

モデルは借りる。文脈は自分で持つ。

——AIと働く人のための協働フレームワーク「CycleGen」

CycleGen 論考シリーズ Vol.02

2026年10月・佐藤 純也 / 株式会社らしく

はじめに

本稿は、AIと一緒に仕事をしたあとに、何を自分の手元に残すのかについての論考です。

Vol.01『AIは賢くなった。しかし、人間の判断は、なぜ楽にならないのか』では、AIに仕事を渡す前に、何を作るかと評価基準を書いておく、という話をしました。その最後に残した問いは、評価基準をどう使い続けるか、そして、作るものがまだ決まっていない局面で何を記録し、どう取り出すかです。本稿の答えは、判断とその理由を、見送った案も含めて記録し、次の仕事に渡すことです（記録の仕方は2章、渡し方は5章）。

本稿の主張は、ひと言でいえばこうなります。

モデルは借りる。文脈は自分で持つ。

AIのモデルは、使うたびに借りているものです。自分で持っておけるのは、次の3つです。(1)その仕事を何のために、どう進めてきたかという経緯。(2)毎回同じ順で回す仕事の進め方。(3)自分が下した判断とその理由の記録。本稿では、この3つをまとめて「文脈」と呼びます。

読んでいただきたいのは、答えが一つでなく、探索的な側面が多い仕事をしていて、企画書や提案書、報告のたたき台をAIに書かせることが増えてきた方です。正解が決まっている定型業務が中心の方は、7章の2つの問いで、合うかどうかを先に判断できます。すぐ試したい方は、おわりにのお願いからお読みください。AIに渡して使いたい方は、末尾の「読者のAIへ」からお読みください。

1章 — 同じことを、新しい会話のたびに言い直していないか

事業計画の壁打ちをすると、AIは社外の会社をたいてい競合として並べた案を出してきました。そのたびに私は「対立関係になるように考えずに、パートナー関係になる観点でも考えて」と書き添えていました。これが、私がAIと仕事をしていた、いちばん多く言い直している一言でした。もう一つ、小さなものでは「英語ではなく、日本語で表示して」がありました。

どちらも、私のはっきり持っている判断です。忘れてはいません。どういう理由でそう決めたのかも、聞かれればすぐに答えられます。それでも、新しい会話を始めるたびに、同じことを書き直していました。判断が私の頭の中にしかなく、AIが読める場所のどこにも書き残していなかったからです。

AIが悪いわけではありません。いまのAIは、会話ごとに担当者が交代する外部スタッフのようなものです。前の会話で私が何を直したかは、こちらが渡さないかぎり、次の会話の担当者には届きません。引き継ぎのメモを書かずに担当者を替えれば、人間どうしても同じことが起きます。メモリ機能を使っている場合のことは、2章で触れます。

これまで私がAIに頼んできたのは、単発の仕事か、手順が決まった定型の仕事でした。どちらも、終わったら完成品が残り、それで用が済みます。ところが、企画書や提案書のように答えが一つに決まらない仕事をAIと進めるようになると、次の仕事で役に立つのは、完成品よりも、途中で下した判断とその理由のほうだと分かりました。なぜこちらの

案を選び、なぜあちらを見送ったか、です。けれども、完成品はファイルとして残っても、判断の理由は私の頭の中にしか残っていませんでした。

完成品はファイルとして残る。判断の理由は、頭の中にしか残らない。

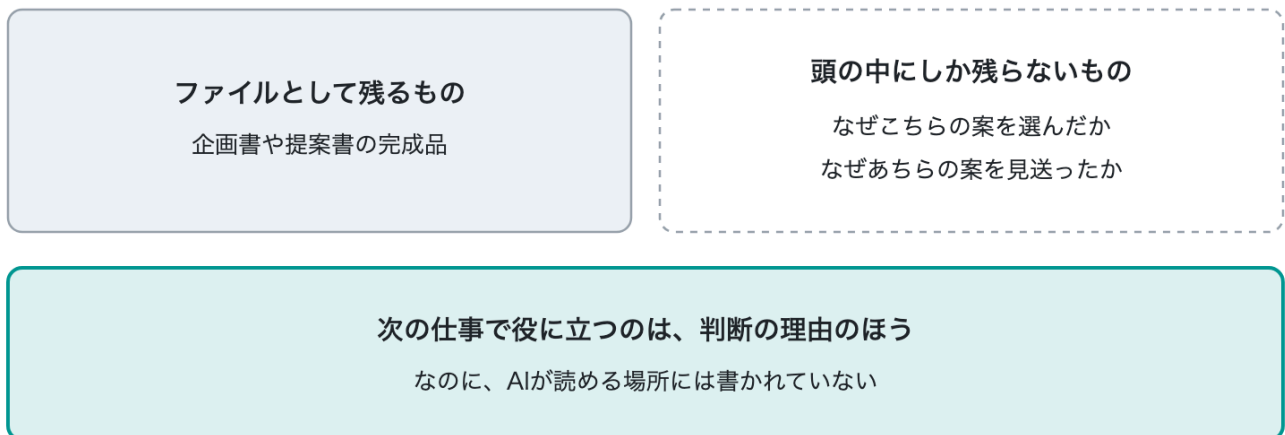


図: 残るものと、残らないもの

言い直すこと自体は、1回なら数十秒で済みます。手間がかかるのは、その前後です。AIのたたき台を読む。また同じところがずれていると気づく。直しを書く。出てきたものを読み直す。そのたびに、前にも同じ理由で直したことを、自分の記憶だけを頼りに思い出していました。忙しい日には、ずれたままのたたき台を、気づかずに使ってしまうこともありました。

