

A²P² AI Assisted Programming Process

酒匂寛
(有)デザイナーズデン
sako.hiroshi@gmail.com

要旨

本事例報告では、生成 AI を活用したプログラム開発の実践例をご紹介します。

1. はじめに

近年、チャット型生成 AI をプログラミングのアシスタントとして活用する事例が着実に増加しています。部分的なコード生成は確かに便利になりましたが、これらをプログラム開発プロセス全体に効果的に組み込む点については、まだ多くの課題が残されています。

たとえば、チャットとエディタの間で頻繁にコピー＆ペーストを繰り返す作業だけでも煩雑です。また、開発ワークフロー全体を見渡すと、コードを書く行為そのものは（確かに重要ではありますが）全体の中では限られた割合に過ぎません。

2. 現在の状況

開発アシスタントツールの分野では、たとえば Bolt.new, Replit Agent などのエージェント型 AI 開発ツールが注目を集めています。これらは、自然言語による指示だけで複数ファイルから成るプログラムを生成し、特定環境へのデプロイまで実行してくれる強力なツールです。いわば「AI によるノーコードツール」として、理想的な条件下では非常に効率的なソリューションとなります。

しかし、仕様が複雑化したり、途中で仕様変更が必要になったりした場合、大きな課題に直面することがあります。これらのツールは人間がコードを直接編集することをあまり想定していないため、問題が発生した際に外部からの指示だけで解決することが極めて困難になりがちなのです。

また、既存システムのリファクタリングや保守を行うためには、コードベースを綿密に分析する必要がありますが、コード生成に重点を置いたツールではそのようなきめ細

かな対応が難しくなります。

本報告では、生成 AI を効果的に開発環境に統合することで、単なる「丸投げ」ではなく、人間と AI が協調する効果的な開発アプローチを提案します。

3. アプローチ

開発プロセスに登場する中間的な成果物のうち、エッセンシャルなものにスポットライトを当てて、徐々に抽象度を下げていくアプローチをとります。

といっても難しい話ではなく、一見ウォーターフォールにもみえる「要求定義」「システム要件(仕様)」「実装」の抽象レベルです。これらの3つの抽象レベルの関係はたとえマネジメントスタイルが変わっても変化しません。

最後には様々なドキュメント(利用者マニュアル、ライブラリ初心者のための解説書)などの生成にも触れています。

4. 終わりに

本事例報告が、生成 AI を身近なパートナーとして使いこなすためのヒントとなれば幸いです。