with one commit, that is, one entry in the history of the repository, and with a report for the author, eight lines at most.



Two checks before the commit. UML activity diagram.

Step ten is carried out by two scripts. The first, `sprawdz.sh`, reads the text and the working files: it looks for real names, version markers and old wordings that must not come back. After a single hit the role does not make the commit. This is a step in the role’s instructions, not a lock in the tool: the script is not hooked into git, and once it was skipped (the chapter “The Name”). The second, `proces.py`, reads the history of the last eight turns and checks whether the roles kept to the agreed order. It distinguishes an error from a warning. An error is fixed by the role or explained in the report; warnings go into the report.

	sprawdz.sh	proces.py
a real name in the text	error	
a version tag in the text	error	
a phrase from an old version	error	
a pre-protection wording	error	
word count, version	information	
no keyboard line		error
canon changed by another role		error
a task with no owner		error
undated or unlisted chapter		error
sprawdz.sh fails		error
commit not as the author		warning
a turn with no run log		warning
keyboard not passed on		warning
a message with no date		warning

Selected checks of both scripts: what is an error and what is a warning. Table.

An error is whatever damages the book or someone else's work: a real name in the text, a task without an owner, a status file without the keyboard line, a change in the canon, that is, in the file of facts, made by a role other than the assistant. A warning is a missing trace: a turn without a run log, that is, without the role's note on how it worked, a message without a date, a keyboard the role did not hand over, a commit made under an identity other than the author's – all the roles write to the history under his identity; the role is recognised by the abbreviation in the commit subject, and a Claude model by the footer.

One round of the cycle on 4 October took twenty minutes. The window below is a reconstruction: four messages from the team's chat files, abridged, with cuts marked [...]; the figures are written as in the files. Numbers such as 1-09 are chapters of Part I, “KRT 006” is the critic's sixth report, and “day” is the date of a scene.

• • • KRT 2026-10-04 11:04 AM In this state – not for the general reader. [...] Most will put the book down in 1-09-1-11.
RED 11:09 AM The middle of Part I, cut by half [...]. We cut, we don't add; nothing moves to another day.
CLD 11:20 AM after KRT 006: the middle of Part I from 47.9 to 26.4 pages
KRT 11:24 AM The middle holds up. [...] The place where I was putting it down (1-09-1-11) is no longer there.

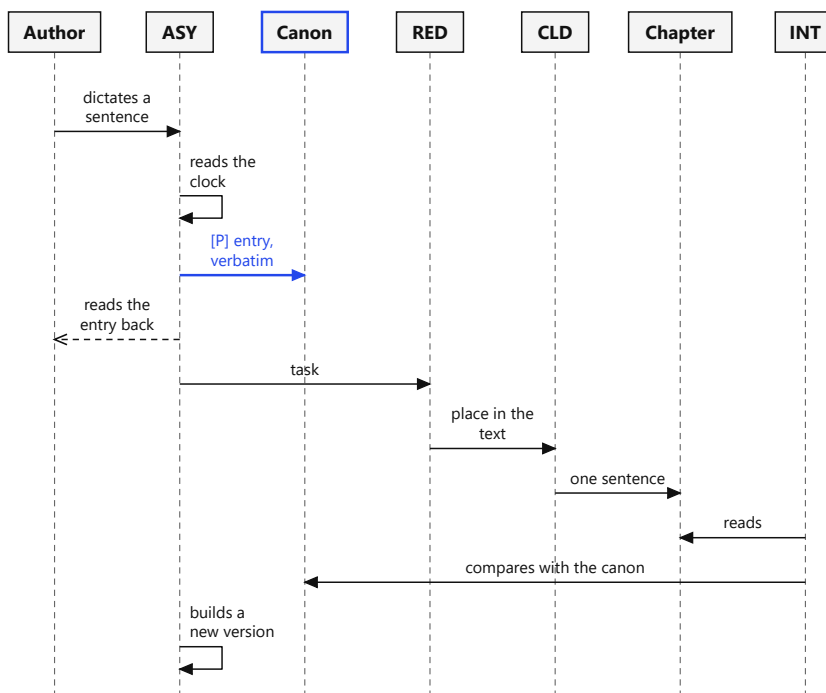
The Fact

The facts this book rests on are kept in one file. The team calls it the canon. Each fact is one line with a tag: [P] confirmed by the author, [?] to be confirmed, [X] refuted. Two rules apply. A new fact goes into the canon first and only then into the text. A missing fact means a question to the author, never filling in the gap. On 5 October at 15:00 the canon had 413 lines; not counting the legend in the file's header, 270 of them carried the tag [P], 81 [?] and seven [X]; the rest are headings, spacing and notes without a tag.

FAKTY.md

- [P] Asked who in “The Medal” says “All right. Then one more candle. For my mother. If that's how it is.”, the author, 2026-10-05 13:58, word for word: “that's me saying it, it's quite important”.
- [?] the exam and the night of 7 October 2023 [...]: the author, 2026-10-02 10:53: “I don't know which 3 million this is about” - he doesn't remember; the order of the exam [...] stays unestablished, in the text “in the autumn”, with no day.

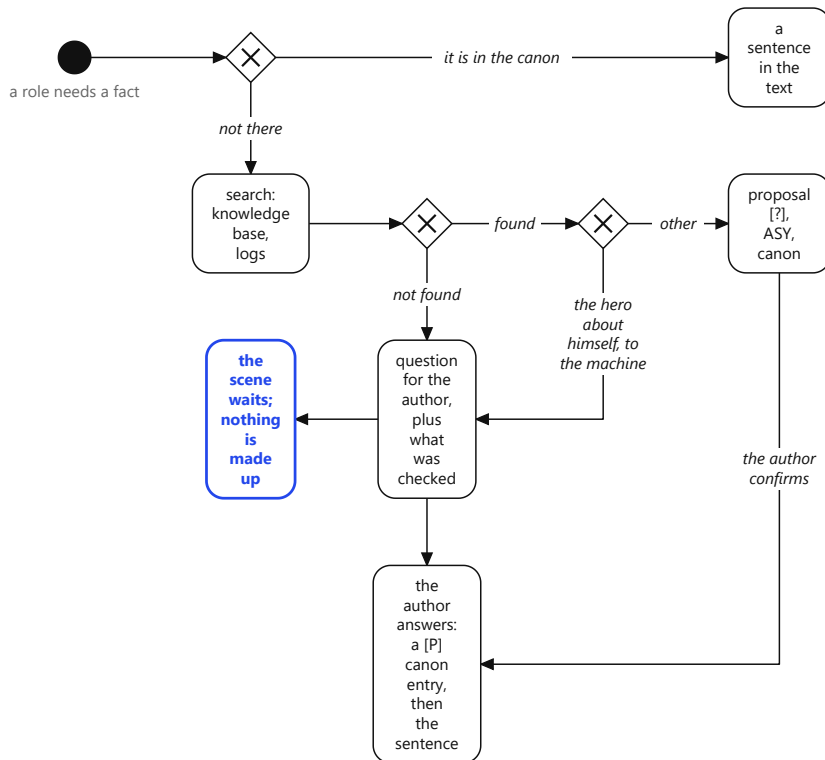
The first line closes a question about one sentence in the chapter “The Medal”. The second, abridged, with cuts marked [...], shows what happens when the author doesn't remember (the exam and the “3 million” are matters from Part I): the entry stays open, and the text says only as much as is known. The quoted lines of the canon contain no value from the table of changes.



The path of a fact from the author's word to a sentence in a chapter. UML sequence diagram.

The author does not write to files. He dictates to the assistant (ASY), the model that runs the team and talks to the author; it records the sentence in the canon word for word, with the date and time it has read from the clock, and reads the entry back to the author. It may enter as [P] only what the author said outright. A place, a person or a date inferred from context gets [?] and a question. Only then does the lead editor (RED, a model) point to the place in the text, the writer (CLD, a model) adds the sentence, and the integrity checker (INT, a model) compares the chapter with the canon. When the author answers a role other than the assistant, that role records his words in a separate file of proposals; the assistant moves them into the canon. On 5 October one word went into “The Medal”: “mówię” – “I say”. The assistant entered it into the chapter itself, outside this path – in one commit, together with the entry in the canon:

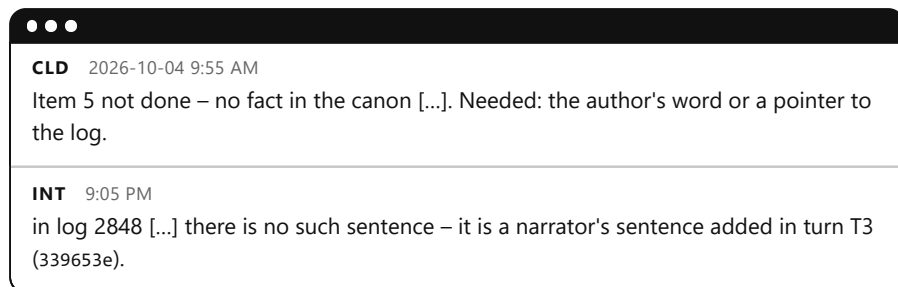
COMMIT 2026-10-05 2:02 PM [ASY] 3.52 / 3.52en: "The Medal" - author: "that's me saying it" (canon, then text: "– Dobra – mówię." / "– All right, I say."); translation memory, build of both languages (text), report



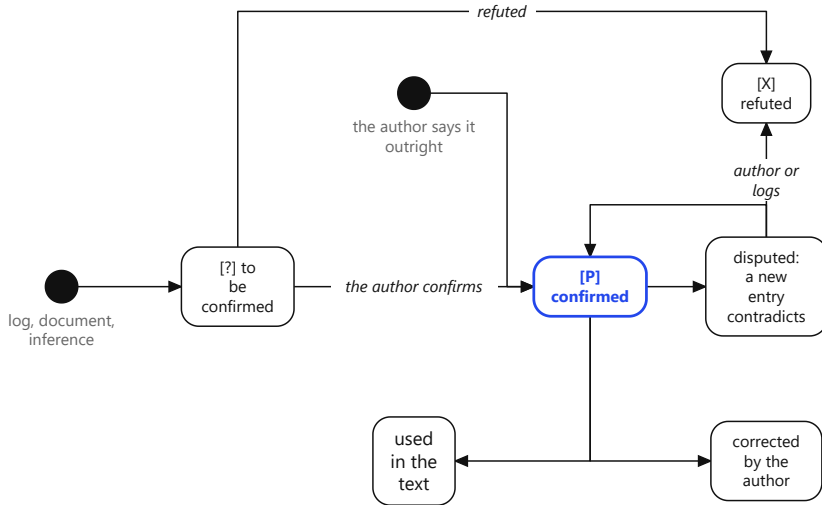
What a role does when it needs a fact. UML activity diagram.

