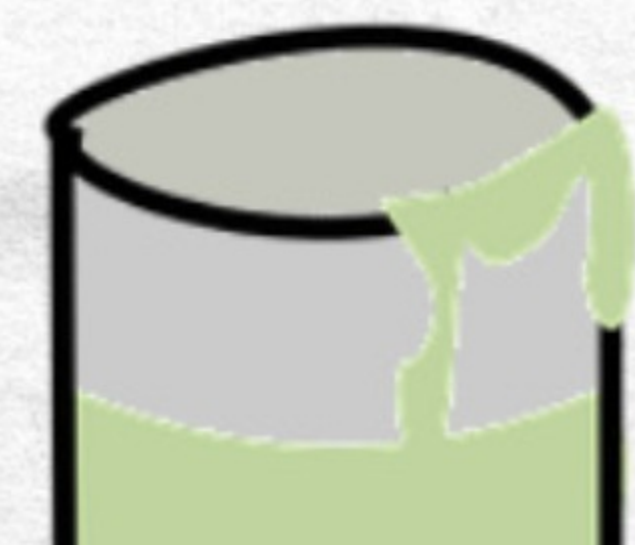


テック記事ライター 作った件について



2025/11/13



猫とゲームを愛するエンジニア

IT関連の情報発信（とゲームや日常をポスト）
よかったらフォローしてください!!