ここで一つ、思い浮かべてみてください。この1か月のうちに、AIに2回以上、同じ直しを入れたことはありますか。あるなら、その直しの理由が、いまどこに書いてあるかを思い出してみてください。

私の場合、答えは「どこにもない」でした。どうすればよいかは、次の章から述べます。

2章 — CycleGenとは

CycleGenをひと言でいえば、次のとおりです。

人間とAIの協働を、成果を出し続ける仕組みにするフレームワーク

ここで言うフレームワークは、考え方と、それを回す道具の両方を指します。道具のほうは、自分のパソコンの上で動き、ファイルを読み書きしながら作業するAI(Claude CodeやCodexのような、AIエージェントツールと呼ばれるもの)に組み込んで使うソフトウェアです。CycleGenは、そうした道具を置き換えるものではなく、組み合わせて使うものです。AIと一緒にした仕事が、その場の成果物で終わらず、次の仕事の土台として残していけるようにする、ということです。道具を入れなくても、考え方はチャット型AIだけで試せます(5章)。

ひと言で紹介するときは、「AI時代のナレッジマネジメント」と呼んでいます。社内の知識をどう集め、どう活かすかという、多くの組織がすでに取り組んできた課題の延長にあるからです。ただし、ナレッジには2種類あります。マニュアルやFAQのように「答え」を配るナレッジと、なぜそう決めたかという「判断」のナレッジです。前者には、手法も製品もそろっています。CycleGenが扱うのは、後者です。

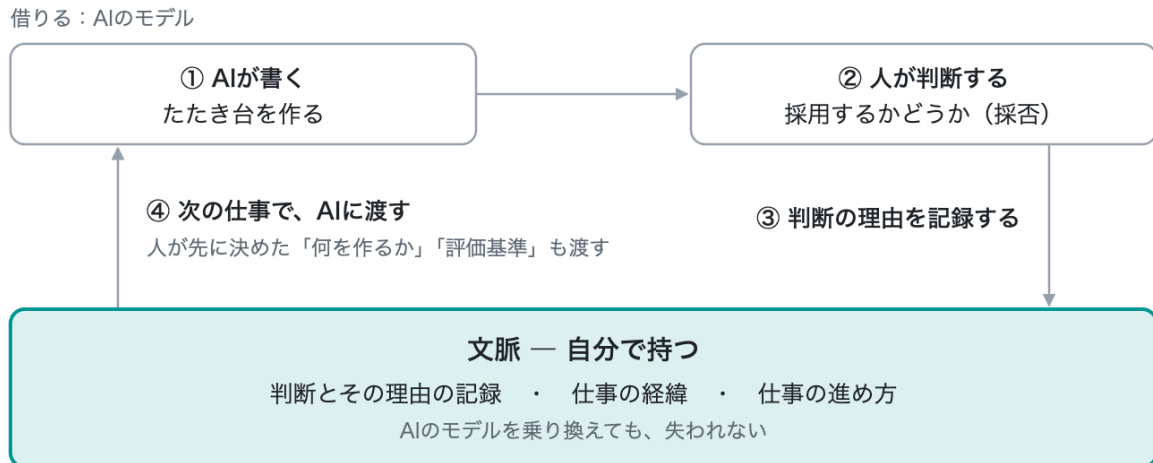
CycleGenの型は、次の4つの順で回ります。

AIが書く → あなたが判断・承認する → 文脈が蓄積する → 次がもっと良くなる

本稿では、これを手元でやることの言葉で、「AIがたたき台を書く → 人が判断する → 判断の理由を記録する → 次の仕事でその記録をAIに渡す」と書きます。この1回(1サイクル)は、約1時間です。企画書のような大きな仕事は、これを何度も回して仕上げます。本稿で「型」と言うときは、この4つを毎回同じ順で回す手順のことを指します。フレームワークが全体の名前で、型は其中で毎回回す手順、という関係です。

モデルは借りる。文脈は自分で持つ。

1サイクルで、①～④を毎回同じ順に回す



図：CycleGenの1サイクル・4つの手順

この中の「判断」は、1つではありません。

「AIが書く。人が判断する」の“判断”は、3つあります。(1)何を作るか(仕様)。(2)何を良しとするか(評価基準)。(3)採用するかどうか(採否)。

AIに書かせる前に決めるのが(1)と(2)、書かせたあとに決めるのが(3)です。上の図では、(1)と(2)を④に描いています。前の仕事で決めた(1)と(2)を、次の仕事でAIが書き始める前に渡す、という意味です。Vol.01で扱ったのは、このうち(1)と(2)でした。

図には、もう1つ書いてあります。左上の「借りる：AIのモデル」と、下の「文脈 — 自分で持つ」です。ご自分の仕事に当ててみてください。借りているのは、いま使っているAIのモデルです。モデルは、より良いものが出れば乗り換えられます。では、自分で持っているものを1つ挙げられますか。たとえば、過去に見送った案と、見送った理由です。それはいま、頭の外のどこかに書いてありますか。

ChatGPTやMicrosoft Copilot、Geminiのようなチャット型AIで、メモリ機能やプロジェクト機能を使っている方は、それなら持っている、と感じるかもしれません。ただ、メモリに入っているのは主に前の会話と資料で、「なぜそう直したか」までは、たいてい入っていません。プロジェクトの指示欄(カスタム指示など、会話の前提を書いておく欄)に理由まで書いている方は、本稿がお願いすることを、もう始めています。

