

推しチーム設計シート

2人目のAI部下を増やす前に、役割・記憶・チェック体制を整理する実践ワークブック
桐崎和也 note特典 第2弾 / 2026-06-26 draft v2



AI部下スターターキットと推しチーム設計シート

このシートで作るもの

第1弾の「AI部下スターターキット」は、1人目のAI部下を作るための資料でした。

この第2弾は、その次です。

1人目が動き始めたあと、2人目、3人目をどう増やすか。

どのAIに何を任せるか。

どの記憶を共有して、どの記憶を分けるか。

誰が作り、誰が確認し、誰が最後に決めるか。

ここを決めるためのワークブックです。

AI部下は、増やせば増やすほど強くなります。

でも、いきなり増やすと崩れます。

役割がかぶる。

記憶が混ざる。

同じ作業を別々に始める。

誰が確認するのか分からなくなる。

最後は、人間が全部見比べることになる。

それでは、AI部下ではありません。

ただチャット欄が増えただけです。

このシートの目的は、AIを増やすことではありません。

自分の仕事を前に進める「推しチーム」を作ることです。

1. なぜ、いきなり10人に増やしてはいけないのか

悪い1人目 / 良い1人目

チャット欄の群れから、仕事を進める相棒へ

The image compares two AI assistant scenarios. On the left, 'Bad AI Assistant' shows a user frustrated with a chaotic chat interface. The chat history is a jumble of repetitive and unhelpful prompts like 'AI (GPT) 要件をまとめてください' and 'AI (GPT) もう少し詳しく教えてください'. The user's notes are messy and incomplete. On the right, 'Good AI Assistant' shows a user working smoothly with a structured chat interface. The chat history is organized with clear roles and standards. The user's notes are clear and actionable. A large arrow points from the 'Bad' side to the 'Good' side.

悪い1人目
たくさんいるのに、誰も覚えてくれない

良い1人目
役割・基準・記憶を持ち、一緒に前へ進める相棒に

AIパートナー (1人目)
あなたの仕事を前に進める、専属パートナー

役割 (ロール)
企業の壁打ち相手
情報整理と構造化
提案と意思決定の支援
レポート作成のサポート

判断基準 (基準)
目的達成に寄与するか
事実ベースであるか
シンプルで再現性があるか
時間対効果が高いか

記憶 (メモリ)
プロジェクトの目的
ターゲット・条件
これまでのやり取り
意思決定の経緯

レポートの型 (アウトプットスタイル)

要約サマリー	比較表	提案書	アクションプラン	次の手
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

相談・壁打ち ▶ 整理・分析・提案 ▶ 意思決定・実行へ

文脈を理解し、伴走してくれるから、仕事が前に進む!

悪い1人目と良い1人目

AIを毎日使っていると、こう考えたくくなります。

文章担当AI

リサーチ担当AI

投稿担当AI

画像担当AI

開発担当AI

品質チェックAI

営業AI

秘書AI

記憶管理AI

分析AI

並べるだけなら、強そうに見えます。

でも、最初からこれをやると、たいてい管理が破綻します。

理由は単純です。

チームに見えて、実際には役割の境界線がないからです。

たとえば、文章担当AIと投稿担当AIの違いが曖昧だと、両方が同じ投稿文を作ります。

リサーチ担当AIと分析AIの違いが曖昧だと、どちらも同じ競合調査を始めます。

品質チェックAIに止める権限がないと、結局人間が全部止めます。

記憶管理AIが何を覚えるか決まっていないと、全員が違う前提で動きます。

結果、人間の仕事は減りません。

むしろ増えます。

AのAIに説明する

BのAIにも説明する

CのAIには別の前提を伝える

出てきた回答を人間が見比べる

矛盾したら人間が直す

最後に投稿先やルールも人間が確認する

これでは、AIを増やした意味がありません。

推しチームを作る順番は、こうです。

1. まず1人目の役割を固定する
2. 1人目の苦手な作業を見つける
3. その作業だけを2人目に切り出す
4. 1人目と2人目の境界線を決める
5. チェック役と最終判断者を決める
6. 失敗を記憶に戻して育てる

いきなり人数を増やすのではなく、足りない役割だけを増やします。

2. 桐崎和也の発信部で見る、推しチームの例

ここでは、自分たちの運用を例にします。

桐崎和也の個人発信では、AI部下をただ並べているわけではありません。それぞれの担当と止め線があります。

進行AI: 進行と導線の担当

役割:

桐崎和也の発信部全体の進行、特典導線、資料ライブラリ、公開前の整理を担当する。

得意なこと:

- note / X記事 / 特典 / ライブラリの導線整理
- タスクの分解
- 状態管理
- 運営者への報告
- 公開に向けた準備

やらないこと:

- QC担当の最終確認を飛ばす
- 未確認の投稿先に勝手に出す
- 特典と開発メモを混ぜる

今回の例で言えば、第一弾特典、第二弾特典、MCP/APIスターターパックを同じ場所に並べてしまうと、読者の理解が混ざります。

だから、通常特典と開発メモを別枠に分けました。

これは進行担当の仕事です。

QC AI: 品質と法務の担当

役割:

公開前に、品質、法務、表現リスク、軸ズレを確認する。

得意なこと:

- 過剰表現の検出

- 成果保証に見える文言の修正
- 読者に誤解される導線の指摘
- 前回より品質が落ちていないかの確認

止めるべきもの:

- 必ず稼げる
- 完全自動で成果が出る
- 公式連携のように見える表現
- 公開先アカウントの混同

AIチームにQC担当がいないと、作るAIが出したものをそのまま出してしまいます。

でも、発信では「作れる」と「出していい」は別です。

ここを分けるだけで、運用はかなり安定します。

実装AI: 実装と自動化の担当

役割:

サイト、スクリプト、自動化、検証コマンド、デプロイ周りを担当する。

得意なこと:

- ページ修正
- PDF配置
- 自動化スクリプト
- dry-run
- エラー調査

止めるべきもの:

- secretを含む出力
- 本番アカウント混同
- 検証なしのデプロイ
- 仕様未確定の自動実行

実装担当がいると、アイデアが仕組みに変わります。

ただし、実装担当に公開判断まで任せると危険です。

作る人と出す人は分けた方がいい。

リサーチAI: リサーチと候補出しの担当

役割:

伸びている投稿、反応先候補、競合の動き、読者の空気を拾う。

得意なこと:

- Xの引用候補
- 反応が取れているテーマ
- 競合調査
- 投稿ネタの発見

やらないこと:

- 勝手にリプする
- いいねを自動実行する
- 投稿文の最終決定

リサーチ担当は、候補を出すところまでが強いです。

でも、候補をそのまま実行すると事故になります。

候補出しと実行は、別の役割に分けるべきです。

3. 推しチーム設計の基本形



推しチームの原型

最初の推しチームは、3つの役割で十分です。

作るAI:

下書き、案、候補、実装、素材を作る。

確認するAI:

ズレ、リスク、抜け漏れ、表現の強さを確認する。

最終判断する人間:

公開、実行、投稿、支払い、方針変更を決める。

大事なのは、最初から豪華な組織図を作らないことです。

まずは、作るAIと確認するAIを分ける。

そして、最後は人間が決める。

この3点だけでも、AI部下は「チャット」から「チーム」に近づきます。

最小構成の例

作るAI:

note本文、X投稿案、特典PDFの本文を作る

確認するAI:

読者に伝わるか、表現が強すぎないか、前回と重複していないかを見る

人間:

公開するか、どこに出すか、どの表現を採用するか決める

発信チームの例

作るAI:

編集AI。note構成、X投稿、CTA案を作る。

確認するAI:

QC AI。誤解、過剰表現、導線の弱さ、重複を確認する。

人間:

桐崎和也本人。公開判断、思想の強さ、実体験として出す範囲を決める。

開発チームの例

作るAI:

実装AI。HTML、CSS、スクリプト、テストを作る。

確認するAI:

レビューAI。仕様違い、セキュリティ、投稿先混同、デプロイ漏れを確認する。

人間:

本番反映、費用発生、外部アカウント操作を判断する。

4. 2人目を増やしていい条件

2人目を増やす前に、次のチェックをします。

- 1人目に同じ依頼を3回以上している
- 1人目の役割が広がりすぎている
- 作る人と確認する人を分けたい
- リサーチと執筆を分けたい
- 実装と品質チェックを分けたい
- 投稿先やルール確認を別担当にしたい
- 自分が毎回やっている確認作業をAIに渡したい
- 1人目の出力をレビューして改善できている

3つ以上当てはまるなら、2人目を増やす意味があります。

逆に、次の状態ならまだ増やさない方がいいです。

- 1人目に何を任せるか決まっていない
- 出力レビューをしていない
- 失敗を記録していない
- 何を任せないか決まっていない
- ただAIを増やすこと自体が目的になっている

