7. Vibe codingによる医療AIソフトウエア開発の可能性

平原 大助 帝京平成大学健康メディカル学部医療科学科

生成AIの進歩は，医療AIの研究開発だけでなく，ソフトウエア開発そのものを変えつつある。従来，医療AIソフトウエアを作成するには，臨床課題の整理，データ形式の理解，プログラミング，モデル実装，評価，運用設計など，多くの工程を専門家が分担して進める必要があった。近年は，自然言語で目的や制約を伝え，人工知能（AI）と対話しながらコード，テスト，画面，ドキュメントを反復的に作成する開発スタイルが広がっている。このような開発様式は，一般にvibe codingと呼ばれる。

vibe codingは，単に「AIにコードを書かせる」技術ではない。人間が何を作るべきか，どのような制約を満たすべきか，どのように正しさを検証するかを明示し，AIと反復しながら実装へ落とし込む開発文化である。医療分野では，仕様があいまいなまま開発を進めれば，もっともらしく動作するが臨床的には誤ったソフトウエアが生まれる危険がある。特に，医用画像，診療記録，検査値のように，誤りが患者の不利益に直結する領域では，生成された結果が「動く」ことよりも，評価可能，追跡可能であることが重要である。本稿では，vibe codingの概念と代表的な開発環境を概説した上で，DICOMビューワと自動セグメンテーション，音声入力からSOAP（subjective，objective，assessment，plan）記録を作成するソフトウエアを例に，応用可能性と注意点を述べる。

Vibe codingと開発環境

現在のAIコーディング支援は，単なるコード補完から，リポジトリ全体を読み，複数ファイルを編集し，コマンドを実行し，テスト結果を見ながら再修正するエージェント型へ移行している。代表的な環境には，Visual Studio Code + GitHub Copilot，Claude Code，OpenAI Codex，Gemini CLI，Google Antigravity，Cursor，OpenCodeなどがある（表1）。

重要なのは，どのツールを使うかではなく，開発者が目的，仕様，制約，検証方法をどれだけ明確にAIへ与えられるかである。「DICOMビューワを作って」ではなく，「匿名化ずみDICOMのみを対象とし，患者情報を画面やログに表示せず，CTシリーズをスライダーで閲覧できる研究用ビューワを作る」のように，利用範囲と禁止事項を含めて指示することが望ましい。vibe codingの質は，プロンプトの巧拙だけでなく，仕様書，テスト，レビュー観点を人間が用意できるかに依存する。近年は，要件，受入条件，テスト方針，禁止事項をMarkdownなどで明文化し，AIに参照させながら段階的に実装するspec-driven developmentの実践も広がっている。医療AI開発では，短いプロンプトの積み重ねよりも，仕様書とAIの対話を中心に据える方が，後の検証と監査に堪えやすい。

実装例1：DICOMビューワと自動セグメンテーション

医療AIらしい応用例として，DICOMビューワに自動セグメンテーション機能を組み込む開発を考える。この機

表1 代表的なAIコーディング支援環境と医療AI開発での使いどころ

環境	特徴	使いどころ
Visual Studio Code＋GitHub Copilot	一般的な統合開発環境（IDE）内で補完，チャット，編集，エージェント機能を利用できる	既存研究コードの改修，テスト作成，ユーザーインターフェイス（UI）作成
Claude Code / OpenAI Codex	リポジトリを読み，編集，実行，検証を支援するエージェント型環境	複数ファイルにまたがる機能追加，デバッグ，文書化
Gemini CLI / Google Antigravity	コマンドラインインターフェイス（CLI），またはエージェントを前提とする開発環境	プロトタイピング，ブラウザ操作を含むアプリ作成
Cursor / OpenCode	AIを前提としたエディタ（Cursor），またはターミナル中心のエージェント（OpenCode）	小規模アプリ，研究支援ツール，ローカル開発

〈0913-8919/26/￥300/論文/JCOPY〉