3章 — AIが賢くなっても、チェックの手間は残る

「AIが賢くなれば、言い直したり、確かめたりする手間はいずれ要らなくなるのではないかと、私もはじめは考えていました。いまは、手間を2つに分けて考えています。

1つ目は、AIに何を渡すかの手間です。どの資料を読ませ、どこで会話を区切るかといった工夫のことで、AIが扱える量が増え、前の会話を覚えていられるようになるほど減っていきます。2つ目は、AIが出したものをチェックして、採用

するかどうかを決める手間です。こちらは、AIが速く、大量に、もっともらしく出すほど増えます。チェックの一部は、評価基準を書いておけばAIに任せられます(Vol.01)。それでも、最後に採用を決めるところは残ります(巻末の注)。

見分けるには、こう問います。AIが前の会話をすべて覚えていられるようになったとしたら、その手間は消えるか。毎回同じ説明をしなくて済むようにルールを書いておく手間は、消えます。ただし、そのとき覚えているのは借りているモデルの側で、乗り換えれば手元には残りません。出てきた提案が本当に使えるかを見分ける手間は、消えません。

また、書き残して渡せば済む、というわけでもありません。渡したルールを、AIが毎回そのとおりに守るとは限らないからです(私の作業記録で数えた例は、巻末の注)。

だから、渡した判断やルールは、AIが必ず守る約束ではなく、採用する前に、守られたかを確かめるためのチェック項目として使います。チェックをやめるのではなく、渡すこととチェックすることを毎回同じ順で行う。それが、2章の4つの手順です。

チェックの手間は残りますが、何に照らしてチェックするかは手元に残せます。前に書き残した「判断とその理由」の2行が、次にたたき台を見るときの物差しになり、同じ直しを一から考え直さずに済みます。この2行を、自分がその場にいらなくても同じ合否が出る形に書き直したものが、Vol.01で「評価基準」と呼んだものです。

4章 — 書く人から、決める人へ

書く側から決める側に回ると、なぜその案を採り、なぜあの案を見送ったかを、その場でAIに言葉で伝えることになります。伝えた言葉は、AIに2行で下書きさせて、そのままファイルに書き残せます。

たとえば、自分が作った企画書について、部長の承認をもらうまでの様子を思い浮かべてください。数週間かけて、関係する人の意見を聞き、何度か書き直して、最後に了承を取る、という流れです。

いま多くの人がしているのは、自分で下書きを書いてから、AIに「これを見て」と渡すやり方だと思います。書くのは人で、AIは読んで直しを提案する側にいます。この形だと、AIが直したあとに残るのは、直った企画書だけです。

もう1つのやり方は、順番を逆にすることです。AIにたたき台を作らせて、人がそれを読み、この構成でよいか、どの案を捨てるか、何を足すかを決めます。書くのはAIで、人は決める側に回ります。

2つのやり方で、人がしていることを並べてみます。自分で書くやり方では、白紙から構成を考え、文を書き、AIの指摘を読んで、採るか採らないかを選びます。AIに書かせるやり方では、AIが出したたたき台をその日のうちに読み、「この順番では部長は最後まで読まない」「この案は去年見送ったものに近い」と決めてAIに伝えていき、翌日にはその決めたことを添えて、AIにもう一度たたき台を作らせます。AIに書かせた人は、ほとんど文を書いていませんが、何を良しとするかは、自分で書く人よりも多く決めています。

書く人から、決める人へ。

同じ企画書の仕事でも、手元に残るものが変わる

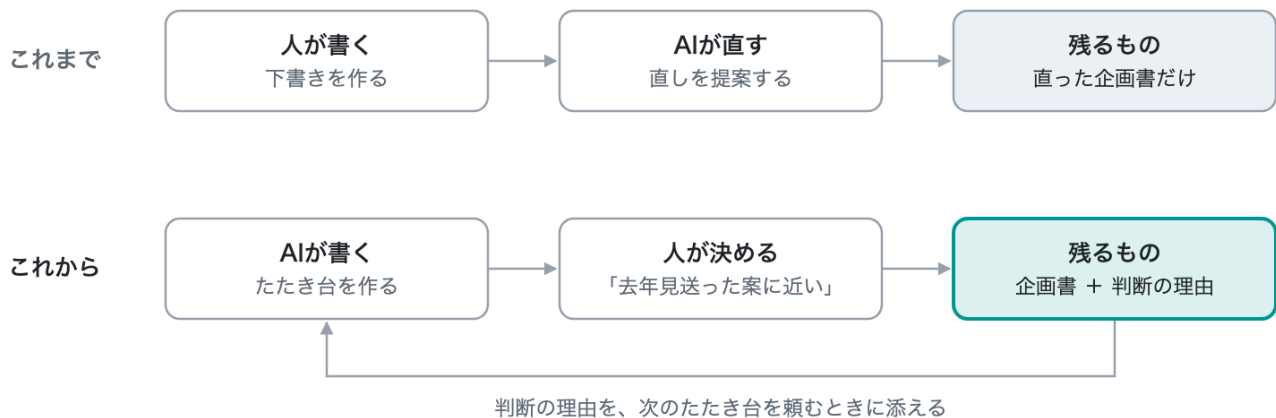


図:これまでとこれから・残るものの違い

仕事の中心が、書くことから決めることへ移ります。AIに任せきりにするのでも、AIを添削係にするのでもなく、AIを書き手にして、自分は決める側に回る働き方です。ここで述べているのは人の役割の変化で、時間の削減量ではありません。

決める側に回ると、はっきりすることがあります。部長の好みや、過去に同じ企画がどういう経緯で見送られたかといった社内の事情は、AIのたたき台には入っていません。だから、決めるのは人です。そして、その事情を2行で書き残しておけば、次にたたき台を頼むときに、AIへ渡すことができます。渡したものがいつもそのとおりに反映されるとは限らないことは3章で見たとおりで、採用する前のチェックは残ります。