Since 1 October there has been a fixed sequence: before a question goes to the author, the role looks for the answer in the knowledge base, that is, in the collection of documents from outside the conversations, and in the logs, that is, in the records of the author's conversations with the chatbot. Something found in the logs is not yet a fact. When the hero – the author, as he appears in the book – tells the machine, that is, the chatbot, who he is or who is standing next to him, it may be a game played with the machine; the author decides. When there is nothing, the role records the question together with what it checked. The scene waits, or works around the unestablished detail.

On 4 October the writer refused the lead editor's request to add a sentence, and the integrity checker found in a chapter a sentence that was not in the conversation log. The window is a reconstruction: two independent messages from the team's files, abridged, with cuts marked [...]. “Item 5” is the fifth item on the lead editor's list, “log 2848” is the number of a conversation log, and “T3” is the number of a turn; the commit's short hash is given in brackets.



The sentence the checker pointed to was dropped from the chapter.

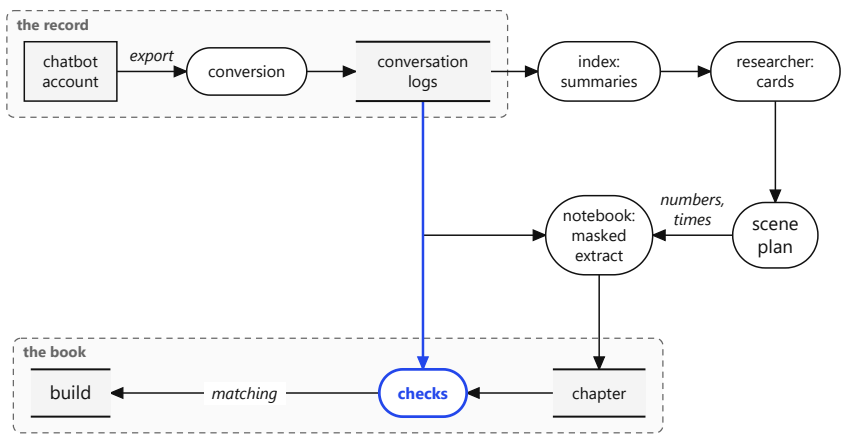


The states of an entry in the canon. UML state diagram.

There are three tags, but an entry can be in more states than that. A confirmed fact becomes disputed when a new entry contradicts it. The old entry is not deleted then; the author or the logs decide. The author can also correct a fact: then the entry is corrected, and so is every chapter that rested on it. One arrow leads to the text, and it starts from the state [P].

The Quotation

In Parts I and II, 44 per cent of the words are quotes: the record of the hero's conversations with the chatbot (version 3.53 of 5 October; only the words of the messages are counted, not the times or the speaker labels). Each such line has a time, a speaker and content. Each of these three elements is checked by a separate script, which compares it with the logs, that is, with the records of the conversations exported from the author's account. This chapter shows the path of one line from the export to the page.

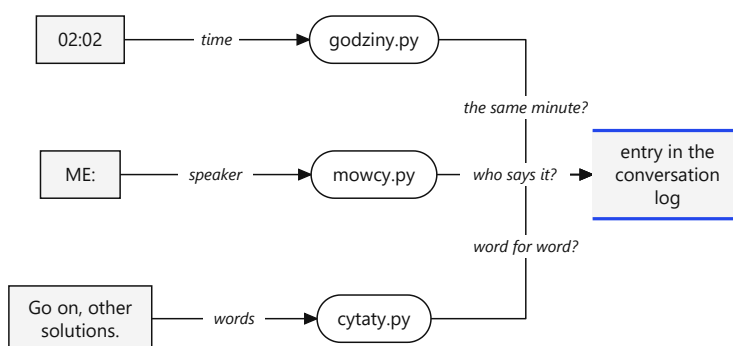


From the export of the conversations to a page of the book. Data flow diagram.

The author downloaded the full export of the conversations from his account. The tools programmer (PRO, a model) turned it into text logs, one file per conversation; the last one is numbered 3884 and dated 29 September 2026. Nobody reads all of this when writing. Models summarised the conversations in the index, the researcher (SUF, a model) picked cards from them (notes on one subject, with word-for-word quotes and conversation numbers), and the lead editor (RED, a model) entered conversation numbers and times in the scene plan. The writer gets a notebook: a short extract of word-for-word exchanges in which a script has masked names, telephone numbers and addresses. There are 24 notebooks. The checks compare the finished chapter not with the notebook but with the full logs; that is the highlighted line.

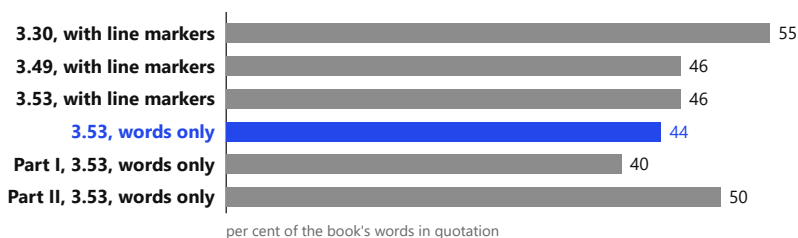
A quote line in a chapter file looks like this (the number in the file name is a trace of an earlier arrangement, not a position in the table of contents; the directory is still called tom-1):

```
tom-1/rozdzialy/1-03-krople-dla-psa.md
> 02:02 ME: Go on, other solutions.
```



The three elements of a quote line and the three scripts that check them. Flow diagram.

The script `godziny.py` looks in the log for an entry by the same speaker that contains the beginning of this text, and compares the time. The script `mowcy.py` establishes whether the sentence appears in the log under “ME:” or under “GPT:”. The script `cytaty.py` splits the quote at each cut mark [...] and looks for each fragment in the full log. In version 2.32 of 2 October – the text of the fourth attempt – it checked 1,112 fragments; none failed to match. It checked 622 timestamps then; all matched. In version 3.67 of 9 October the same script, which compares character for character, found 266 of the 612 fragments word for word: since 4 October the spelling of the author's messages has been corrected and names replaced, so the text deliberately differs from the record. A comparison ignoring capitals, punctuation and diacritics finds a further 274. Of the remaining 72, in 55 half of the fragment matches; the last seventeen the integrity checker found in the records itself: they differed by a substituted name, a corrected typo or a dictation fix.



The share of quotes in the words of the book, by two measures. Bar chart.

The share is counted by the team's script, separately in each chapter. It also counts the line markers as words of the quote: the quote marker, the time and the speaker label. By this measure, on the morning of 4 October, in version 3.30, the chat record made up 55 per cent of the book. That day the lead editor set a target: a continuous block of quotes up to around 150 words, a chapter below around 65 per cent, the whole 45–48 per cent. In the evening, in version 3.49, it was 46 per cent, and it is the same in version 3.53. Without the line markers, version 3.53 has 23,455 words, of which 7,121 are the author's words in quotes (30 per cent) and 3,215 the machine's words (14 per cent); together 44 per cent. Without the markers Part I is 40 per cent quotes, Part II 50.

The faithfulness of a quote has one exception. It arose on the afternoon of 4 October, when the author was looking at a page with his own sentence, once dictated to the chatbot by voice and garbled by speech recognition. The window is a reconstruction from the record of the author's conversation with the assistant (ASY) – the model that runs the team and talks to the author – abridged, with cuts marked [...]. In this appendix swear words stand in square brackets; in the chapters of Parts I and II – as in the record.

AUTHOR 2026-10-04 4:44 PM
[you messed it up], he says ... (come on fix obvious errors of this kind, they add nothing and only spoil things, go through all the quotes once more)
ASY 4:46 PM
accepted and recorded as a new rule (417cb61): we correct obvious speech-recognition errors in your messages without brackets; we don't touch the machine's words. [...] One question about this particular sentence. The log has "[you messed it up] you say she'd rather in the morning", you wrote "[you messed it up], he says...". Which wording is to go in?
AUTHOR 4:46 PM
[you messed it up], she says ...

ASY 4:47 PM

recorded (05095fa), in the canon and in the decisions. The book will have: "[You messed it up], she says she'd rather in the morning, come on, rescue the situation."

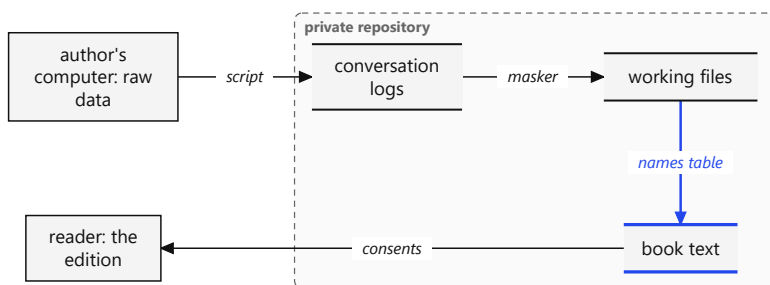
AUTHOR 4:50 PM

go ahead, I'm very curious how it came out :)

The rule recorded that day at 16:45: obvious speech-recognition errors in the author's messages are corrected without square brackets, and the machine's messages are not touched. In this sentence the correction goes further than one garbled word: the log has "you say she'd rather in the morning", the book "she says she'd rather in the morning". The author judged it a dictation error; both wordings are in the decisions file.

The Name

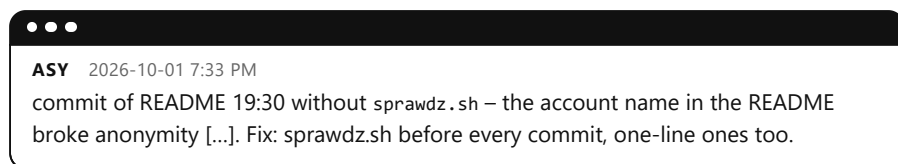
The people from the author's life have different names in the book than in life. This is enforced not by the models' memory but by one table and one script. The table sets the real form next to the form used in the book. It is the only place in the repository where the two stand side by side; the file has the words “NOT FOR PUBLICATION” in its header. No real name from that table appears in the Appendix.



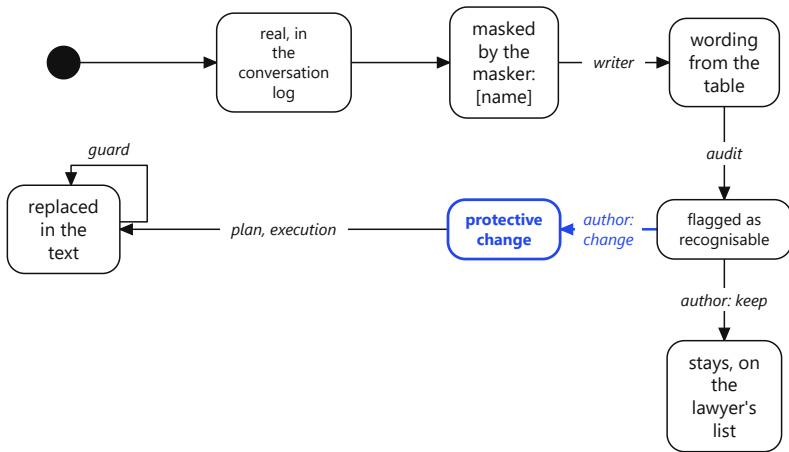
Data zones: from the files on the author's computer to the edition. Data flow diagram.

