

規制の多い医療ソフトウェアの現場に見る スタートアップ企業が医療AI開発の効率と ソフトウェア品質を高めた方法とは

医療機器向けソフトウェア開発にOSS（オープンソースソフトウェア）を活用するAIメディカルサービスの事例から、開発のライフサイクルにコンプライアンスを組み込みつつ効率とスピードを高める方法を考える。



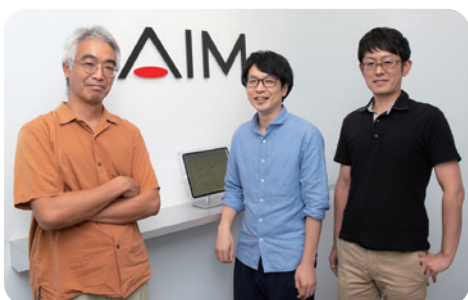
AIメディカルサービス 伊與田 正晃氏



AIメディカルサービス 遠藤有真氏



AIメディカルサービス 山本篤史氏



2017年に設立されたAIメディカルサービスは、AI（人工知能）技術を組み込んだ内視鏡画像解析ソフトウェアの開発に取り組むスタートアップ企業だ。内視鏡の画像診断は経験豊富な医師であっても難しい。同社が開発中のソフトウェアは、病変の見落としや見逃しのリスクをゼロに近づけることを目指して、内視鏡検査の画像から「がん」につながる恐れのある部位を検出し、医師の診断を支援するものだ。

AI技術と医療という異なる領域にまたがるチャレンジには、さまざまな難しさがある。医療機器としての品質が求められるのはもちろん、安全規格や法規制にのっとった開発を進める計画性も必要だ。同社はスタートアップ企業のため、収益を確保するために開発スピードも重視しなければならない。

内視鏡画像のAI解析は未開拓かつ厳格な規制がある領域だ。そのため開発においては新しい技術を応用したり新しいツールを試したりすると同時に、規格の順守とリスクマネジメント、脆弱（ぜいじゃく）性管理なども欠かせない。このような条件を満たしつつ、開発スピードと効率性を維持するためにAIメディカルサービスが選んだ開発ツールは何だったのだろうか。

内視鏡画像解析ソフトウェアの開発における品質とスピードの両立を「Coverity」で実現

AIメディカルサービスは複数の医療機関と共同で、がんやその他の疾患の病変を正確に検出できる画像解析AI技術の研究を進めつつ、そのAI技術を搭載した内視鏡画像解析ソフトウェアの開発を進めている。内視鏡画像解析ソフトウェアの開発を担うのが、同社ディビジョンマネージャーの伊與田 正晃氏が率いる製品開発部門のソフトウェア開発チームだ。

製品開発部門にはAIメディカルサービスの挑戦に魅力を感じ、医療機器開発とは畑違いの分野から転職してきたメンバーもいる。メンバーを増やし開発体制を整える中で不可欠だったのが、ソフトウェア品質を保つプロセスの整備だった。

医療ソフトウェアの開発工程は、一般的なシステム開発と大きな違いはない。コーディング規約を決め、静的解析やユニットテストを実施するといった具合だ。それらに加えて「ISO 13485」（医療機器の品質マネジメントシステム）や「IEC 62304」（医療機器ソフトウェアライフサイクルプロセス）など、医療機器の品質を保証するための国際規格に沿って開発する必要がある。「細かく計画を立ててその通りに開発・検証を進めていることを説明できなければならず、そこが医療ソフトウェア開発の大きな特徴の一つでもあります」（伊與田氏）

こうした経緯から、AIメディカルサービスはソフトウェアの品質を保証するための作業を自動化する必要があると考え、静的解析ツールの導入を決定した。そこで選択したのは、日本シノプシスの「Coverity」だ。すでにさまざまな導入実績があったことや問題検出の精度の高さが決め手となり、2019年1月に導入を完了した。

AIメディカルサービスはプログラミング言語の「C++」で内視鏡画像解析ソフトウェアの開発を進めている。開発作業全体はウォーターフォール方式で詳細に仕様を決め、開発プロセスに沿って進めている。開発プロセスによっては、オープンソースのサーバソフトウェア「Jenkins」を用いて継続的インテグレーション（CI）を取り入れている。生成したソースコードを「GitHub」にコミットするとCoverityによる静的解析が稼働し、そのプロセスを経ないとマージできない仕組みになっている（図1）。開発者がソースコードに編集を加えるたびにCoverityの静的解析が動くためすぐに間違ったところが分かり、誤ってバグなどを取り込む恐れをなくすことができた。

AIメディカルサービスの製品開発部門ソフトウェア開発チームでマネージャーを務める遠藤有真氏は次のように話す。「静的解析によって機械的にソースコードを確認することで人間が細かなところまで確認せずに済み、その分開発作業に集中できるようになったと感じています」

同じくソフトウェア開発チームに所属する山本篤史氏は、Coverityのメリットを次のように話す。「Coverityの解析によってメモリの二重解放などの問題を正確に見つけ、早期につぶすことができました。ソフトウェアをユーザーに提供した後にこうしたバグが判明すると、問題の箇所を特定して修正するのは大変です。無駄な時間や労力を使わずに済んでいます」

山本氏はCoverityが出力するレポートの分かりやすさも評価している。「『ここではこういう問題が起き得るから、こういう風に書き換えた方がいいね』というように、ソフトウェア開発チーム内でもより良いソースコードの書き方を考えるきっかけにもなっています」(山本氏)

手作業では不可能なOSSの脆弱性管理には「Black Duck」を活用

AIメディカルサービスはOSS(オープンソースソフトウェア)の管理も課題としていた。内視鏡画像解析ソフトウェアの開発環境は「Linux」を採用している他、機械学習フレームワークの「PyTorch」やアプリケーションフレームワークの「Qt」など、さまざまなOSSを活用している。