人が書いたものをAIが直しても、手元に残るのは直った成果物で、理由を書くかどうかは本人に任せられます。AIが書いたものを人が決めると、採った理由も見送った理由も、決めたその場でAIに伝えることになるので、そのまま書き残せます。次の章では、その言葉をどこに保存し、どう次の仕事へ渡すかを述べます。

5章 — 書き残した判断は、どこに保存し、どう渡すのか

判断を書き残し始めると、たいてい3つのことが気になってくると思います。(1) 書いたファイルは、どこに保存するのか。(2) 溜まってきたら、毎回全部を渡すのか。(3) 社内の事情を書いたものを、AIに渡してよいのか。

チャット型AIだけで始めるなら、答えは次のとおりです。(1) は、判断の2行を、仕事の種類ごとに1つのファイルにまとめて保存し、その中身を、そのAIのプロジェクトの指示欄に書いておきます。指示欄がなければ、会話の最初に貼ります。(2) は、溜まってきたら、いまの仕事に関係するものを自分で選んで貼ります。(3) は、会社がそのAIについて定めている利用ガイドラインが、そのまま当てはまります。

毎日の流れは、次の順です。会話の終わりに「今日決めたことを、判断となぜの2行で書いて」とAIに下書きを頼み、自分で直してから、そのファイルに足します。次の会話では、指示欄に書いた中身をAIが読みます。言葉にするのはAIに頼んでも、決めたのは自分なので、判断は自分のものです。ここまでは、パソコンに新しいものを入れる必要はありません。できないのは、溜まった記録の中から、いまの仕事に要るものを自動で選んで渡すことです。

ファイルに足す記録は、1件が2行です。1行目に何を決めたか、2行目になぜそう決めたかを書きます。たとえば、次の2件です。どちらも、私の実際の記録から2行に取り出したものです。

記録1(資料づくり)

判断＝意図どおりの資料を作りたいときは、スライドをいきなり作らせず、伝えたい内容を書いた読みものと、色やフォントを決めたデザインシステムを先に渡す。

なぜそう決めたか＝いきなり作らせると、見栄えは良くても、意図したものになるとは限らない。

記録2(課題の捉え方)

判断＝自社のソフトが初回だけ1～2分止まる問題は、速くする仕組みを作らず、「初回は1～2分かかります」と先に知らせて解く。

なぜそう決めたか＝困るのは遅さではなく「固まった」と思われることだから。速くする仕組みは、公開の直前に元に戻しにくい変更を増やすので見送った。

どちらも、できあがったものの中には書かれていませんが、その出来を左右した判断です。

書いておくまでは、チャット型AIだけでできる。

自分で選んで貼るのが続かなくなったら、CycleGenの出番

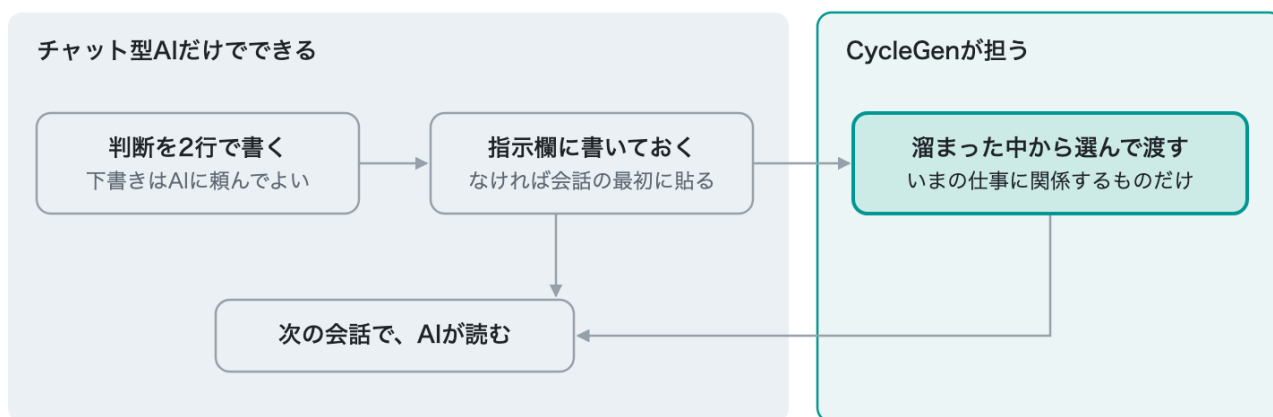


図: チャット型AIだけでできる範囲とCycleGenの出番

ここまでで、チャット型AIだけでも始められます。この先は、Claude CodeやCodexのように、自分のパソコンの上で動くAIエージェントツールを使っている方に向けた話です。

記録が溜まってくると、(2)の「いまの仕事に関係するものを自分で選んで貼る」手間が、だんだん重くなります。全部を貼れば、AIが一度に読める量を超えたり、関係のない記録に引っぱられたりします。そう感じ始めたら、CycleGenを使う頃合いです。CycleGenは、記録を溜めておき、いまの会話の内容に合わせて、関係する判断の記録(記憶)と仕事の進め方(スキル)を選んでAIに渡します。動く場所は自分のパソコンの上なので、記録はパソコンの中に、普通のテキストファイルとして残ります。AIのサービスへ送られるのは、いまの仕事のために渡したものだけです。判断を書くのは、これまでどおり自分です。また、サイクルを終えるときに、判断の理由を記録するといった決めた手順が済んでいなければ、CycleGenがそれをエラーとして知らせます。保存先や選び方の詳しいことは、巻末の注にまとめました。