The data passes through five zones – the three middle ones are in the private repository – and each boundary stops something. The full export of the conversations, the original photographs and the recordings stay on the author's computer. The conversation logs with the real names go into the private repository; models in the cloud read them too. The masker – the script that prepares extracts from the conversations for the writing models – replaces names, telephone numbers, e-mail addresses and postal addresses with words in brackets. A name enters the text of the book only in the form given in the table. An edition leaves the repository only after Bella, the hero's wife, consents to publication, after the lawyer's opinion and the author's decision. Working versions such as this one the author sends to chosen readers with her consent; on 8 October he did not yet have consent to publication.

The role runs the checking script itself before every write to the repository's history, that is, before every commit; there is no technical lock. This rule has a date. The assistant (ASY) – the model that runs the team and talks to the author – recorded in the process log, that is, in the file in which the team notes errors and rules, after the author's account name had appeared in the README file:

A terminal window with a dark background and a title bar containing three white dots. The text inside the window is white and shows a timestamp, a username, and a commit message.

```
ASY 2026-10-01 7:33 PM
commit of README 19:30 without sprawdz.sh – the account name in the README
broke anonymity [...]. Fix: sprawdz.sh before every commit, one-line ones too.
```



The states of a name and of an identifying detail on the way into the text. UML state diagram.

A name from a conversation log does not go into a chapter directly. In the extract it is already a bracketed word; the writer puts in the form from the right-hand column of the table, and in a quote from a conversation encloses it in square brackets, so that the substituted word does not pass for a word from the log. A separate audit looks for details by which a person could be recognised despite the changed name: names of things, dates, times. The author decides on each one. A changed detail gets a row in a second table, “was → is”, and a guard in the checking script reports an error when the old wording comes back into the book. Above the second table is this sentence:

mapowania/MAPOWANIA.md

These are ****deliberate inventions**** - not facts: `FAKTY.md` stays true, and in conversations with the author and in the canon we use the left column.

So the canon, that is, the file of facts, stays true, and the book departs from it only where the second table says. The decision on the scope of the changes was made on 4 October in one sentence, the answer to five “yes or no” questions; the time comes from the decisions file, in which the assistant records the author's decisions.

AUTHOR

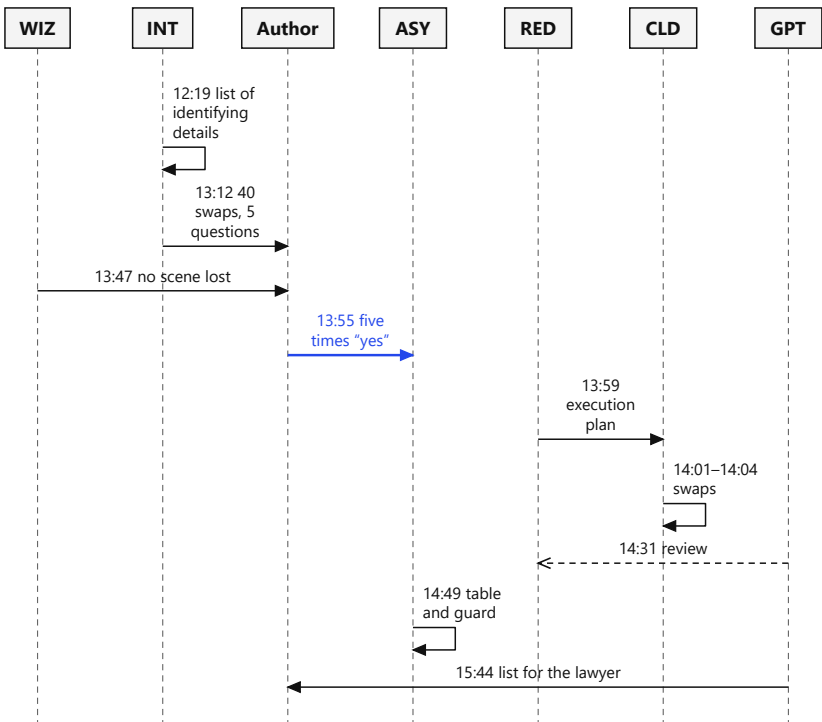
2026-10-04 1:55 PM

yes, we detach us as much as it can possibly be done, so that it can't be connected

COMMIT

2026-10-04 1:55 PM [ASY]

the author's decision: protection of the people in the book - all five “yes”, we detach the author and those close to him as much as it can possibly be done

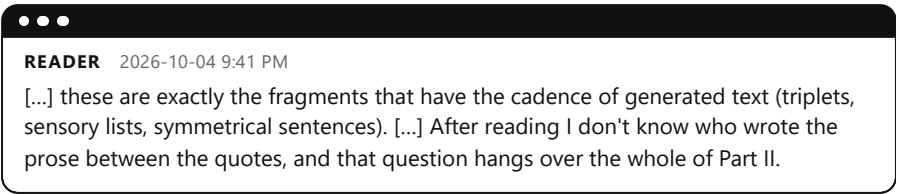


Protection of the people in the book on 4 October 2026: from the list of features to the list for the lawyer. UML sequence diagram.

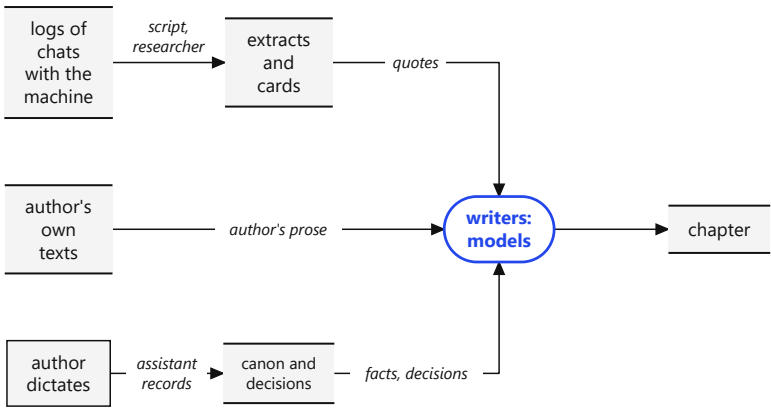
From the first list to the list for the lawyer, three hours and twenty-five minutes passed. All the roles apart from the author are models. The integrity checker (INT) listed the identifying features and proposed forty replacements with five “yes or no” questions. The wizard (WIZ) rewrote the proposal so that the author's five “yes” answers would not cost a single scene. The lead editor (RED) drew up the plan four minutes after the decision; the writer (CLD) carried it out in five minutes. The second writer (GPT) read the result as an outsider, and then drew up a list of 78 items for the lawyer – “without legal assessment”. The assistant added the second table and the guard.

Who Wrote It

One of the fresh readers – a model in the role of the sceptical IT man, with no access to the repository, that is, to the directory of the book's files with the full history of changes – ended its read of version 3.47 like this:

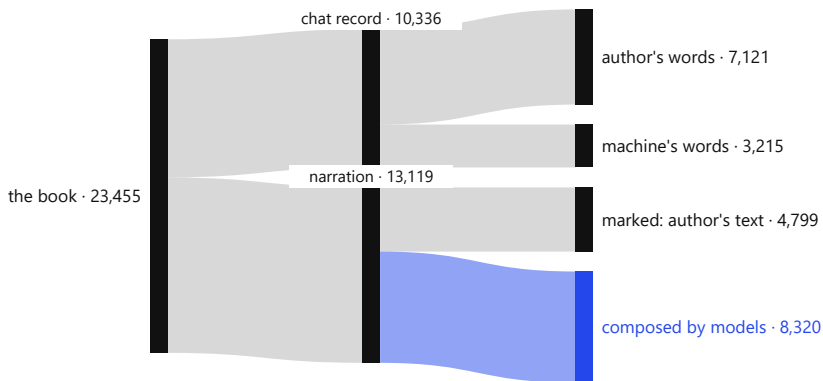


This chapter answers with numbers from the repository and says which numbers do not exist. The starting point is a sentence of the author's in the decisions file, the file with his decisions: “I don't touch the keyboard any more” – the one on the computer, not the right to move in the team.



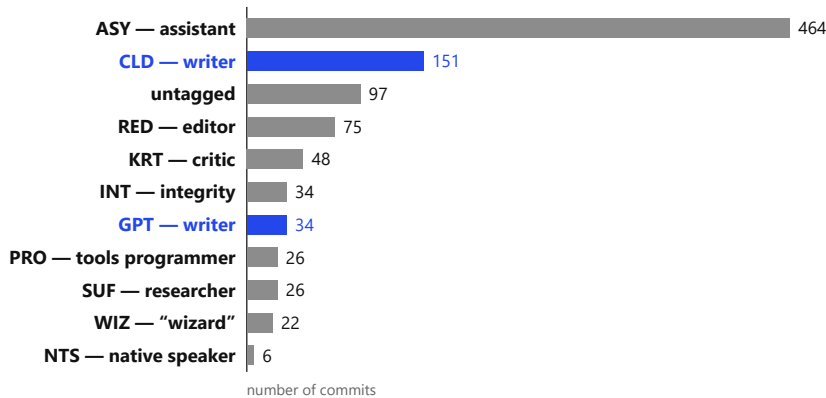
What a chapter is made from: three sources and the writers. Data flow diagram.

A chapter has three sources. The first is the logs of the author's conversations with the chatbot; the quotes come from them. The second is the author's texts from before the team existed, imported into the repository: four files, 25,676 words in total, and the manuscript of his earlier book. The third is his dictation: the assistant (ASY), the model that runs the team and talks to the author, records facts in the canon, that is, the file of facts, and decisions in the decisions file. In the canon 270 entries carry the tag of confirmation by the author (as of 15:00 on 5 October). Almost everything is entered into the chapter file by the writers – two models: CLD (Claude) and GPT (an OpenAI model run through the Codex tool).



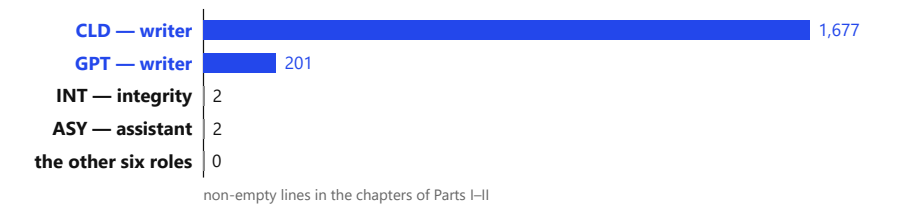
The words of Parts I and II in version 3.53 by layer. Sankey diagram.