X : @ochtum18

↳ Github : ochtum

Youtubeは忙しいのであきらめた＼(^o^)/



ターゲット層：

- ・テックブログを書きたいけど余裕がない人
- ・テックブログを時短したい人

Qiita記事を「自分の言葉、文章」で
書いてくれるAIはないものだろうか？



とにかく手抜きしてすごいものが
作りたいんだ！！

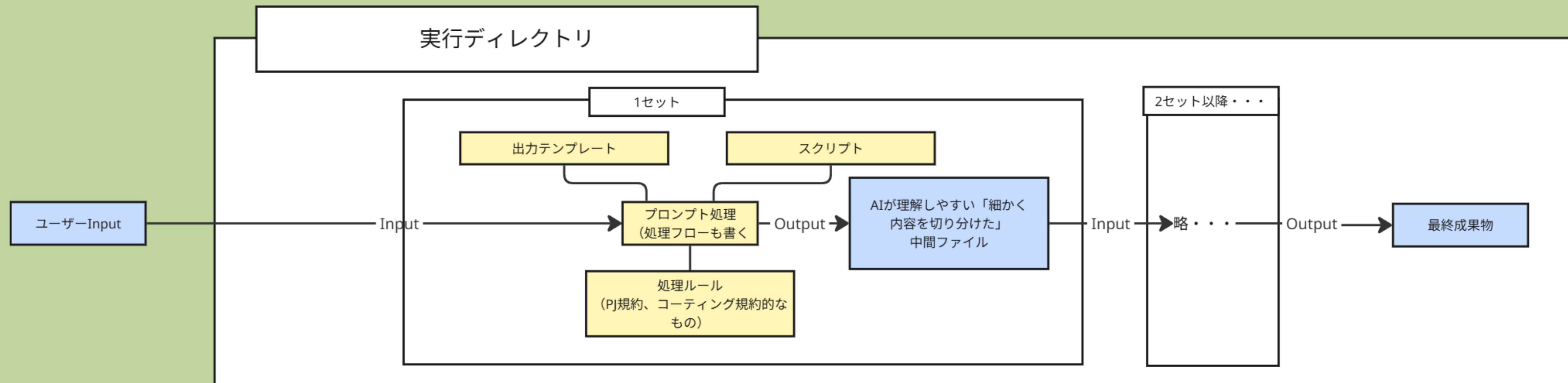
Spec Kitとは？

AIを活用した仕様駆動開発(SDD)を実現するための、OSSツールキット。

GitHubから2025年9月2日に
発表された。

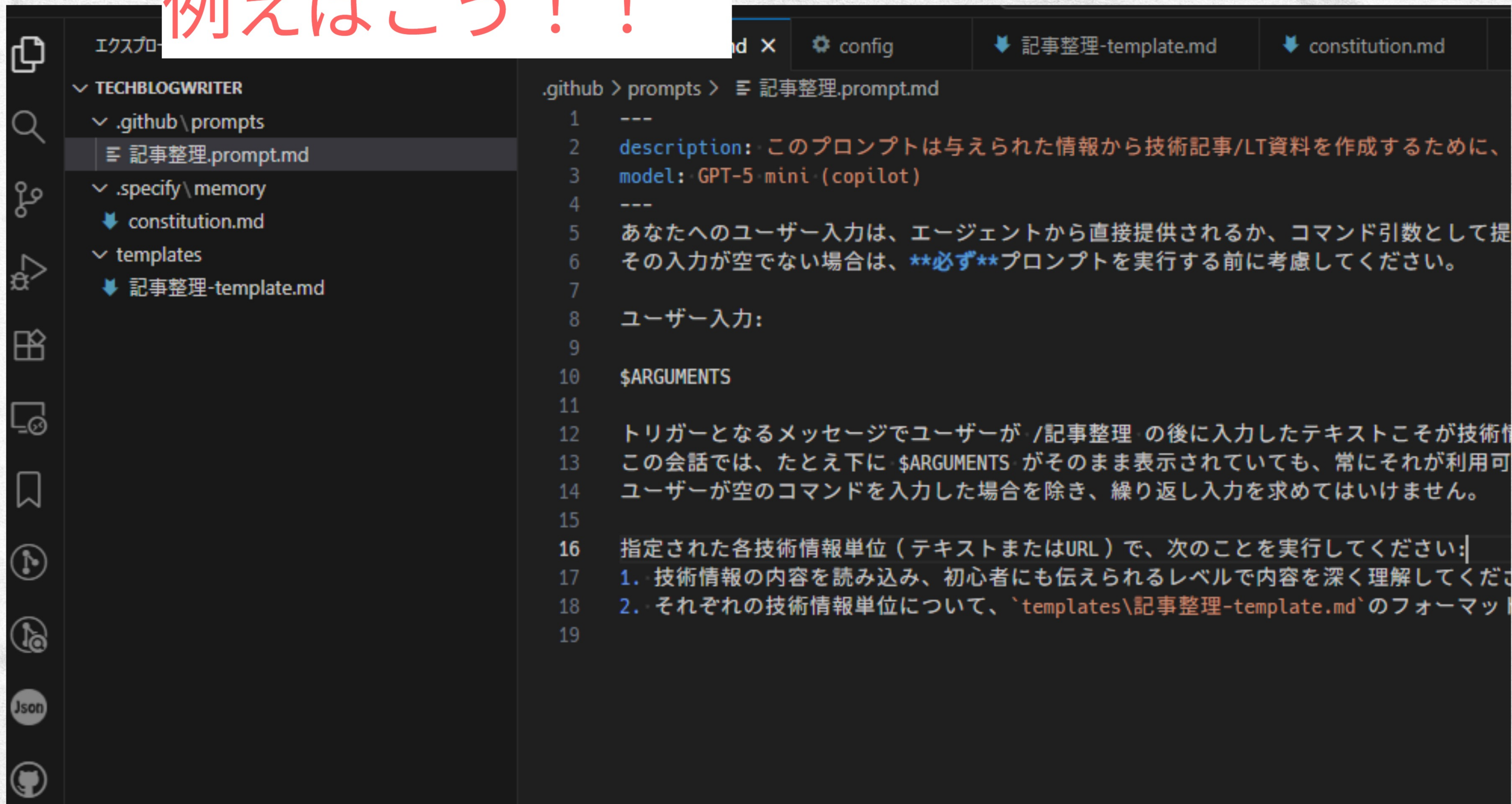
これを試してみた

別件でSpecKitを調査していて、
紐解くとSpecKitはこういう
構成であることに気づいた！！



この考え方を流用すれば
自作SpecKitが作れるのでは？

例えばこう！！



```
.github > prompts > 記事整理.prompt.md
1  ---
2  description: このプロンプトは与えられた情報から技術記事/LT資料を作成するために、
3  model: GPT-5 mini (copilot)
4  ---
5  あなたへのユーザー入力、エージェントから直接提供されるか、コマンド引数として提供されるか、
6  その入力が空でない場合は、**必ず**プロンプトを実行する前に考慮してください。
7
8  ユーザー入力:
9
10 $ARGUMENTS
11
12 トリガーとなるメッセージでユーザーが /記事整理 の後に入力したテキストこそが技術情報です。
13 この会話では、たとえ下に $ARGUMENTS がそのまま表示されていても、常にそれが利用可能な情報です。
14 ユーザーが空のコマンドを入力した場合を除き、繰り返し入力を求めてはいけません。
15
16 指定された各技術情報単位（テキストまたはURL）で、次のことを実行してください:
17 1. 技術情報の内容を読み込み、初心者にも伝えられるレベルで内容を深く理解してください。
18 2. それぞれの技術情報単位について、`templates\記事整理-template.md`のフォーマットに従って記事を作成してください。
19
```