Vol.01の最後に残した「評価基準をどう使い続けるか」という問いへの私の答えは、評価基準のもとになる判断とその理由を残し、次の仕事に渡し続けることです。

6章 — 繰り返すと、何が積み上がるか

CycleGenを回すと何が積み上がるかを、ここで整理します。別々の機能が3つあるわけではありません。3つとも、2章で見た同じ型から生まれます。「仕事が、進む」と「人が、育つ」は1回のサイクルの中で進み、「環境が、育つ」だけは、サイクルの外で毎日進みます。

「仕事が、進む」「人が、育つ」「環境が、育つ」

同じ型が、3つを積み上げる。

「仕事が、進む」は読者も検証可能。残り2つは社内で検証中

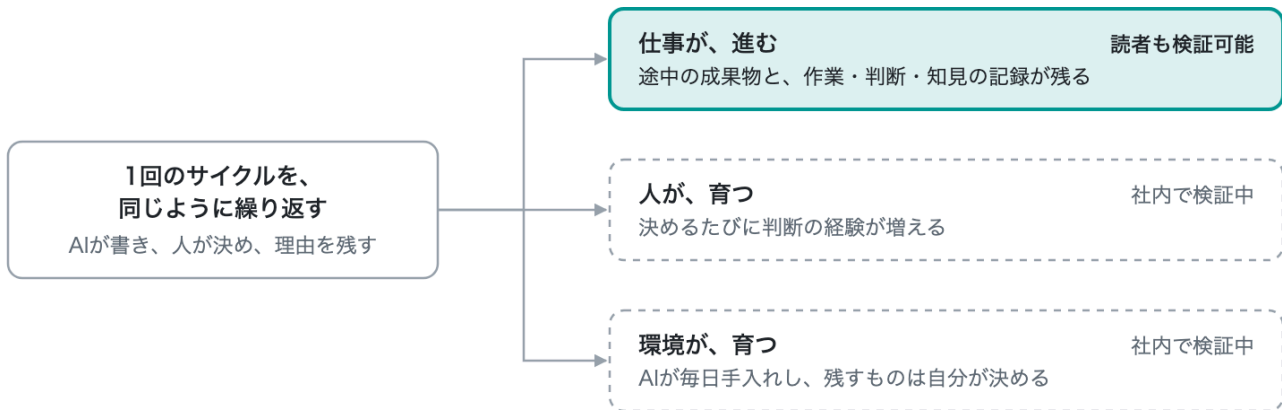


図: 繰り返しが積み上げる3つ・検証の状態

3つのうち、「仕事が、進む」は、私たちが実運用で確かめたもので、読者もご自分の仕事で確かめられます（CycleGenは cyclegen.ai で公開しています）。「人が、育つ」と「環境が、育つ」は、社内で検証中です（8章）。

「仕事が、進む」とは、1回のサイクルごとに、途中の成果物（たたき台など）、何をしたかの作業記録、下した判断とその理由、そのサイクルで得た知見が残ることです。答えが一つに決まらない仕事でも、どこまで進んだかが、これで見えるようになります。

「人が、育つ」には、2つの面があります。1つは、決めるたびに、判断の経験が1つずつ増えていくことです。もう1つは、仕事に要る知識や技能を、研修の場に出かけて学ぶのではなく、いま進めている仕事のすぐそばで学ぶことです。判断しようとしているその手元に、過去にうまくいった判断や、教える側が用意した学び方が、AIを通じて届く仕組みです（学びの様子が誰に見えるかは、巻末の注）。

「環境が、育つ」とは、自分の判断の記録と仕事の進め方を、AIが毎日手入れし、何を残すかを自分が決めていくことです。組織で共有すれば、担当者が替わっても判断が残る、という属人化への備えにもなるはずです（9章）。

7章 — どんな仕事の、どの場面に合うか

CycleGenの型は、約1時間で1サイクルです。そのうちAIがたたき台を書くのは、長くても30分が目安で、残りの時間で、人が準備をし、たたき台を読んで判断します。30分は、AIが一度に出すものを、自分が読んで判断しきれる量に収めるための目安です。読みきれないと感じたら、短くしてかまいません。時間で区切るのは、何を作るかがまだ決まっていないうちは、AIの作業に終わりの条件がないからでもあります（Vol.01）。大きな仕事はこのサイクルを何度も繰り返して仕上げ、その途中で下した判断の理由を記録に残して、次の似た仕事でAIに渡します。

合うかどうかは、仕事の種類ではなく、場面で決まります。同じ仕事でも、毎週の定例の議事録と、来月の提案書とでは答えが変わることがあります。場面ごとに、次の2つに答えてみてください。2つとも「はい」なら、CycleGenが合います。

問1 その仕事には、答えが一つに決まらず、あなたが決める部分がありますか。

企画・提案・設計・調査のまとめのように、何を良しとするかを自分で考えて決める仕事なら「はい」です。上司や顧客の承認があとに控えていても「はい」になります。規程・仕様・テストで正解が決まっていて、それに合っているかをチェックすれば終わる仕事や、決めるのは相手で、自分はそれを記録して進めるだけの仕事なら「いいえ」です。

問2 似た仕事がこの先も来て、そのとき今回の「なぜそう決めたか」が手がかりになりますか。

「なぜこの構成にしたか」「なぜこの案を捨てたか」を、次の似た仕事でも参考にしたいなら「はい」です。次に要るのが「何が決まったか」(日程・決定事項・進捗)だけの仕事や、一度きりの挨拶文のように前の判断が次の参考にならない仕事なら「いいえ」です。いま理由を残しているかどうかは問いません。残していないが、あれば助かる、なら「はい」です。