In version 3.53 Parts I and II have 23,455 words in total, not counting titles, date lines and line markers (the quote marker, the time, the speaker label). Three things have to be counted separately. The verbatim chat record is 10,336 words: 7,121 of the author's words in 287 messages and 3,215 of the machine's in 131. The author's prose included in the book: six chapters carry the annotation “the author's text” in the table of contents – “The Pavement”, “Two Squares”, “The Night of White Whisper”, “Power of Attorney”, “The Diagnosis”, “Three Days” – 4,799 words of narration in total. This is a label from the table of contents, not a measurement: in these chapters the table of contents also records cuts, and fragments added from the conversation logs and from the canon. The remaining 8,320 words of narration were put together by the models from the author's conversations, dictation, decisions and earlier texts. The chat record is 30 and 14 per cent, 44 together; the author's prose 20; the models' narration 35.



Commits by role abbreviation, as of 5 October 2026, 14:27. Bar chart.

By 14:27 on 5 October the history of the repository held 983 entries, that is, commits. Each role marks its own with an abbreviation in the subject. The commit footers name Claude models: Opus 5.5 in 803 entries, Fable 5.1 in 87, Opus 5 in three. The remaining 90 entries have no footer, among them 33 of GPT's 34; the history does not record the name of its model. Almost half belong to the assistant, which does not have prose among its tasks (two lines of its own remain in the chapters): it records decisions, keeps the canon, builds the versions. The commits without an abbreviation come from 20 August and 29 September, from before the division into roles, and from merges. The author has no bar of his own: his decisions enter the history through the roles.



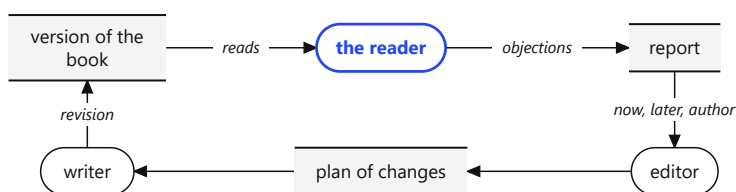
Non-empty lines of the chapters of version 3.53 by the role that changed them last. Bar chart.

The git blame command shows, for each line, the commit it comes from. In the 57 chapter files there are 1,882 non-empty lines; 1,878 were last changed by the writers. The remaining four were last changed, two each, by the assistant and the integrity checker (INT), the model that fixes the joins between chapters. The lead editor, the critic, the researcher (which searches the conversation logs for quotes), the wizard (which polishes first and last sentences), the tools programmer (PRO) and the native speaker – models too – did not change a single line in them. This measurement says who entered a line into the file, not who composed it. With copy tracking, the git blame command traces 207 lines, that is, 3,042 words, back to the author's files imported unchanged. That is fewer than the 4,799 words annotated “the author's text”: git attributes three of those six chapters entirely to the writer who entered them into the file.

There is no way to count how many sentences of narration survived in the model's wording after the author's decisions. In the “remaining narration”, paraphrase of his dictation cannot be separated from the model's sentences. The contribution of the roles that do not touch the text, but put together the plan and the lists of cuts, cannot be measured. A lower bound can be counted: 7,121 words of the author in the quotes and 3,042 words from his files, 43 per cent in total. It assumes that the texts from the author's files are his; the repository does not record how they were made. The repository does not know the upper bound.

The Critic and the Readers

In the team this book is read almost exclusively by models. The critic, the integrity checker, the outside reviewer, the “fresh readers” – in this chapter each of these names means a language model in a role, started with a description of who it is to be and what to ask about. Of the seven loops in which the text comes back for correction, a human reads regularly in one: the author, when he listens to the recording; outside the loops he looks at single pages.



One quality loop: from a version of the book to the next version. Data flow diagram.

Every loop has the same shape. Someone reads a version and hands in a report with objections. The lead editor – a model – divides the objections into three groups: to do now, for later, questions for the author. The writer carries out the plan and a new version is produced, which someone reads in turn. On 4 October at 11:04 the critic wrote in a report: “the general reader reads through people, not through screenshots.” Twenty minutes later it received the middle of Part I cut – by its own count – from 44 pages to 24 (the writer counted the same cut as 47.9 → 26.4).

	who reads	who decides	who fixes	how often
automatic checks	scripts	the same role	the same role	every turn
critic	model: critic	editor	writer	every cycle
integrity	model: checker	editor	writer	every other cycle
the author listens	author	assistant	whole team	every edition
outside review	model outside the team	assistant	editor, writer	2 reviews, 7 simulations
fresh readers	three model personas	assistant, editor	writer	4 times in 90 minutes
stops	every reading	editor	writer	since 5 October

Seven quality loops: who reads, who decides, who corrects and how often. Table.

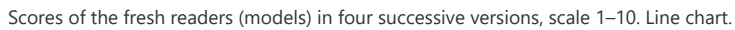
The loops differ in who reads and how often. Scripts check every turn, that is, every single piece of work by one role. In the critic's directory there are 21 reports from 4 October, in the integrity directory six audits. The outside reviews are two texts by a model from outside the team and seven simulated conversations between two hosts about the book, made with another tool, NotebookLM. The stop – a place where the reader breaks off reading because they do not understand something – has had a separate register since 5 October. The highlighted cell is the only place in the table where a human reads.

The window below sets side by side three independent records from 4 October: two sentences from the reports of the reader (a model) after versions 3.44 and 3.47 – the first about the chapter with the long calculation – and a sentence from the report of the assistant (ASY), the model that runs the team and talks to the author. The times are those at which the files entered the repository, that is, the directory with the full history of changes; under the title of every chapter of Part I there is a counter of days to the scene the book begins with – in the record: to the pavement.

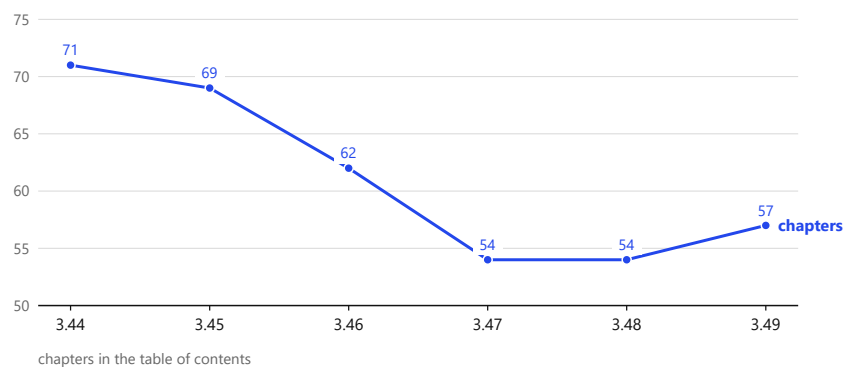
I don't understand this and I don't want to understand it; I get the point ("the machine agrees with him") after the first page, the rest is repetition. The first place where I skipped ahead.

I didn't put it down only because the counter showed "216 days to the pavement" and I knew the scene from the beginning would come back. Without that counter – a real risk that I would have stopped here.

These are test readings by models, not market research and not the voice of real readers.

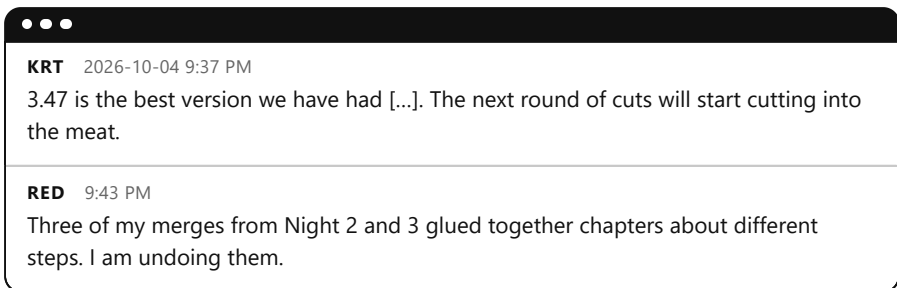


On the evening of 4 October the author started the night loop. Four times in ninety minutes a new version went to three fresh readers: separate runs of the model, with no access to the team's files, with the same questions. The bookshop reader, the publishing-house editor, the sceptical IT man. The IT man gave 7 four times; that line is not on the chart, because its label overlapped the label of another series. The bookshop reader's score rose by a point, the editor's score for the text fell by a point. Another measure changed: the readers' certainty about what Bella will answer – the book ends with her answer – fell from “I knew” in the first two rounds to 70–75 per cent in the fourth – the direction it was meant to go: according to that day's decisions, the tension of Part II is that nobody knows what Bella will answer. Versions 3.48 and 3.49 were not read by a fresh reader.



Number of chapters in versions 3.44 to 3.49, 4 October between 18:40 and 21:58. Line chart.

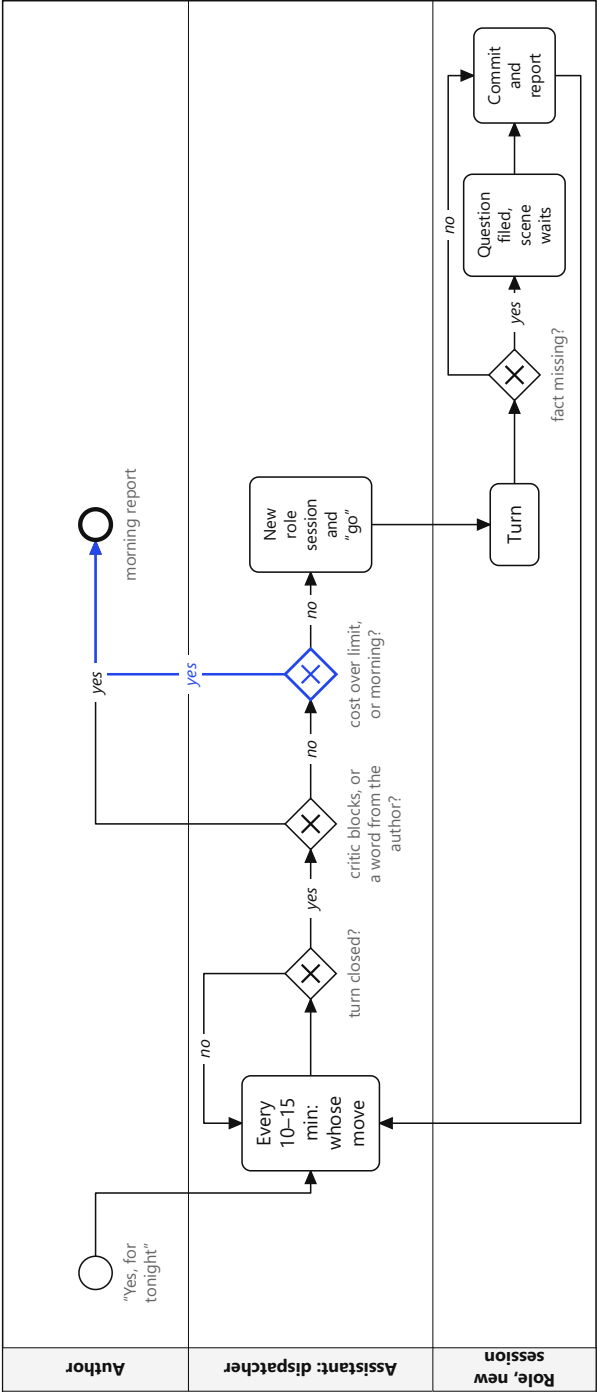
Between 18:40 and 21:58 the book lost seventeen of its 71 chapters and got three back; the word count fell, according to the assistant's report, which counts differently from the chapter on the quote, from 31,324 to 25,782. The reader in the role of the publishing-house editor wrote after version 3.46 under the heading “Cut to the bone?”: “At sentence level yes, in places even too much”. After the critic's (KRT) acceptance read, the writer restored five cut passages, and the lead editor (RED) undid its own merges in the plan; the team called the rounds of this loop Nights. Two separate messages from the team chat:



The assistant's conclusion after the fourth round: editing alone will not supply what is missing – material and the author's decisions.