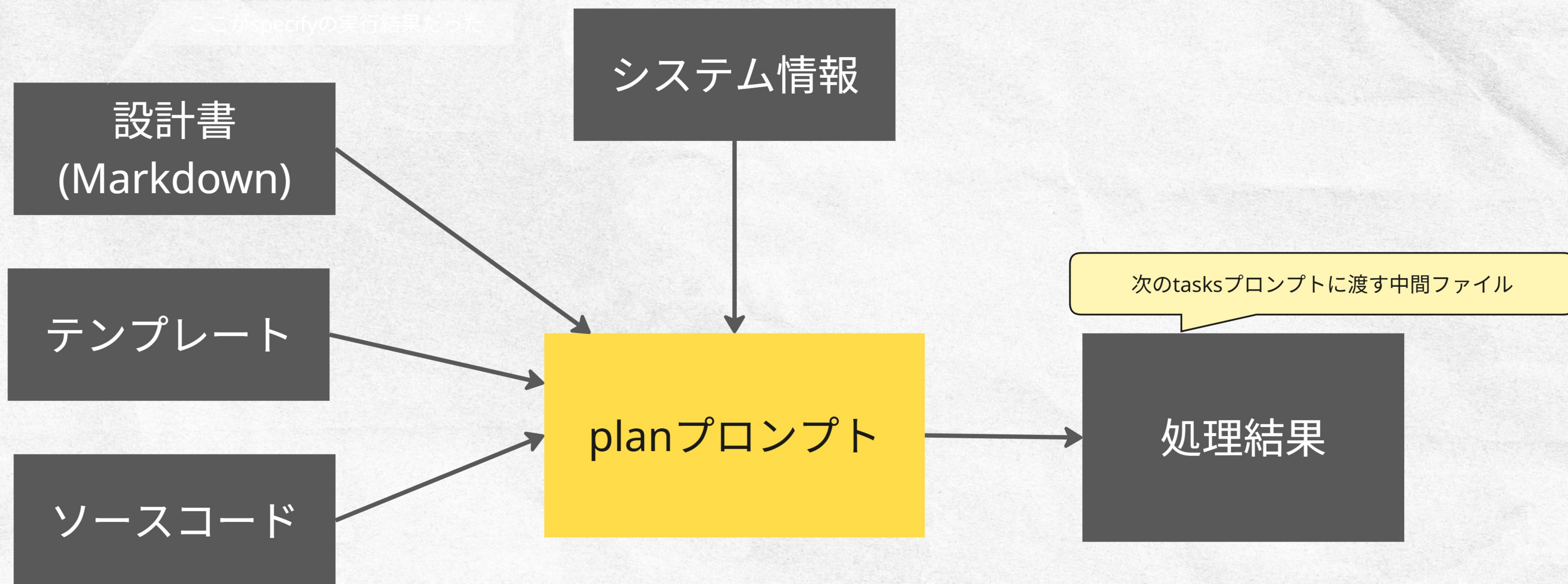

この考え方を流用して
テック記事ライターを作ってみた

興味がある方はこちら→

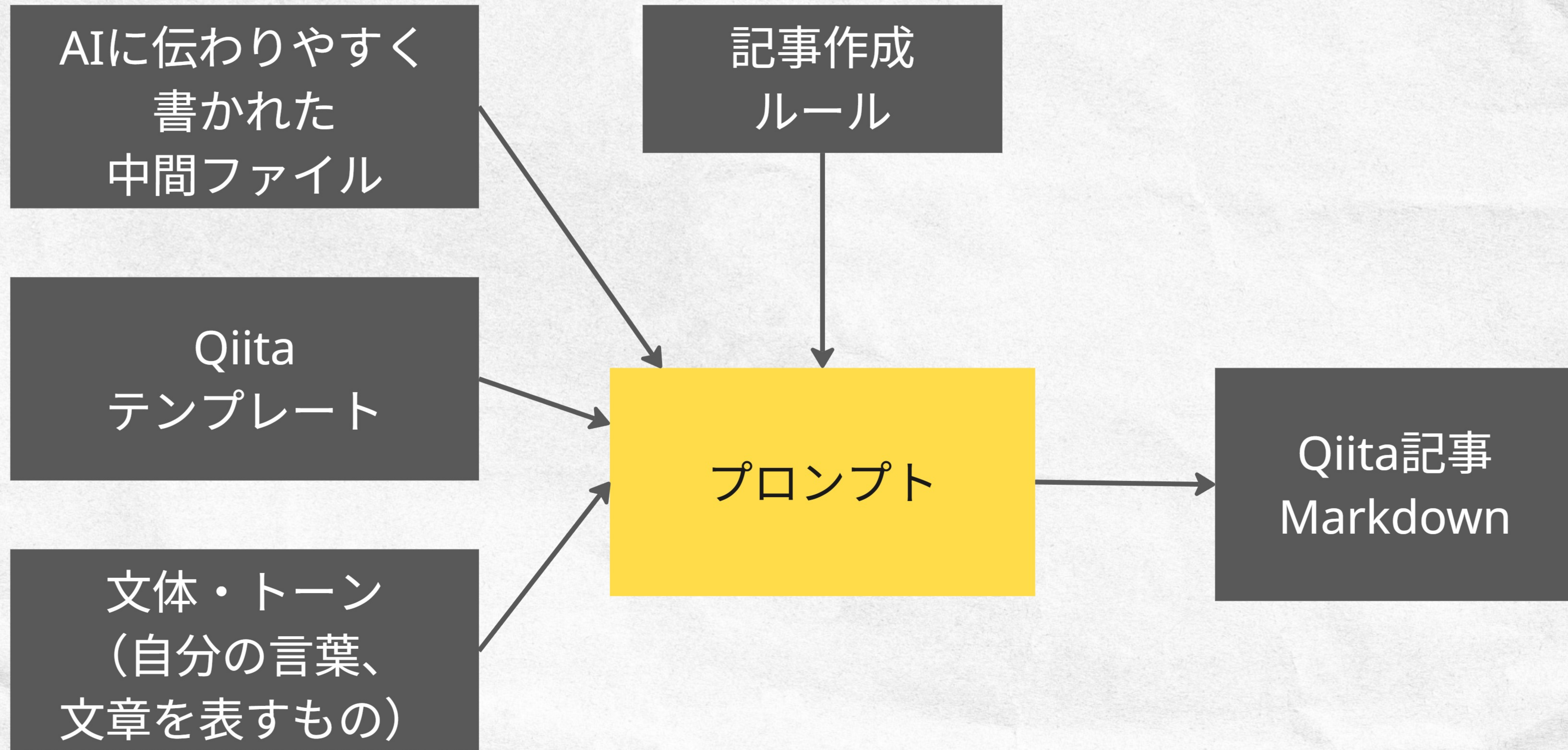


改めてテック記事ライター用に
SpecKitに必要な内容を整理すると
どうなるか？

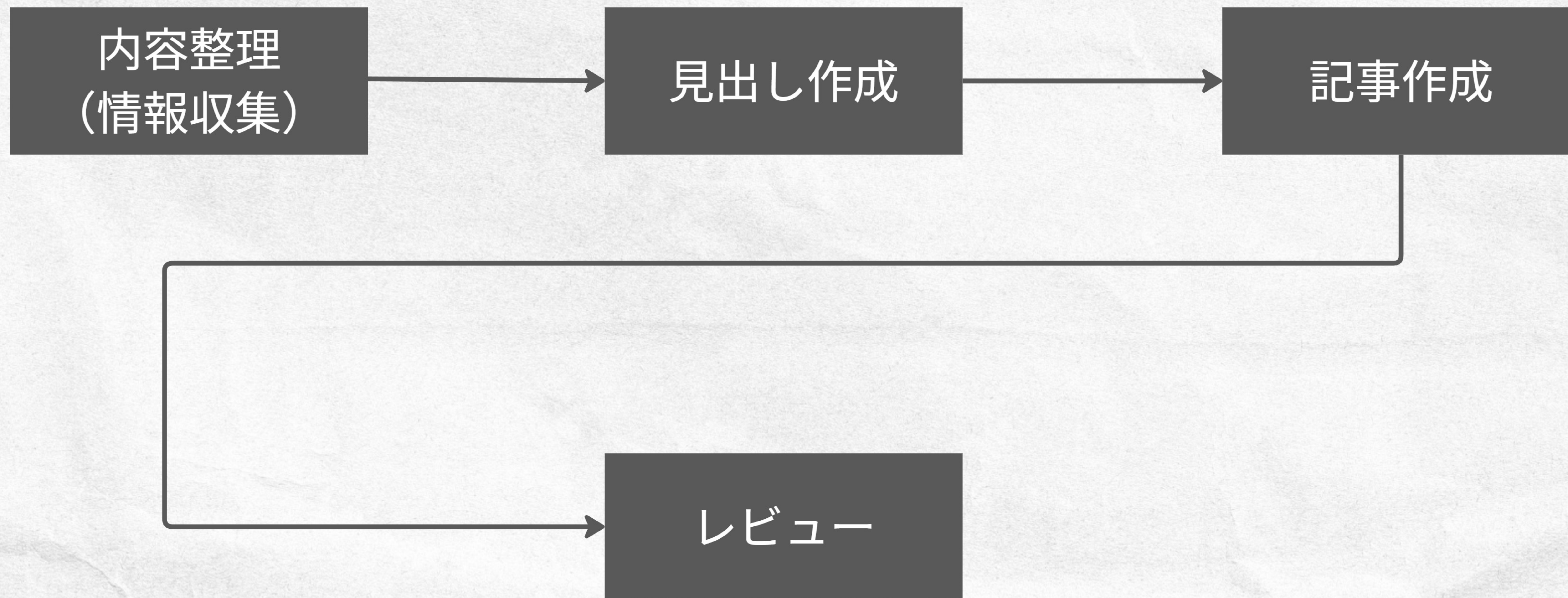
プロンプトに渡すもの (SpecKitでシステム改修)



プロンプトに渡すもの（テック記事ライター）



プロンプトの全体の流れ (イメージ)



これを試した結果・・・

AI Shell 入門 — PowerShell とスタンドアローンで始める AI シェル運用ガイド

Azure PowerShell AI OpenAI AISHell

最終更新日 2025年10月29日 投稿日 2025年10月29日 569 views

✔ ※お役に立てたらストック、いいねをよろしくお願いします！！

< 📄 本記事のターゲット層 >

- Windows ユーザー
- PowerShell ユーザー
- 開発者 / DevOps エンジニア
- システム管理者
- コマンドラインで AI を活用したいエンジニア

◆ イン트로: AI Shell とは何か

AI Shell (aish) は、コマンドライン上で AI エージェントと対話できるインタラクティブなシェルです。従来の GUI ベースの AI クライアントと異なり、シェル上で直接プロンプトを投げ、生成されたコマンドやコードをそのまま作業シェルに貼り付けたり、サイドカーとして PowerShell と並列に動かして効率的に作業できます。本記事では、Windows 環境を中心に PowerShell と AI Shell の違い、導入手順、基本操作、トラブルシューティング

いい感じに仕上がった
＼(^o^)／

◆ インストールと初期設定 (Windows 向けハンズオン)

ここでは Windows + PowerShell を前提に、最短で動かす手順を示します。インストール方法は主に 2 通りあります。手動でリリース実行ファイルを手入手して PATH に置く方法と、PowerShell の PSGallery からインストールする方法です。まず PSGallery を使う手順を示します。

- 1.) GitHubの[リリースページ](#)からインストール
- 2.) 解凍後、任意のパスに配置して、環境パスを設定
- 3.) PSGallery を信頼済に設定

```
Set-PSResourceRepository -Name PSGallery -Trusted
```