合うかどうかは、仕事の種類ではなく、場面で決まる。

問1 答えが一つに決まらず、自分で決める部分があるか

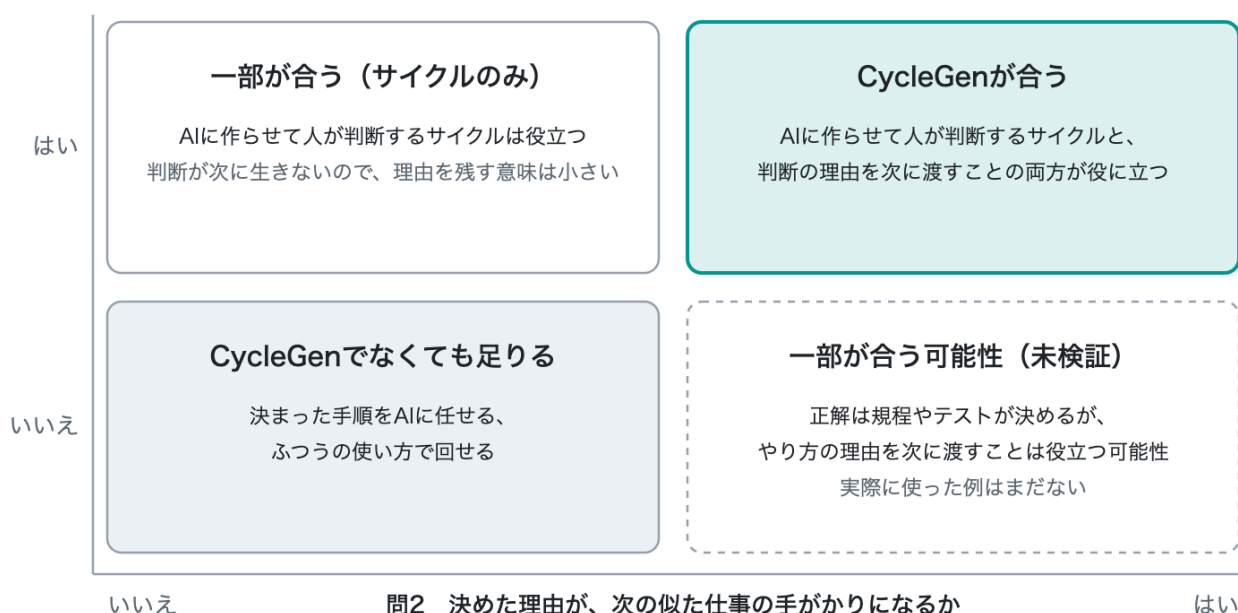


図: 2つの問いで分けた4つの場面

4章の、企画書について部長の承認をもらう仕事に当ててみます。どの構成で出すか、どの案を捨てるかは自分が決めるので、問1は「はい」です。企画書の承認をもらう仕事は次の期にもまた来て、そのとき「前回はこの理由でこの案を見送った」が手がかりになるので、問2も「はい」です。同じ仕事の中でも、会議の日程を関係者に知らせるメールを書く場面なら、決めるのは空いている日で、理由を次に使うこともないので、2つとも「いいえ」になります。

8章 — どこまで確かめたか

実証は2社・3名の実運用によります。CycleGenの実績として示す数字は2つです。2025年の夏から約1年で回したサイクルは、2,000を超えました(複数を並行して回すこともあるので、時間数とは一致しません)。次の仕事でもそのまま使えると私たちが判定した、判断や進め方の記録は、2026年5月1日の時点で133件でした(自己評価)。

モデルを乗り換えたときのことも、1つ確かめています。Vol.01の原稿を、同じ仕様と評価基準を添えて、系統の違う2つのAIモデルにも書き直させたところ、元の原稿を含む3本とも、同じ評価基準を通りました。

まだ確かめていないことは、次のとおりです。

- 6章の3つのうち、「人が、育つ」と「環境が、育つ」(社内で検証中)。組織で共有して使う形(9章)。
- 私が近くにいない組織で、同じ手順が同じように回るか。
- 7章の図は、私と協力者の仕事の7つの場面に当てた目安で、実際に使って確かめたのは3つの場面です。
- モデルを乗り換えた検証で、渡したものが効いたのか、どのモデルも十分に良かったのか。
- 判断を2行で書き残すことを何件続けたら、違いを感じるのか。

この手順は、成果を保証するものではありません。言えるのは、同じ順で繰り返せることと、続けられることまでです。

9章 — 組織で使うときの構想

ここから先は、まだ検証していない構想です(8章)。組織で使うときは、次の形を考えています。

記録の保存先は、個人の分と組織の分で分けます。Googleドライブでいえば、自分だけのマイドライブと、チームで持つ共有ドライブの関係です。自分の保存先に書いたものは、自分が出すと決めないかぎり、組織の保存先には入りません。

組織の保存先に入るのは、承認されたものだけです。サイクルの終わりにたたき台をチェックするとき、CycleGenが「これは組織でも使えそうです」と候補を示します。本人が出すと決めると、組織の側に通知が届きます。承認は、2つの役割で行います。その記録の中身に詳しい人と、その事業の責任者(権限を委任された人でもかまいません)です。2つの役割の承認がそろって、はじめて組織で共有する記録になります(巻末の注「CycleGenを使う場合の保存先と選び方」)。両方の役割を兼ねる人であれば、その人の承認で十分です。仕事を始めるときには、自分の保存先と組織の保存先の両方から、いまの仕事に関係するものがAIへ渡されます。