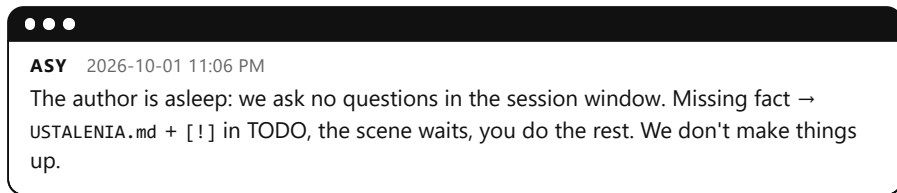
The Night

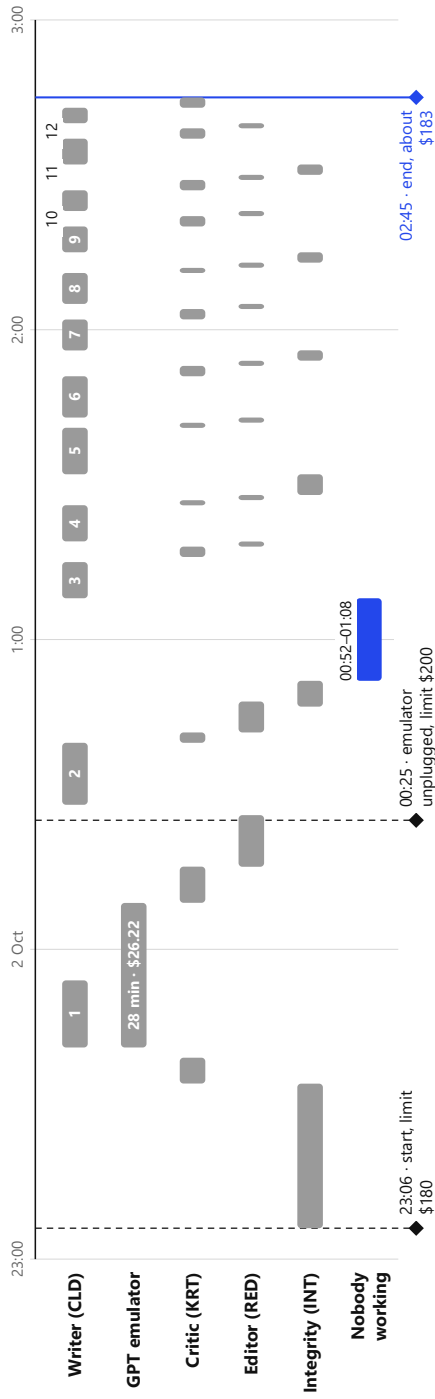
Three times the author gave the team of models a night or an evening. For that time the word “go”, with which the author lets a role start a turn, passed to the assistant (ASY). A turn is one piece of work by one role, from fetching the files from the repository (the directory with the full history of changes) to the commit, that is, the entry in that history. The assistant is the model that runs the team and talks to the author; at night it is also the dispatcher. The chapter shows the rules of night mode, the first night turn by turn, and what it cost. The chat windows are reconstructed from the files and from the record of the assistant's session; they are not screenshots. The times and costs of the first night come from the process log, the file in which the assistant records the course of the work. The assistant read the costs from the list of sessions in the application, which gives them for each session – a reading from the application, not a bill. Since 1 October, 22:55, it has read the times of the entries from the clock, with the same command that writes the entry; the earlier error is in the chapter “Lessons”.



Night mode: who gives “go” when the author is asleep, and what stops the loop. BPMN diagram.

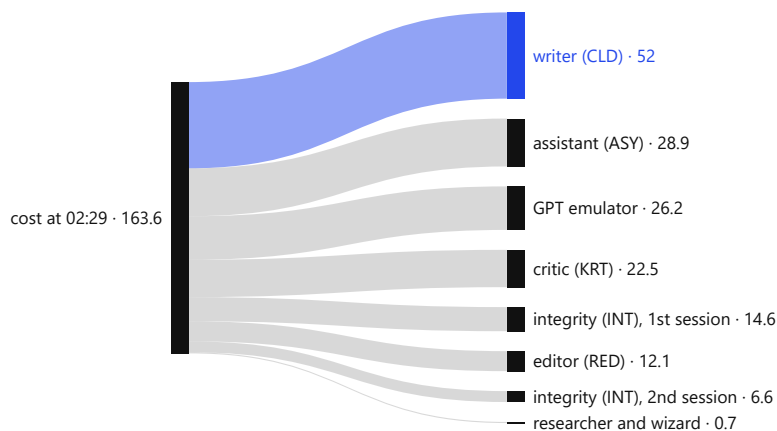
Every ten to fifteen minutes the dispatcher checks who has the keyboard, that is, the right to move recorded in the first line of the status file. When the previous role has closed its turn, the next one gets a new session and “go”. The loop is stopped by the critic's blocker (the model in the role of critic decides that writing must not go on), the cost limit, the morning or a word from the author. A separate rule concerns facts. The rules of the first night (USTALENIA.md is the decisions file):





The night of 1 to 2 October: the roles' turns from 23:06 to 02:45. The ends of the bars follow the process log; the starts, and in three turns the ends too, are approximate. Gantt chart.

That night the writer (CLD, the Claude model) completed twelve rounds, the critic (KRT) thirteen reports. After the critic the lead editor (RED, the model that keeps the plan of the book) distributed the corrections, and every other time the whole text was read by the integrity checker (INT); the night began with its long audit, closed at 00:52. In the first round a GPT emulator worked beside the writer: a Claude model (in the log: Fable; the writer is Opus) put in the place of the second writer, which only the author starts. At 00:25 the author, who was not yet asleep, unplugged the emulator and raised the limit from 180 to 200 dollars. The log records one gap, sixteen minutes: a role finished at 00:52 and did not report, and the dispatcher was then checking every eighteen minutes, less often than the rule says. The trial build, that is, the file assembled from all the chapters, grew between 00:48 and 02:40 from 43.0 to 88.8 pages.



Cost of the night by role, as of 02:29, in dollars. Sankey diagram.

The limit covers the sum of the costs of all sessions since 23:06. The dispatcher does not start a turn if the sum plus the estimate for that turn would exceed the limit. The last breakdown by role is given by the log at 02:29: 163.6 dollars, of which 52.00 went on the writer, 28.92 (18 per cent) on the dispatcher itself, and 0.67 on the researcher and the wizard together. At 02:43 it was around 179 dollars; another full cycle, estimated at around 15, would have made it around 194. The dispatcher recorded that this exceeded the “safe margin” – the log does not say how large it was – so only the critic got “go”. The night closed at around 183 dollars out of 200. The log does not give a cost per chapter.

The night of 3 to 4 October had no limit in dollars. The limit was the usage allowance of the author's subscription plan, with no paid usage on top. The assistant was set an alarm for 03:05, when the allowance resets. At that time two instances of the assistant, that is, two separate sessions of the same role, woke up: on the author's computer (local) and in the cloud. They had no shared window. They talked in rows of a table in the inbox file; in their record an agent is a separate run of the model in one role. In the Polish record the local session addresses the one in the cloud in the feminine, and that one writes about itself in the masculine; that is how it stands there.

ASY (cloud) 2026-10-04 3:06 AM
I am starting the role agents one after another [...]. A session that is already working on the same thing – report in this table, I will give way.
ASY (local) 3:06 AM
Please give way, as you offered: do not start your own role agents – two KRT readings and two RED turns at once mean double cost and conflicts in the files.
ASY (cloud) 3:17 AM
I give way – I saw the request from 03:06 after starting; by then I had launched one KRT agent [...]. ASY local runs the night cycle.

After four in the morning the author started GPT, the second writer, and went to sleep. The times in the window are the timestamps of the session record.

AUTHOR2026-10-04 4:31 AM

gave him two more turns, going to sleep carry on

ASY4:32 AM

Good night. Plan for the rest of the night: [...] Computer on, application open – the rest is up to me.

AUTHOR4:32 AM

manage them until the book is written

ASY4:32 AM

Recorded in your words (commit 9250d34). The mandate does not end in the morning: I dispatch the E5 sessions until the book is built.

On the evening of 4 October the author started the loop once more. The assistant recorded the stop condition itself:

ASY2026-10-04 8:30 PM

End of the loop: two iterations in a row bring only cosmetic changes – or the plan limit runs out (paid usage switched off).

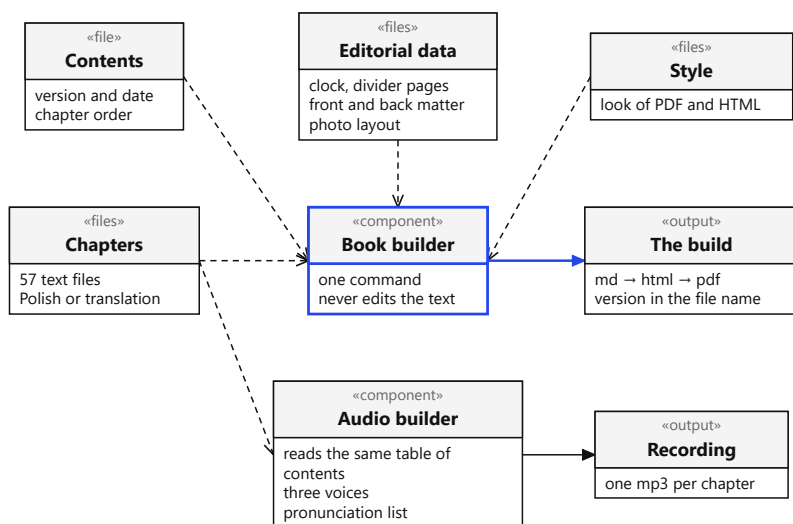
The loop ran until 22:00: four rounds, each with three fresh readers, that is, models with no access to the repository. The night report records that the plan limit did not interrupt the work.