医療機器開発にはOSSを使わないという姿勢のベンダーもあるが、AIメディカルサービスは開発効率やスピードを向上させるためにOSSを積極的に使っている。伊與田氏は次のように話す。「特にAIシステムは、全ての部品を自分たちでーから作ることは現実的ではありません。ただしOSSを活用するには、ライセンスや脆弱性を効率的に管理できる仕組みが必要だと考えました」

OSSの利用時は各ソースコードやライブラリのライセンスを把握して違反しないようにする必要があり、日々新たに発見される脆弱性を解消していかなければならない。一般的には脆弱性情報を取得するためのツールや仕組みを作成し、定期的に更新できるような仕組みを自分たちでつくる必要がある。しかしこれは工数が掛かる大掛かりな作業になってしまう。そこでAIメディカルサービスはOSSの管理にもツールを活用し、自動化を進めた。

あらためて市場を調査し、トライアル後に採用を決めたのは、日本シノプシスの「Black Duck」だった。C++のソースコードを管理対象にできることが採用の決め手になったという。伊與田氏は、Black Duckの利用料金とサポート体制も高く評価している。医療ソフトウェアの開発プロセスには、開発に利用するツールのリスクマネジメントを実施して妥当性を確認するフェーズが含まれている。「ツールに不明点や不具合があったときに、サポート窓口に聞いても解決できない問題があっ

てはなりません。技術的な事柄も含めていつでも相談できる点が大きな安心感になりました」(伊與田氏)

AIメディカルサービスは、2019年10月にBlack Duckの利用を開始した(図2)。「ツールなしにOSSの管理はできないと思っています。どのソフトウェアがどのライセンスを採用しており、どのようなリスクがあるかを示してくれるBlack Duckを今後の管理にも活用したいと考えています」(遠藤氏)

世界的に実績のあるツールがアピールに IEC 62304取得を経て挑戦を継続

AIメディカルサービスが開発中の医療ソフトウェアは、2021年6月28日付でIEC 62304(医療機器ソフトウェアライフサイクルプロセス)およびIEC 62366-1(医療機器ユーザビリティエンジニアリング)の認証を取得した。特にIEC 62304の適合について遠藤氏は「Coverityのような世界的に実績のあるツールを使っているところが安全性のアピールにつながっています」と信頼を寄せる。

この認証取得を一つのステップにして、AIメディカルサービスは引き続き医療とAIという2つの領域を掛け合わせ、最先端の領域に挑戦し続ける。CoverityやBlack Duckといったツールを使い、バグを減らしセキュリティを高めつつ効率の良い開発を進めるといふ。「顧客や社会の潜在ニーズをくみ取りながら新たな内視鏡画像解析ソフトウェアを開発し続け、医療分野だけでなく社会全体のAI技術活用促進に貢献したいと考えています」(伊與田氏)

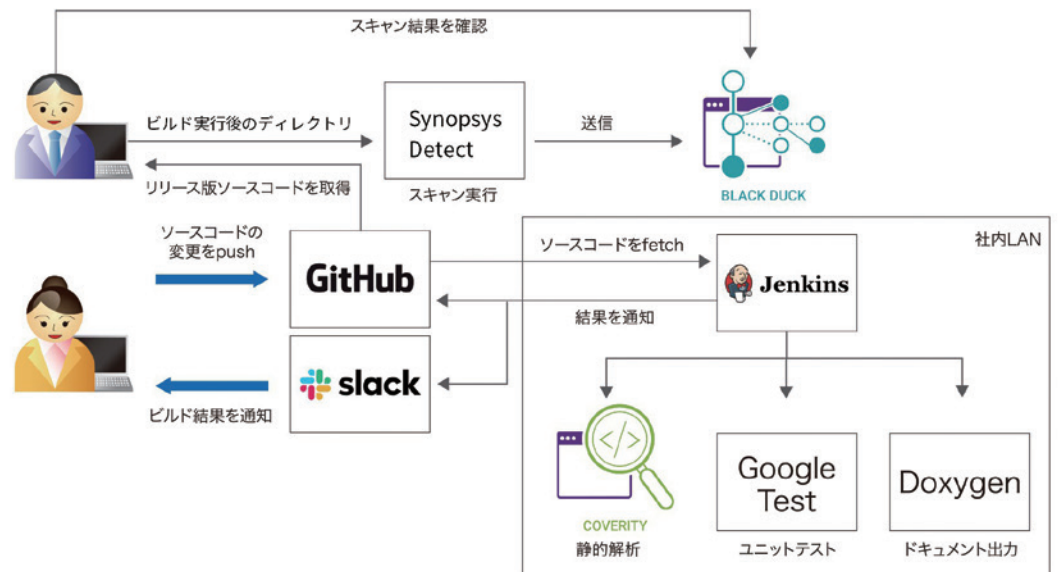


図1 AIメディカルサービスのソフトウェア開発における品質管理の仕組み(AIメディカルサービス資料を基に編集部で作成)



図2 「Coverity」と「Black Duck」が担う工程(AIメディカルサービス資料を基に編集部で作成)



<https://www.synopsys.com/jp/software>

日本シノプシス合同会社ソフトウェア インテグリティグループ

〒158-0094 東京都世田谷区玉川2-21-1 二子玉川ライズオフィス

TEL: 03-6746-3600 Email: sig-japan@synopsys.com

©2021 Synopsys, Inc. All rights reserved. Coverity は Synopsys, Inc. の登録商標です。その他の会社名および商品名は各社の商標または登録商標です。