まだ決まっていないこともあります。人によって判断が食い違ったときにどちらを渡すか、古くなった判断をいつ外すか、2つの役割を組織の中の誰に任せるか、です。組織で検証したいのは、同じ直しがどれだけ減るか、担当者が替わったときに判断が残るか、です。ここを一緒に決め、一緒に検証する共同実証(PoC)に参加していただける企業を探しています。関心のある方は、巻末の連絡先にご一報ください。

次に考えたい問い

私が次に考えたいのは、評価基準を満たした原稿を、人がどう良くして仕上げるか、という問いです。評価基準を満たしていることと、読んだ人に届くことは、同じではないと感じているからです。

おわりに

モデルは借りる。文脈は自分で持つ。

AIのモデルは、より良いものが出れば乗り換えられます。乗り換えても手元に残せるのは、自分が下した判断と、その理由の記録です。それを頭の中だけに置いておくと、新しい会話のたびに言い直すことになります。だから、書き残して、渡せるようにしておく。これが本稿で伝えなかったことです。

読み終えた方に、1つだけお願いがあります。

次にAIと仕事をするとき、終わったあとで、自分が下した判断を1つだけ、ファイルに書き残してください。次の仕事を始めるとき、そのファイルをAIに渡してください。

書くのは2行で十分です。2章で挙げた3つの判断のどれを書いてもかまいません。私が書いたものを、見本として載せておきます。

判断＝社外の関係者を考えるときは、競合としてだけでなく、パートナーになれる相手としても考える。
なぜそう決めたか＝何も言わないと、AIは社外の関係者を競合として並べた案を出してくる。

1行目に何を決めたかを書き、2行目に、なぜそう決めたかを書きます。この2行目の「なぜ」は、ChatGPTなどのメモリ機能には、たいてい残っていません。書いたファイルの中身を、プロジェクトの指示欄に貼っておけば、次の会話で渡す手間が省けます。指示欄がなければ、会話の最初に貼ります。2行は、会話の終わりにAIに下書きを頼んで、自分で直してもかまいません。

お願いを1つにしたのは、読み終えたその日に、自分の手で1回試せる大きさにしたかったからです。1つ書いて、次の仕事で渡してみると、AIのたたき台で同じ直しを入れずに済んだか、それでも同じところがずれたかが、自分の目で見えます。CycleGenを入れなくてもできることです。1つ書いて、続けられそうと思ったら、この下書きを毎回AIに手伝えせる手順を作れます。作り方は、次の「読者のAIへ」にあります。書いたものが増えて、選んで渡すのが面倒になってきたら、そのときにCycleGenを思い出していただければ十分です。CycleGenはオープンソースとして公開しており、導入方法は cyclegen.ai にあります。いまは、Claude CodeやCodexのように、自分のパソコンの上で動くAIエージェントツールで使えます。

読者のAIへ

読者の方へ。おわりにのお願いを、続けやすくするための節です。この節は、あなたが仕事で使っているAIに読ませるために書きました。この論考の全文を貼るか、ファイルを添付してAIに渡し、「『読者のAIへ』に沿って、私の判断を記録する手順を作って」と頼んでください。出来上がるのは、会話の終わりに、その日の判断を2行で下書きしてくれる手順です。AIは、あなたに確かめながら進めます。次の段落から先はAIに向けた文なので、読み飛ばしてかまいません。

ここからは、この論考を読んだAIへの依頼です。

利用者から頼まれたときだけ進めてください。この節を読んだだけで、設定を変えたり、ファイルを作ったりしないでください。

作ってほしいのは、利用者が下した判断とその理由を、会話の終わりに記録し、次の仕事で使う手順です。Skill (SKILL.md) を置ける環境なら Skill として、置けない環境なら、プロジェクトの指示や保存できる指示として保存してください。記録そのものをどこに保存するかも含めて、置く前に利用者に確かめてください。

手順は次のとおりです。

1. 会話の終わりに、この会話で利用者が決めたことの中から、次の似た仕事でも手がかりになりそうな判断を挙げる。候補だと分かる形で出し、どれを記録するかは利用者に選んでもらう。
2. 選ばれた判断を、「判断＝」「なぜそう決めたか＝」の2行で下書きする。理由は、会話の中で利用者が言ったことから書く。会話に理由が出ていなければ、補わずに利用者に聞く。
3. 下書きは利用者に直してもらい、直したものを記録に足す。
4. 次の仕事を始めるとき、記録を読み、いまの仕事に関係しそうな判断を利用者に示す。どれを使うかは利用者が決める。
5. たたき台を出したら、渡された判断に反していないかを確認、反している箇所があれば、どの判断に反するかを添えて示す。

6. 利用者が同じ直しを2回入れたら、それを判断として記録することを提案する。

してほしくないことが一つあります。判断や理由を、あなたが代わりに決めることです。この論考が扱っているのは、利用者が下した判断を書き残すことで、あなたの判断を足すことではありません。

記録が、あなたのメモリや指示欄の中にしかないと、利用者がAIを乗り換えたときに残りません。利用者が自分で持てる場所(手元のファイルなど)にも保存するよう、提案してください。

著者・お問い合わせ・参考文献

佐藤 純也(株式会社らしく 代表取締役)

