2月の新着記事

1記事が投稿されました。AIを活用したUIコンポーネント開発の効率化に関心がある方に、ぜひ読んで頂きたい記事です。

生成AI時代のフロントエンドテスト戦略：なぜStorybook 10 + Vitest 4を選ぶのか

記事の紹介本文に入る前に、記事タイトルに含まれる「Storybook 10」「Vitest 4」について簡単に説明させてください。どちらもフロントエンド開発のツールです。

- Storybook 10 …UIコンポーネントをアプリ本体から切り離して確認・共有・開発するツール
- Vitest 4 …JavaScript系言語のテスト基盤。ブラウザを使ったコンポーネント検証にも対応

生成AIの速度を生かすために、新しいテスト戦略を

「人間がコードを書き、人間がレビューし、人間がテストを書く」時代のテスト戦略では、AIの生産性を活かしきれません。
この課題に対応するために、新しい戦略と実現方法を提案したのが本記事です。

本記事の基本方針は「AIが自律的に修正まで行う」です。具体的には次のような流れです。

- 人間が書いた仕様書を元に、生成AIがコンポーネントを**生成**、仕様書通りか自動的に**検証**、検証結果を踏まえて自動的に**修正**

タイトルの「Storybook 10 + Vitest 4」を新しい戦略に採用する理由

生成AIの活用によって効率の良いテスト戦略を実現するためには、以下が必要になります。

- 人間とAIが同一ファイルでコンポーネントの仕様を理解可能な状態にしつつ、AIが生成・検証・修正のサイクルを自律的に回せる環境を用意する

上記の実現について、筆者は記事タイトルのツールの活用を提案しています。

- **Storybook 10**の標準フォーマット（以下CSF3）で、コンポーネントの振舞いの仕様を記述することで、同じファイルを見て人もAIも理解できるようにする。さらに**Vitest 4**のテスト仕様書としてCSF3を使用する。
  これらを、AIが自律的に動作する基盤として使用する。

興味深いのは、AIとの連携手段として自然言語だけでなく、仕様を表すフォーマット自体を活用している点です。言われてみれば、自然言語だけにこだわる必要は無いです。

今回は考え方の核となる部分だけを紹介しました。記事本体では、AIに自律的に動作してもらう方法がより具体的に解説されています。記事本体もぜひご覧ください。

特集： 開発プロジェクトでのGitHub Copilot活用事例

GitHub Copilotで、開発（主にレビュー／受け入れ）の負荷を軽減した事例を２つ紹介します。

GitHub Copilotを用いたAIによるコードレビューの活用

１記事目は、有識者にレビューを依頼する「前」の工程（セルフレビュー）に焦点を当てた事例です。GitHub Copilotによるコードレビューを取り入れ、以下の課題に取り組みました。

- 抽象度の高い規約、例えば「各モジュール・クラス・関数は単一責任の原則に従っているか」等のレビューは、人の目で行うしか無かった。
- 開発者によってセルフレビューの質にバラつきがあった。

GitHub Copilotによるコードレビューを導入した結果、開発者のセルフレビュー時間の短縮と、品質向上を実現出来ました。
これにより、有識者はレビュー時にコード品質の最終チェックに集中できるようになり、アーキテクチャ評価や戦略的判断により多くの時間を確保可能になりました。

記事本体では、AIコードレビューの運用上の注意点や、組織への展開の工夫も扱っています。

- AIコードレビューの注意点及び、AIのコード提案に対する人間の関与の重要性
- 「組織全体の効率化」に向けた施策
  - レビュー用プロンプトを簡単に呼び出すVSCode拡張「Promptis」の作成と外部公開
  - コードレビュー用プロンプトの整備

詳細はWebサイトで記事本体をご覧ください。

GitHub Copilotを活用した大規模開発 ~オフショア開発での実践と知見~

２記事目は大規模・多拠点（オフショアを含む）開発で、コーディング～受け入れテストに焦点を当てた記事です。最大100名規模で並行開発する前提では、ツールを配るだけでは活用方法が属人化し、コード品質もばらつきます。

GitHub Copilot導入に当たって以下を行い、使い方のばらつきを抑えました。

- GitHub Copilotを用いた開発フローの整備
- GitHub Copilot Chatで使用するプロンプトの標準化
- GitHub Copilot Chatで使用するプロンプトの最適化
- プロンプト呼び出しの簡単化

このような工夫を取り入れつつ行ったGitHub Copilot活用のコンセプト検証で、受け入れ担当の負荷軽減などの効果が確認できました。詳細はWebサイトで記事本体をご覧ください。

発行日：03月19日