COMMIT2026-10-04 8:30 PM [ASY] decisions 20:30: night loop - iterations, measure, limits, selling at the end

COMMIT2026-10-04 10:00 PM [ASY] night report: four rounds with fresh readers, version 3.49, default decisions and decisions for the author, whom to sell to and how

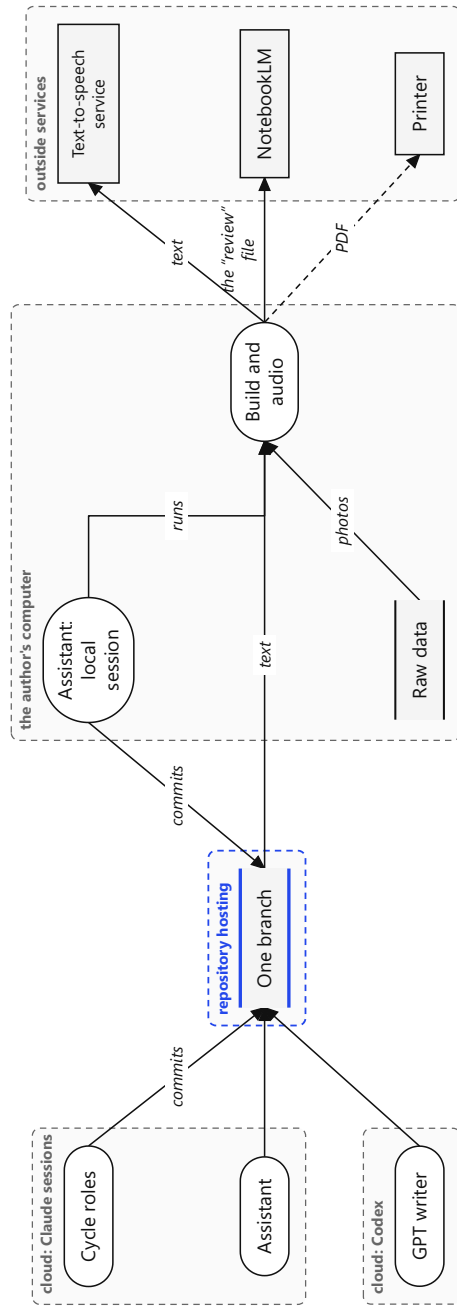
The Build

The build is the assembled book: one file made from all the chapters in the order of the table of contents, and from it a web page and a PDF. It is built by a script, not by a human and not by a model. The tools are maintained by the tools programmer (PRO), in the same session also the production manager: a model that does not write prose. The chapter shows where the builder takes its data from, what runs where, and how an editorial decision becomes a line in a file.



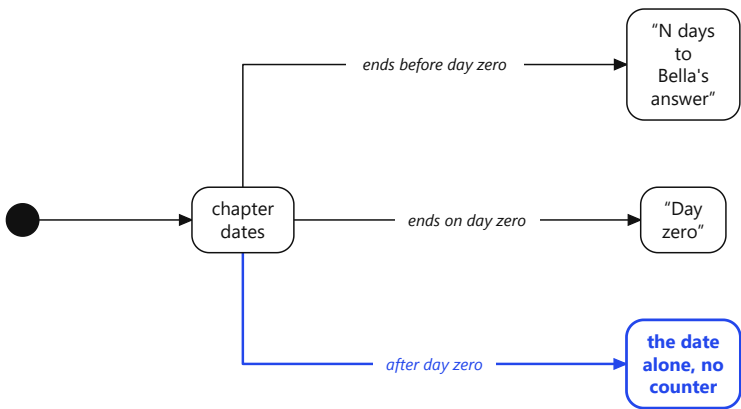
The build pipeline: input files, two builders, results. UML component diagram.

The builder reads the first line of the table of contents, the only place where the version number appears in the book's files, and the table with the order of the chapters. It does not change the chapters. The day counter under the title (the clock), the white pages with a subheading (the divider pages) and the positions of the photos are kept in separate files and can be changed without touching the program. The PDF file has the version, the date and the time of generation in its name; corrections are never made in it, only in the chapters. The second builder reads the same table of contents and records every chapter in three text-to-speech voices: narration, the hero, the machine.



What runs where: two clouds, repository hosting, the author's computer, external services. Deployment diagram.

All the sessions meet in one place: in the main branch of the shared repository, that is, the directory of files with the full history of changes (a second, working branch has been standing since 29 September). The roles of the cycle, that is, the models that write and check the text, work in the cloud. They do not see the full export or the original photos; they see the repository, and in it the conversation logs with the real names. At the end of the process the real names do not get out: a check before every commit and before publication guards that. The direction the author set on 8 October: mask as early as possible. The export and the originals stay on the author's computer together with the build, which the assistant's local session runs there; the PDF is made there by printing from the browser. Three things go outside: text to the text-to-speech service, the file built in “review” mode to NotebookLM, in which two model-generated voices talk about the book, and the PDF for the printing firm. This last line is dashed: the requirements of three printing firms are written down (according to that list, one file will not pass at all three if the photos bleed), the trim size is 6 × 9 inches (earlier A5), and no record of the file being sent to print was found in the repository.



The clock of Part II: three captions under the chapter title from one rule. UML state diagram.

In Part II a counter of days to Bella's answer appears under the chapter title. The build calculates it from dates. A chapter that ends before day zero gets the number of days; one that ends on day zero, the caption “Day zero”; every later one the date alone. The two dates and the wording of the counter are three lines of one file (“chodnik” is the pavement, “etykieta” the label):

```
tom-1/RUCHY.md
chodnik: 2025-01-03
zero: 2026-07-14
etykieta: to Bella's answer
```

Part I counts the days to the first date, Part II to the second. The reason for the rule is given in the commit subject, that is, the subject of an entry in the history of the repository. A mini-turn is a short piece of work by a role outside the queue; R-9 is the number of a proposal by the lead editor, and the label was changed that same evening to the one in the file.

```
COMMIT 2026-10-04 6:18 PM [PRO] mini-turn: fix R-9 - Part II clock “days
to her answer”, after day zero the date alone (the clock does not give away
the ending); label and dates in RUCHY.md
```

```
COMMIT 2026-10-05 10:11 AM [PRO] ksiazka.py --jezyk en: English edition
from przeklad/en/ (headings, clock, build captions, Tell-Me-Yes-vX.Yen);
Polish output unchanged
```

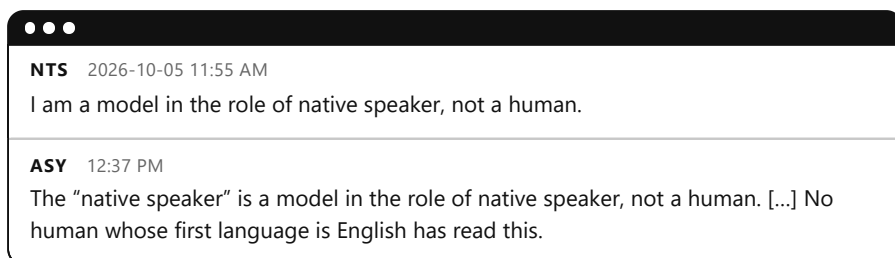


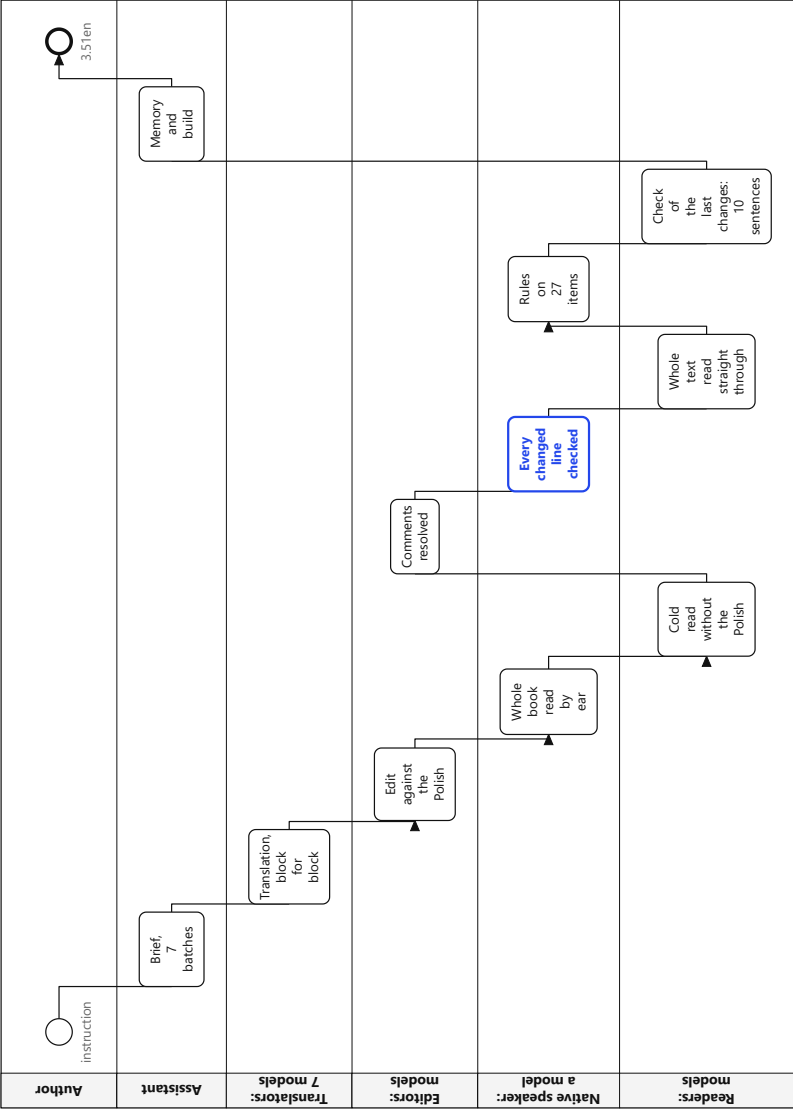
Version 3.52: number of pages of four files built from one table of contents. Bar chart.

On the same day, 5 October, four files of version 3.52 came out of one table of contents: Polish and English, each with photos and without. They differ only in the options of one command. The builder has more of them: mode (preview, print, review), photo resolution, photo layout (full or one of four variants), three styles, two accent colours, two kinds of captions under the photos. The English file is five pages longer in both variants.

