

生成 AI 事例共有

| | |
|-------------------------------|-----------|
| 部門 | 東京本部 |
| 報告者 | 片山 直樹 |
| 工程(要件定義/基本設計/詳細設計/開発/テスト/その他) | その他 (PoC) |

1. 事例

COBOL から Java へのリプレース提案に向けた PoC（概念実証）を実施した。

大量の COBOL プログラム群の中から、指定した処理方式に該当するプログラムを抽出し、それらが Java へ言語変換可能かどうかを検証した。

- ・使用ツール：Cursor Pro 版（バージョン 1.0）
- ・実施時期：2025 年 6 月

2. うまくいった点

PoC の実施にあたり、いきなり COBOL から Java への言語変換に着手せず、まず AI に PoC 実施計画書の作成を指示した。

この際、Agent モードではなく、Plan モード（作業手順とチェックリストのみを作成させ、コード生成は行わせないモード。要件を深掘りしながら構成する）を使用した。

Plan モードにより、各作業タスクを丁寧に洗い出すことができ、洗い出したタスクを一つずつ順に実行させることで、意図と異なる作業を回避し、無駄のない PoC 実施が可能となった。

3. 反省点・改善点

言語変換後の機能実装の網羅性を検証するため、まず Claude 3.7 Sonnet（thinking）モデルを用いてユニットテストを作成・実行した。

しかし、Gemini 2.5 Flash Preview モデルに切り替えて再確認したところ、前回見逃されていたケース漏れや機能未実装が摘出された。

また、同一モデルを用いた場合でも、テスト実施の翌日に再チェックしたところ、異なる観点からの不備が発見されるケースもあった。

この経験から、1つのモデルだけで検証を完了させるのではなく、複数のモデルを用いてクロスチェックすることの重要性を認識した。

4. 共有可能な事例（プロンプト・設定等）

以下は、実際に使用した Plan モードプロンプトの例である。

目的

作業手順とチェックリストを作るだけ。コードは書かない。

ルール

- コード・テスト・ファイル編集は禁止

新しいアイデアや技術が実現可能か？

口でしかお客さん？

「Yes is important」
Input と Output も

自作？

図の絵で見た。

★ 同い経験

copilot 1つと GPT-4o

1つは Claude Sonnet
に代えて

Python FW の Django の件

yam さんとは？
組み立てたら

あちこちのリスト作ってくたして

アホバカの
アホバカは

GPT-4o に代えて

全然違う内容に代えて
した

- 要件について**質問を1つずつ**して、明確な回答を得てから次の質問をする
- わからなければ質問を深掘りして明らかにする
- 最後に「計画書を作成していいか」確認する

OKが出たら

- Markdownで`- []` チェックリストを書く
- 各項目は1つの具体的作業にする

進行の流れ

質問 → 回答 → 深掘り → 確認 → 仕様書作成

□ JLSA 勉強会
(リモ) + オンライン

~~Google~~ Google workspaceへ移行
X-ut Google. にするMT="o

• yamlの作成 ~~から~~ ~~テスト~~
T="T"と リスコードを書き出す

↓
リスコを書きだす ~~じゃ~~ コメントも書くようにして

• サンプル作成 ~~から~~

↓
読み込み

↓
コード生成