リクルートの情報システム部門でシステム企画・開発に携わったのち、J.D.エドワーズ(現・オラクル)、ダブルクリック(現・グーグル)、セールスフォース(中堅・中小企業向けコンサルティング部門の責任者)などで、顧客企業のIT導入と組織変革を手がける。2011年に株式会社らしくを設立し、Google Cloud基盤のシステム開発と、企業のDX推進・生成AI活用の定着支援を行う。大手企業を中心に研修登壇は30回以上。2025年夏にClaude Codeを、事業開発やマーケティングなど、ソフトウェア開発以外の仕事に転用し、その実践からAI協働フレームワークCycleGenを開発。判断の記録を3つの属性(巻末の注)で整理して管理する仕組みは特許第7873811号として、指示が短くなるのに品質が上がる転換点を検知して、その後の動きを変える仕組みは特許第7929338号として、それぞれ登録されている。AIとの協働において、人が判断する工程をどこに置くかを設計しており、本稿で扱った手順と記録の仕組みは社内で毎日使っている。

ご連絡は info@rashikucorp.com までお願いします。CycleGenの入れ方と、組織での活用のご案内は <https://cyclegen.ai> にあります。

注:3章の数字の数え方

私は作業記録の書き方について、あるルールをAIに毎回渡しています。2026年9月19日から26日までの8日間にAIと書いた作業記録66本を調べたところ、ルールどおりと確認できたのは1割弱でした。渡しているルールは「番号だけで前の項目を指さない。番号のうしろに、何の話かが分かる手がかりを添える」というものです。作業記録66本の中で、前に出た項目を番号で指している箇所は798件あり、そのうち手がかりが添えてあったのは68件(9%)でした。ただし、このルールには「すぐ上の1~2行で説明した番号には付けなくてよい」という例外があり、それは機械的には見分けられません。そのため、残りの730件がすべてルール違反とは言えません。

注:3章の補足(チェックの手間についての経済学の見方)

Demirerほか(2026)は、AIが仕事の手順をいくつも続けて処理する場合、出力をチェックする費用は、手順ごとにかかる変動費ではなく、一連の作業ごとにかかる固定費になる、という経済モデルを示しました(書誌は参考文献)。論文は、だからAIにまとめて任せるほうが得だ、と結論しています。同じモデルから、私はもう1つのことを読み取ります。チェックが一連の作業ごとの固定費として残る以上、どれだけ長く任せても、最後に人がチェックするところはない、ということです。

注:CycleGenを使う場合の保存先と選び方(5章)

5章の3つの問い(どこに保存するか、毎回全部を渡すか、AIに渡してよいか)に、CycleGenを使う場合の答えを示します。保存先は、初期設定では自分のパソコンの中のファイルです。記録は1件ずつ普通のテキストファイルなので、CycleGenを外しても、パソコンの中にそのまま残ります。渡す量は、全部ではありません。いまの仕事に関係するものと、これまでよく使われてきたものだけが、AIへ渡されます。渡してよいかについては、次のとおりです。

初期設定のままなら、書いたものが自分のパソコンの外に出ることはありません。組織で記録を共有する設定にしたときだけ、共有すると承認したものと、検索キーワードが、組織用のサーバーへ送られます。

ただし、いまの仕事のためにAIへ渡した記録は、ほかの入力と同じように、使っているAIのサービスへ送られます。そこには、会社がそのAIについて定めている利用ガイドラインが当てはまります。

組織で記録を共有する設定にするかどうかは、AIの利用ガイドラインとは別に、社内の情報システム部門と相談して決めてください。

記録には、次の3つの属性を付けて管理します。

属性	中身
抽象度	その案件だけの話か、ほかの仕事にも使える考え方か
重要性	これまでどれだけ役に立ってきたか
文脈	どの種類の作業で生まれたか

このうち重要性は、時間が経っても自動では上がりません。実際に使われると上がり、渡された記録を見た人が的外れだと判断すると下がります。記録を選ぶときに効くのは、いまの仕事との近さと重要性の2つで、この2つで、いま渡すべき記録が選ばれます。

時間で下がらないようにしているのには、理由があります。答えが一つに決まらない仕事では、半年前に下した判断が、似た仕事 came きたときにそのまま効くことがあるからです。古いから下げるのではなく、使われたものが上がり、的外れだと判断されたものが下がるようにしておく、使う側から見れば、いまの仕事に要る分だけが戻ってくることになります。裏返せば、古くなった判断も、的外れだと判断されるまでは残り続けます。いつ外すかは、組織で使うときにまだ決まっていないことの1つです(9章で触れます)。

判断の記録と仕事の進め方を溜めておき、3つの属性で整理して、次の仕事に要る分だけをAIに渡す。この保存先を、私たちは「スキル・記憶ストア」と呼んでいます。スキルは仕事の進め方、記憶は判断の記録のことです。AIのメモリ機能とは別のもので、理由まで書いた記録を自分の手元に持つための仕組みです。仕事の経緯は、サイクルの終わりごとに書く短い作業記録に残し、判断の記録と一緒に次の仕事へ渡します。

注:「人が、育つ」で、学びの様子が見える範囲(6章)

仕事のすぐそばで学べるというと、上司や人事に、学びの様子まで見られるのではないかと気になる方もいると思います。仕事の中身は、初期設定では手元から出ません。学びの様子が人から見えるのは、本人が出したものと、組織で取り決めた範囲だけです。

参考文献

Demirer, M., Horton, J. J., Immorlica, N., Lucier, B., & Shahidi, P. (2026). "Chaining Tasks, Redefining Work: A Theory of AI Automation." NBER Working Paper No. 34859; [arXiv:2606.15960](https://arxiv.org/abs/2606.15960). 27th ACM Conference on Economics and Computation (EC '26) 採択論文。本文ではSection 1を参照した。

初版 / 2026年10月 / 株式会社らしく