- 4.) PSGallery から AIShell をインストール

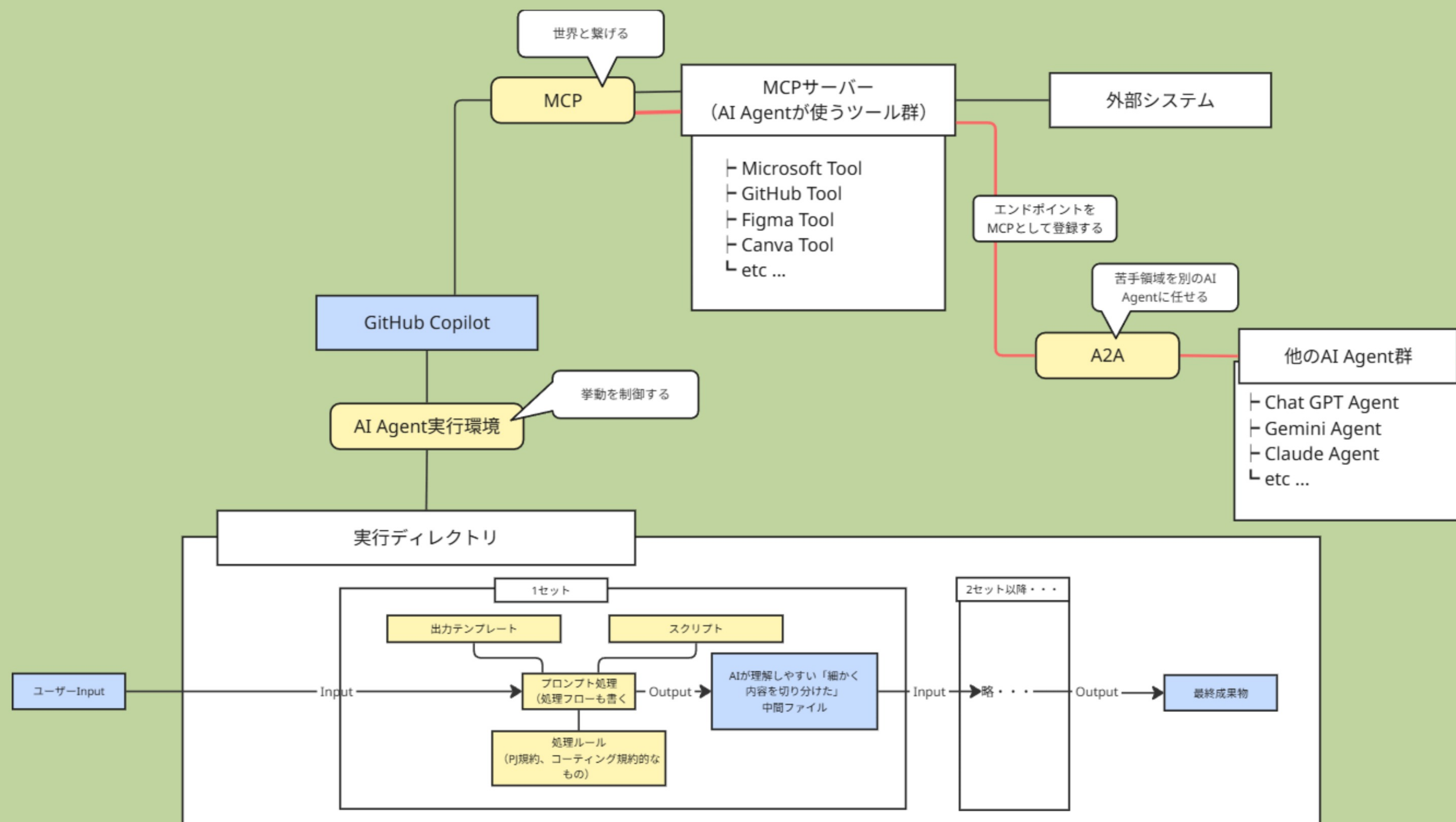
※本記事公開の時点ではプレリリースのため、コマンドに `-Prerelease` をつけています。

```
Install-PSResource AIShell -Prerelease -Repository PSGallery -TrustRepository -AcceptLicense
```

(必要なら) PSReadLine を更新してからモジュールをインポートし、`Start-AIShell` で起動します。

え！ハンズオンもある！！

いまはGitHub Copilotの環境をこう捉えている



Thank you



Qiita記事はこちら→