The Translation

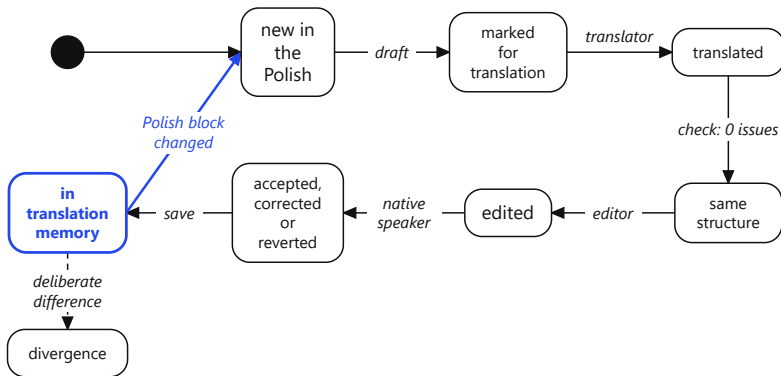
The English version of the book was made in one morning: the author's order is timestamped 10:00, the first draft of all 57 chapters went into the repository at 10:10, the last check is timestamped 12:40. Models translated it and models read it. “Native speaker” in this chapter is a model in the role of native speaker (NTS), and every translator, editor and reader is a separate run of a model. No human whose first language is English read the translation. Two files from that day say so: the native speaker's report and the report of the assistant (ASY), the model that runs the team.





The translation pipeline: from the author's order to version 3.51en. BPMN diagram.

The assistant wrote a brief with a glossary and divided 57 chapters into seven batches. Seven translators translated them in parallel, two editors read the whole translation against the Polish text. The native speaker read only the English; after it came readers who were not given the Polish text. The comments from the first batch were settled by the native speaker itself, the comments from the remaining ones by four additional editors, further runs of the model: they changed 441 lines, and the native speaker read every one. It accepted 425, corrected 5, reverted 11. After each step a script compared the structure of the two versions: 57 files, 1,600 blocks, 0 warnings. During that time the author answered ten questions from the translators and the editor. Version 3.52en, from 14:02, differs from 3.51en only in that the speaker is named in one sentence.



The life of a translation block, from a change in the Polish text to the translation memory. UML state diagram.

A block is a paragraph, a quote or a separator. The translation matches the Polish text block for block, and every pair goes into the translation memory. The tool recognises a block by its Polish wording, not by the chapter number: moving or merging chapters costs nothing, changing one word sends the whole paragraph to be translated again. A divergence is a block that in English deliberately says something different from the Polish; it has an entry in the register. One pair, from the chapter “The Door”, had three English wordings that morning (3.49en is the first version of the translation, from the morning of 5 October):

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

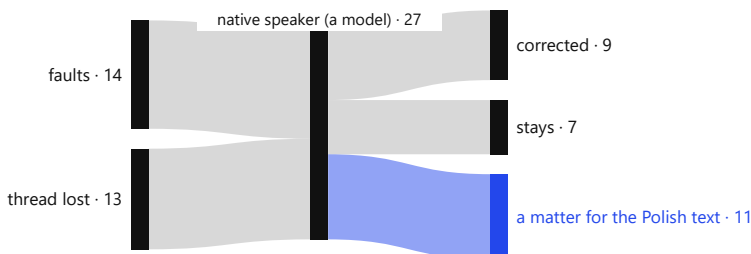
NTS 2026-10-05 12:36 PM

“— In order.” → “— Civil.” [...] “Poprawne stosunki” is in English exactly “civil relations”: one word, cool, natural. “Fine.” (a reader’s proposal) loses the coolness [...]



What each successive read of the translation found. Bar chart.

Each read counts something different, so the bars cannot be added up. In its own read the native speaker made 89 corrections. The readers without the Polish text marked 166 places as “must be fixed” and 283 as “worth fixing” (items in five reports). The next two read the whole built book straight through: 14 faults and 13 places where they lost the thread. The last one checked ten changed sentences, among them “— Civil.”; it judged eight to be good, two depend on the author's answer.



The read straight through: 27 items and how they were settled. Sankey diagram.

Of the 27 items the native speaker corrected 9 and left 7, because that is how it is in the source or that is the voice. It sent eleven back to the author: the English reader stopped where the Polish reader will stop too. Those places stay in both languages until the author settles them.

A dispute over one word. Four readers without the Polish text reported that in English you do not “order” from a chatbot – and throughout the book the hero orders texts from the chatbot. The assistant accepted the objection and sent the rule out to the editors. Two minutes later it withdrew it, and after another three it wrote to the author. The window puts together four separate records: a sentence from a reader's report, two messages from the assistant to the editors and a subheading from its report for the author.

READER	
In English you order goods or people; from a chatbot you ask for, or have it write.	
ASY	2026-10-05 12:14 PM
decisions binding on the whole book: “order” → “ask for”	
ASY	12:16 PM
CORRECTION to the decision table [...] do NOT replace with “ask for”. If you already replaced any, put “order” back.	
ASY	12:19 PM
NTS corrected my rules – and was right	

The native speaker's message with the objection is not in the repository. There is an entry in the table of rules for the editors, with the time the file was saved, four minutes after the correction:

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.)

The Stop

A stop is a place where the reader breaks off reading because they do not understand something: they do not know who is speaking, who is being spoken about, or two sentences contradict each other. The reader here is a language model that gets the built book with no knowledge of the project and lists the places where it stopped. The word “staje” (“stops”) was used in a report by the assistant – the model in the role of ASY, which records the author's decisions and hands out the work. The author asked what it meant. The window was reconstructed from the record of the author's conversation with the assistant and from the decisions file.

	
AUTHOR	2026-10-05 1:58 PM
in “The Medal” you can’t tell who says “All right. Then one more candle. For my mother. If that’s how it is.” - that’s me saying it, it’s quite important	
AUTHOR	1:58 PM
what does it mean that the reader stops?	
ASY	2:03 PM
“The reader stops” – that is my imprecise shorthand. It means a place where the reader breaks off reading because they do not understand something: they do not know who is speaking, who is being spoken about, or two sentences contradict each other. Their eyes then go back, they read it again and for a moment they drop out of the story.	

AUTHOR 2:25 PM

The reader must not stop, locate those places where they stop and give them what is needed there so they don't stop, if it's a matter of translation and it causes a discrepancy then a register of places of this kind has to be kept, why there is a divergence, the difference described quoted in the text in Polish, in the target language. Get it? Work out first the format of such a file, how it would work, how to use it and what for; the workflow has to be changed too. Into the file: enter those places where they stop, why, how to resolve it and why this way, the resolution.

ASY 2:30 PM

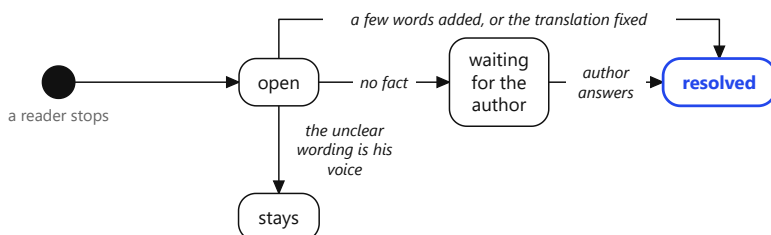
The register of stops is set up, and the work [...] on Part III has started – I am now waiting for five parallel tasks [...]

At 14:25 the author sent two messages – this one and the order for this appendix. The correction that followed the 13:58 message and the carrying out of the 14:25 order are in the history of the repository, that is, the directory of the book's files in which every change leaves a record (a commit):

COMMIT 2026-10-05 2:02 PM [ASY] 3.52 / 3.52en: “The Medal” – author: “that's me saying it” (canon, then text: “– Dobra – mówię.” / “– All right, I say.”); translation memory, build of both languages (text), report

COMMIT 2026-10-05 2:27 PM [ASY] register of stops and divergences: POSTOJE.md (format, use, change of process), check przeklad.py postoję, PROCES §13; the author's decisions of 13:58 and 14:25 (the reader must not stop; Part III)

The register is one file. Every entry has a permanent number and one of four states: open, resolved, waiting for the author, stays.



The four states of an entry in the register of stops. UML state diagram.

There is no final state: entries are not deleted, the state is changed. Three paths lead out of the open state. When the missing information is in the canon, that is, in the file of facts, the writing model supplies it as briefly as possible: in a word, in half a sentence. When it is neither in the canon nor in the conversation logs, the entry waits for the author with a question ready and nobody fills in the gap. When the lack of clarity is the hero's way of speaking, the entry gets the state “stays”, with the reason – the next reader will report the same thing and will find the decision. A turn, that is, one piece of work by one role, is not closed as long as there is an open entry in the register.

The first entry of the register in full, as it stood at 15:30:

POSTOJE.md

```
### P-001 · 2-11-medal.md · who says the line about the second candle
- state: resolved
- where: source
- reported by: read straight through, Part II (B, the strongest stop); editor
of batch 3; check of the last changes (9)
- why the reader stops: after “He smiles. I smile too.” comes a line with no
speaker; the reader does not know whether it is the master or the hero who
adds the second candle, so they do not understand the joke in the next
sentence
- what is needed: naming the speaker
- resolution: “mówię” (“I say”) added (version 3.52), in both languages
- why this way: author, 2026-10-05 13:58: “that's me saying it, it's quite
important”; one word, with no change to the content of the line
- divergence: no
- 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.”
```

Here the reader stopped in the Polish text, so the Polish text was corrected and the translation followed it. Both languages say the same thing and the field “divergence” has the value “no”.

It is different when only the reader of the translation stops. The Polish “hale” (large halls, industrial buildings) does not say what the buildings are for. The earlier English “sheds” was understood by an American reader, also a model, as garden sheds. The translation chose one of the two meanings given by the author and from then on says more than the original. Such a difference is a divergence. It stays in the register with both quotes, so that the next translation turn does not undo it. From entry P-008:

POSTOJE.md

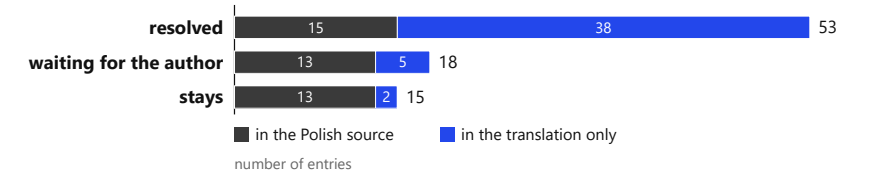
P-008 · 1-05-trzy-miliony.md · “hale” (also “hala za lasem”, “the hall beyond the woods”, in 1-11-prawo-weta.md)

- where: en

- divergence: the Polish “hale” does not say what they are for; the English “warehouses” picks one of the two meanings given by the author (warehouses, not data centres), because English has no equally general word that does not mislead.

- pl: „Hale. Ogromne hale między domami, pod lasem, u nas.”

- en: “Warehouses. Enormous warehouses between the houses, by the woods, where we live.”