2人目は、便利そうだから増やすものではありません。

1人目から切り出した方が安定する仕事が見えた時に増やします。

5. 役割カードを書く

2人目以降は、必ず役割カードを書いてから動かします。



設定1 役割

【名前】

【役割】

【主な仕事】

-

-

-

【やらない仕事】

-

-

-

【判断基準】

-

-

-

【人間確認が必要な場面】

-

-

-

【共有してよい記憶】

-

-

-

【共有してはいけない記憶】

-

-

-

【相互チェック相手】

ポイントは「やること」よりも「やらないこと」を決めることです。

AIは真面目です。

頼めば広くやろうとします。

でも、チーム運用では広すぎる担当は事故の原因になります。

サンプル: 発信編集AI

【名前】

ミナ

【役割】

桐崎和也のnote / X記事 / 特典導線を整える編集AI。

【主な仕事】

- note構成の改善
- X投稿案の作成
- 特典導線の自然な接続
- 読者が保存したくなる具体例の追加

【やらない仕事】

- 最終投稿
- 本番アカウント操作

- 実績の断定
- 法務判断

【判断基準】

- 実体験が入っているか
- AI部下を育てる思想が残っているか
- 読者が次に何をすればいいかわかるか
- エージェントメモリーズへの導線が押し売りに見えないか

【人間確認が必要な場面】

- 公開判断
- 外部リンクを貼る判断
- 強い表現を使う判断
- 有料商品や登録導線を出す判断

【共有してよい記憶】

- 過去に反応が良かった投稿
- 桐崎和也の文体
- 特典の配布順
- Agent Memoriesの3つの軸

【共有してはいけない記憶】

- secret
- 未公開の内部情報
- 他アカウントの混同につながる情報

【相互チェック相手】

QC担当AI

2.

3.

サンプル: 桐崎和也の発信判断

【優先すること】

1. 一般論ではなく、実際にAI部下を運用している実例を入れる
2. AIを便利ツールではなく、役割と記憶を持つ部下として見せる
3. 読者が次に真似できるテンプレートを入れる

【止めること】

1. 完全自動で稼げるように見える表現
2. 公式連携や保証に見える表現
3. 特典、開発メモ、商品導線の混同

【迷った時の判断】

1. 読者の行動が明確になる方を選ぶ
2. ブランドを壊す強い煽りは避ける
3. 公開や実行は人間に戻す

ここまで渡すと、AI部下はただ案を出すだけではなく、止めるべきところで止められるようになります。

7. 記憶を分ける



設定4 記憶

AI部下を増やす時に、いちばん大事なものは記憶の分け方です。

全員に全部を覚えさせると、一見強そうに見えます。

でも、実際は混ざります。

桐崎和也の個人発信のルール。

Agent Memories公式のルール。

漫画アフィリのルール。

法人noteのルール。

別アカウントの投稿方針。

これらが混ざると、投稿先を間違えたり、読者に見せるべきではない情報を出したりします。

だから、記憶は分けます。

全員で共有する記憶:

- ブランド方針
- 禁止事項
- 重要な約束
- 公開済みの事実

担当AIだけが持つ記憶:

- 作業手順
- 担当領域の失敗
- よく使うテンプレ
- 担当アカウントの注意点

人間だけが持つ記憶:

- secret
- 未確定の重要方針
- 外部に出せない情報
- 感情的な判断が必要なこと

サンプル: 特典制作チームの記憶分担

全員共有:

- 第一弾はAI部下スターターキット
- 第二弾は推しチーム設計シート
- MCP/APIスターターパックは別枠の開発メモ
- 公開前に本番HTMLを確認する

編集AIだけ:

- note読者向けの言い回し
- CTAの流れ
- 読者が欲しくなる具体例

QC AIだけ:

- 過剰表現
- 成果保証リスク
- 投稿先混同リスク

実装AIだけ:

- PDF配置先
- ライブラリHTML
- デプロイ手順
- 検証コマンド

人間だけ:

- 公開GO
- アカウント操作
- secret
- 最終的な事業判断

記憶は、量よりも境界線です。

誰が何を覚えるかを定めるほど、AI部下は安心して任せられる相手になります。

8. 報告スタイルを決める

設定5 報告スタイル

読んで判断できる報告にする

AI AIが要点を整理しました

- ✓ 4つの観点で簡潔に報告
- ✓ 判断と理由を明確化
- ✓ 次にやることを具体化
- ✓ 確認事項を明確化

設定5 報告スタイル

状況

- プロジェクトは計画通り進行中
- 主要タスクの完了率：72%

判断

- 優先順位をB案に変更する
- 理由：効果と工数のバランスが良い

次アクション

- B案の実行に向けて要件を確認
- 関係者への共有とスケジュール調整

確認事項

- 追加予算の承認をお願いしたい
- リリース時期のご意見を確認したい

プロジェクト進捗レポート

報告日：2024/05/04

72%

状況

- プロジェクトは計画通り進行中
- 主要タスクの完了率：72%

判断

- 優先順位をB案に変更する
- 理由：効果と工数のバランスが良い

次アクション

- B案の実行に向けて要件を確認
- 関係者への共有とスケジュール調整

確認事項

- 追加予算の承認をお願いしたい
- リリース時期のご意見を確認したい

承認済

リリース時期

設定5 報告スタイル

AI部下が増えると、報告が長くなりがちです。

長い報告は、一見がんばっているように見えます。

でも、人間が判断できない報告は役に立ちません。

推しチームでは、報告スタイルも決めます。

【結論】

【やったこと】

【確認したこと】

【問題・リスク】

【人間に判断してほしいこと】

【次にやること】

サンプル: 特典PDFを作った時の報告

【結論】

第二弾特典PDFの確認版を作成しました。まだ本番リンクには反映していません。

【やったこと】

- 推しチーム設計シートの本文を作成
- note第2弾の画像を再利用
- A4 PDFとして出力
- 先頭ページを目視確認

【確認したこと】

- ページ数
- PDFサイズ
- 主要章の本文抽出
- タイトル重複の有無

【問題・リスク】

- 初版は情報量が薄かったため、第一弾と同じ密度まで増やす必要あり

【人間に判断してほしいこと】

- このPDFを第二弾特典として公開してよいか

【次にやること】

OKならライブラリに配置し、近日追加をPDFリンクへ差し替える

報告は、読み物ではなく判断材料です。

人間が次の判断をできる形にします。

9. 相互チェックの流れ

チーム運用では、1人のAIに全部を任せない方が安定します。

おすすめはこの流れです。

1. 作るAIが下書きや候補を出す
2. 確認するAIがズレとリスクを見る
3. 人間が公開・実行・採用を決める
4. 結果を記録する
5. 次回の記憶に戻す

チェック項目:

- 目的に合っているか
- 読者に伝わるか
- 禁止事項に触れていないか
- 投稿先やアカウントを間違えていないか
- 強すぎる表現がないか
- 既存方針と矛盾していないか
- 次に改善する点が残っているか

ここで大事ななのは、確認役に「直して」と頼むだけで終わらせないことです。

どこがズレたか。

なぜズレたか。

次から何を記憶に戻すか。

ここまで残すと、AI部下は少しずつ育ちます。

レビュー依頼テンプレ

あなたは、この成果物のQC担当です。

以下の観点で確認してください。

1. 目的に合っているか

2. 読者に伝わるか
3. 事実と推測が混ざっていないか
4. 過剰表現がないか
5. 投稿先・配布先・アカウントが混ざっていないか
6. 次に改善すべき点は何か

出力形式:

【判定】

PASS / HOLD / MUST FIX

【良い点】

【修正必須】

【できれば直す】

【人間確認】

10. 7日間のチーム化メニュー

Day 1: 1人目の役割を言語化する

このAI部下は、_____を担当する。

例:

このAI部下は、桐崎和也のnoteとX記事を、読者が保存したくなる形に整える編集担当。

Day 2: 2人目に切り出す作業を1つ選ぶ

2人目に渡したい仕事:

理由:

1人目に残す仕事:

人間が決めること:

Day 3: 2人目の役割カードを書く

名前、役割、やること、やらないこと、判断基準を書きます。

Day 4: 境界線を書く

1人目がやること:

2人目がやること:

どちらもやらないこと:

人間が決めること:

Day 5: 小さな仕事で相互チェックする

例:

編集AI:

X投稿案を5本作る

QC AI:

重複、過剰表現、CTAの弱さを確認する

人間:

1本だけ採用する

Day 6: ズレを記憶に戻す

今回ズレたこと:

原因:

次回から覚えておくこと:

どのAIに覚えさせるか:

Day 7: 3人目を増やすか判断する

- 2人目の役割が安定した
- 1人目との境界線が説明できる
- チェック役が機能している
- 人間が最終判断する場面が決まっている
- 増やしたい理由が明確

全部にチェックが入らないなら、まだ増やさなくて大丈夫です。

11. そのまま使えるチーム設計テンプレ

【チーム名】

【このチームの目的】

【人間が最終判断すること】

-
-
-

【AI 1】

名前:

役割:

主な仕事:

やらない仕事:

判断基準:

記憶すること:

確認相手:

【AI 2】

名前:

役割:

主な仕事:

やらない仕事:

判断基準:

記憶すること:

確認相手:

【AI 3】

名前:

役割:

主な仕事:

やらない仕事:

判断基準:

記憶すること:

確認相手:

【共有記憶】

-

-

-

【分ける記憶】

-

-

-

【チームの禁止事項】

-

-

-

【失敗した時に残す記録】

何が起きたか:

なぜ起きたか:

次から誰が確認するか:

どの記憶に戻すか:

12. サンプル: 桐崎和也 発信部

【チーム名】

桐崎和也 発信部

【このチームの目的】

AI部下、AIパートナー、Agent Memoriesの思想を、note / X / 特典 / ライブラリで読者に伝える。

【人間が最終判断すること】

- note公開
- X投稿
- 特典PDFの公開
- 外部アカウント操作
- 事業方針や価格に関わる表現

【AI 1】

名前: 進行AI

役割: 進行・導線整理

主な仕事: 特典構成、ライブラリ表示、タスク整理、報告

やらない仕事: QC飛ばし、本番アカウント混同、secret操作

判断基準: 読者導線が自然か、約束した特典と別枠が混ざっていないか

記憶すること: 特典順、公開状態、未完了タスク

確認相手: QC AI

【AI 2】

名前: QC AI

役割: QC・法務・品質

主な仕事: 過剰表現チェック、品質チェック、公開前HOLD判断

やらない仕事: 勝手な公開、担当外作業の横取り

判断基準: 誤解、保証表現、導線混同、品質劣化がないか

記憶すること: 過去の事故、禁止表現、QC観点

確認相手: 運営者

【AI 3】

名前: 実装AI

役割: 実装・自動化

主な仕事: HTML修正、PDF配置、スクリプト、検証

やらない仕事: 投稿文の最終判断、アカウント混同、検証なしデプロイ

判断基準: 動くか、壊さないか、検証できるか

記憶すること: 配置先、検証コマンド、デプロイ手順

確認相手: 進行AI / QC AI

【共有記憶】

- 桐崎和也の個人発信とAgent Memories公式は分ける
- 第一弾はAI部下スターターキット
- 第二弾は推しチーム設計シート
- MCP/APIスターターパックは別枠の開発メモ

【チームの禁止事項】

- 投稿先アカウントを混ぜない
- 特典と開発メモを混ぜない
- 完全自動で稼げるような表現をしない
- 公開前確認なしで本番に出さない

このように書くと、AI部下が「何となく手伝う存在」から「担当を持つ存在」に変わります。

13. Agent Memoriesにつなげる理由



次は記憶を持ち出す話へ

ここまで読むと、分かるはずですが。

AI部下をチームにする時、いちばん面倒なのは「毎回説明すること」です。

役割を説明する。

判断基準を説明する。

前回の失敗を説明する。

どの特典が第一弾で、どれが別枠か説明する。

どのアカウントで発信するか説明する。

何を公開してよくて、何を止めるべきか説明する。

この説明を毎回やるなら、AI部下はまだチームではありません。

毎回新人に戻っているだけです。

だから、Agent Memoriesでは「記憶が積み上がるAIパートナー」を作っています。

単にチャット履歴を保存する話ではありません。

記憶を持ち出せる

スキルや外部ツールとつながる

自分専用のAIパートナーとして育てられる

この3つが揃うと、AI部下はただの便利ツールではなくなります。

昨日の失敗を覚えている。
前回の判断基準を持っている。
担当ごとに記憶を分けられる。
必要な時にツールを使える。
そして、人間は最後の判断に集中できる。

この「推しチーム設計シート」は、その入口です。

14. 次にやること

まず、次の5つだけやってください。

1. 1人目の役割を1文で書く
2. 2人目に渡したい仕事を1つだけ選ぶ
3. 2人目の役割カードを書く
4. 作るAIと確認するAIを分けて1回だけ試す
5. ズレたことを記憶に戻す

完璧なAI組織を作る必要はありません。

まず1人。

次に2人目。

それから、必要な役割だけを増やす。

推しチームは、数ではなく設計で強くなります。



Agent MemoriesへのCTA

Agent Memoriesでは、こうしたAI部下やAIパートナーを、記憶ごと育てられる仕組みを準備しています。

開発状況や追加特典は、桐崎和也の資料ライブラリと事前登録ページで順番に案内します。

資料ライブラリ:

<https://agentmemories.jp/kirisaki/library/>

Agent Memories 事前登録:

<https://agentmemories.jp/#join>

第1弾は「AI部下スターキット」。

第2弾は、この「押しチーム設計シート」。

次は、AI部下を毎回新人に戻さないための「記憶ルール」へ進みます。