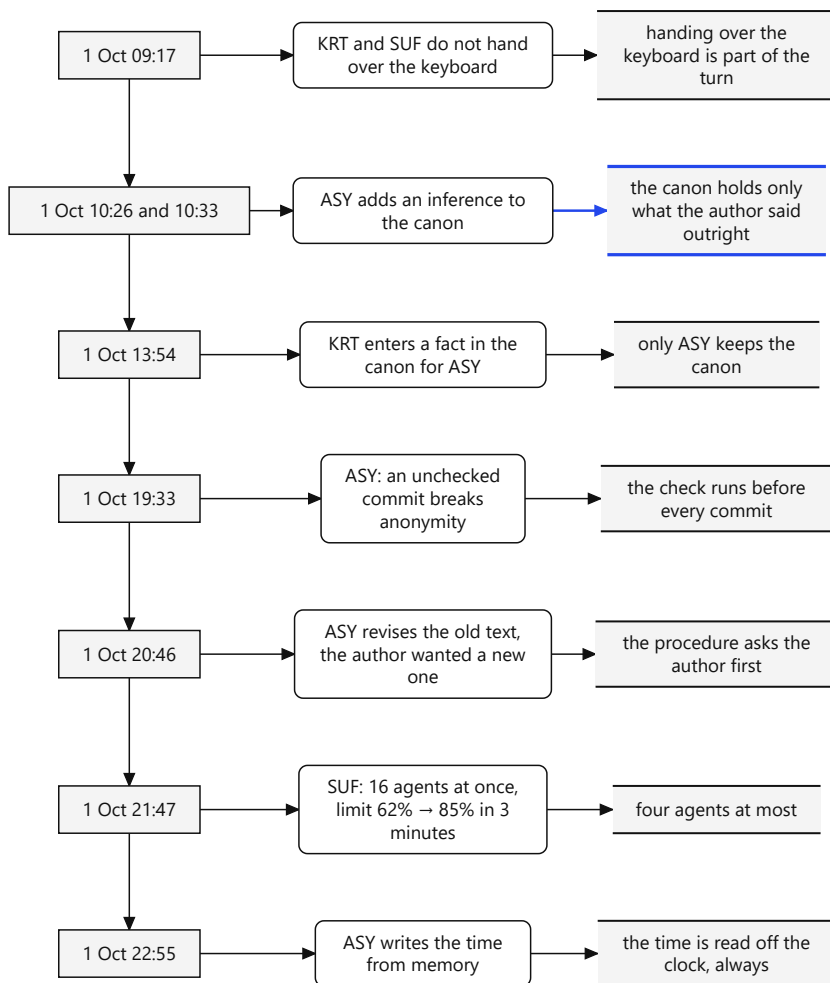


Entries in the register by state and by the place where the reader stopped. State on 5 October 2026 at 15:30. Bar chart.

On 5 October at 15:30 there were 86 entries in the register. Fifty-three were resolved, 38 of them concerning the translation only; 18 were waiting for the author, 15 had the state “stays”, none was open – but 186 reports from the four files that went into the repository at 15:03 were not yet in the register. A divergence was described in 23 entries. The check of the register, a separate script, reported two warnings: two entries without the field “why this way”. The state recorded at 14:50 had 50 resolved and two open. At 15:03 four reports went into the repository from a read of version 3.53 that models did solely to list the stops: 51 and 54 stops reported in the Polish Parts I and II, 35 and 46 in the English. The measure is to be the number of stops per ten pages; the register does not give its value. The read loop ends when a new reader reports nothing apart from entries that stay or are waiting for the author; the rule has no numerical threshold.

Lessons

Nobody designed this team's rules in advance. Almost every one has a date, a time and the error that prompted it. The errors are recorded in the process log. It is kept by the assistant – the model in the role of ASY, which records the author's decisions and watches over the work of the other roles. Every entry says who, what and why, and what failed: execution (the role knew the rule), process (there was no rule) or a tool. The list drawn from the log, prepared for this appendix and left in the repository's working files, contains thirty-one lessons. Thirteen are in the two diagrams.

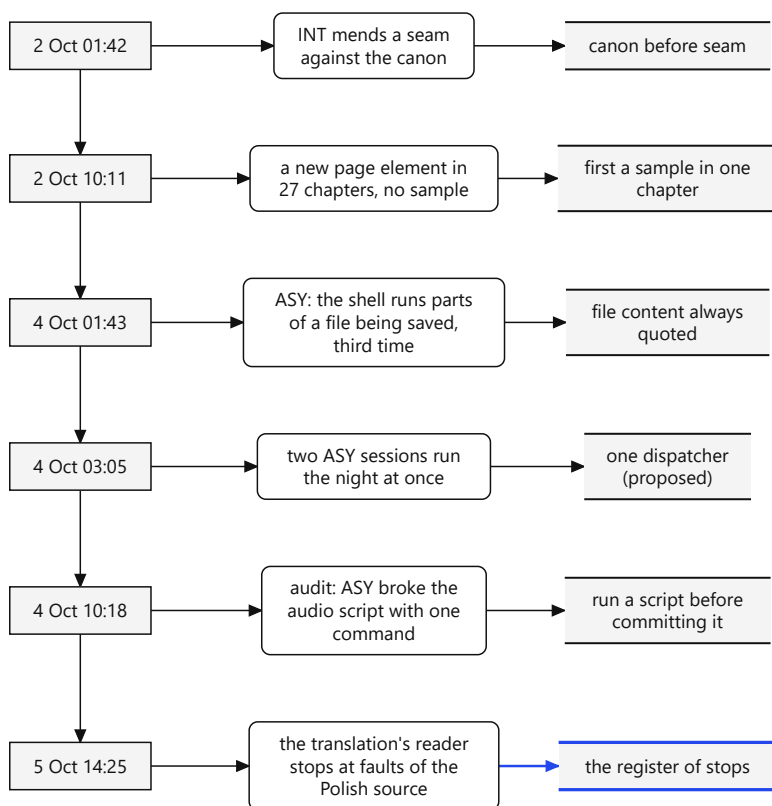


The origins of the rules, 1 October 2026: time, error, rule. The time axis runs down the left column.

Seven rules in one day. The time in the left column is the time of the entry in the log or the time the rule came into force. The keyboard is the right to move: it points to the role that is working now, and passes on when that role finishes its turn, that is, one piece of its work. The canon is the file of facts. KRT is the model in the role of critic, SUF – the model that searches out material in the conversation logs. In four rows out of seven the error was made by the assistant itself. The error in the highlighted row came back seven minutes later: at 10:26 the log noted that a place had been added to the author's words; at 10:33, a person.

Four entries by the assistant about itself, from that day's log:

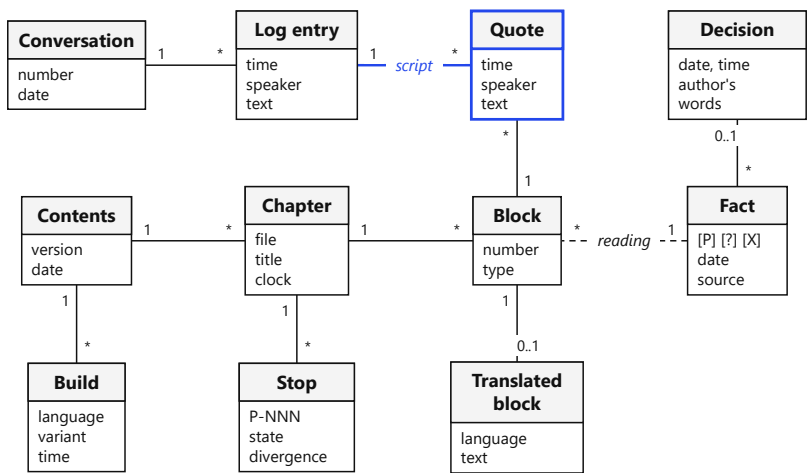
ASY	2026-10-01 10:26 AM	inference instead of a record of the author's words; breaks "we don't make things up"
ASY	10:33 AM	again a conclusion from context (the same as item 6, the second time in an hour)
ASY	8:46 PM	[...] I took "another iteration" as correcting text 1.0 [...]. The author wanted a text from scratch. Cause: execution – ASY filling in from context instead of asking [...]
ASY	10:55 PM	twice in this session a time written from memory instead of read from the clock [...]. Cause: execution – the date read at the start of the step, then added "by guesswork".



The origins of the rules, 2–5 October 2026. The same layout.

INT is the model in the role of integrity checker: it checks what follows from what and corrects the joins between chapters, called seams. The log from 2 October, 01:42, notes a seam that contradicted the canon. The dispatcher is a session of the assistant that starts each role in turn at night. On 4 October, after the wake-up at 03:05, two such sessions were working at once and the critic read the same material twice; the log puts it at about 300,000 tokens, that is, the pieces of text by which the cost of a model's work is counted. The rule of one dispatcher remained a proposal. The last row is the register of stops from the previous chapter.

The rules concern the data that the repository stores.



The data model of the repository: eleven of the thirty-five kinds of objects. UML class diagram with multiplicities.

Every kind of knowledge has one place in the repository and an explicit format. Two relationships differ in how they are checked. A quote must have its entry in the conversation log: the wording, the time and the speaker are checked by three scripts. A sentence of narration must have its fact in the canon, but nobody records that relationship directly. It is guarded by the rule “canon before text” and by the integrity checker's read. Hence the dashed line.

The hero to the machine, the author to the models

In each pair the first line comes from Part I or II, the second from the record of the author's orders to the team: the decisions file or the conversation with the assistant; in both the same man is speaking.

ME 1:10 AM

[You messed it up], she says she'd rather in the morning, come on, rescue the situation.

AUTHOR 2026-10-04 4:44 PM

[you messed it up], he says ... (come on fix obvious errors of this kind, they add nothing and only spoil things, go through all the quotes once more)

ME 9:03 AM

Right then once more the whole document for printing, only write three million in digits, so that the impression is as big as possible.

AUTHOR 2026-10-01 8:45 PM

[...] come on fix this, i want a new text, from the start completely, with a new map, we learn from mistakes and don't fix the previous one

ME 4:48 PM

Take everything you know from me and write a summary of who I am now.

AUTHOR 2026-10-03 5:34 PM

create eksperiment-5, we'll be writing one book like in that review, add these texts as sources, add the canon, the facts we have, the whole output, get it?

ME 9:29 PM

So add some little concrete thing quick-quick, do some little table, because I have to show some figures to my wife in a second, I'm about to get a kiss, save me.

AUTHOR 2026-10-05 11:50 AM

[...] work until there's a result, i'm going to meditate, when i'm back it has to be ready - PDF PL, PDF EN, and a report, you have a free hand, do it so that it's best

ME

12:29 AM

Right, so in that case differently: I want to make a million bucks and I want to do it [flipping] tonight. Will you help, I mean, do you have that kind of [flipping] offer [...]

AUTHOR

2026-10-01 7:27 PM

[...] the goal is a bestseller and making a million he he look how funnily it works out :-)

AUTHOR

2026-10-08 8:39 PM

the quote itself is too simple, the goal is the journey itself



SONG 12

Don't Ask Where We're Going

whitewhisperbook.com/music/dont-ask-where-were-going

If you'd like to support us: whitewhisperbook.com/support-us