特集論文

DevOps×生成AI×セキュリティーを搭載したソフトウエア開発基盤“MelSecOps”

Development Infrastructure Equipped with DevOps, Generative AI and Security “MelSecOps”

尾﨑紀之* *Motoyuki Ozaki*
長峯　基‡ *Motoi Nagamine*
小森裕之† *Hiroyuki Komori*
梅原友幸* *Tomoyuki Umehara*

*設計技術開発センター
†電力システム製作所
‡設計技術開発センター（博士（情報科学））

要　旨

近年，事業環境が大きく変化し，需要予測に基づく見込み生産から市場環境の変化に追従できるフレキシブルなものづくりへの変革が求められており，開発リードタイムの短縮が重要な事業課題の一つになっている。しかし，多くのソフトウエア開発現場では技術的負債の蓄積や開発環境の陳腐化が足かせになり，現場独力では改善が進まない現状があった。そこで三菱電機は，DevOps(注1)，生成AI，セキュリティーなどに関する各種ツールを備えた当社グループでの統一したソフトウエア開発基盤“MelSecOps”を構築しグループ全体に展開することで，開発リードタイムの短縮を推進している。

（注1） ソフトウエアの開発リードタイム短縮・品質向上等のために，開発担当者と運用担当者が協力する手法や仕組み

1. ま え が き

近年，顧客ニーズの多様化，技術革新の加速，政策・規制の変動など，市場環境の変化が激しくなっている。このような状況の中，当社グループでは，グループ内外の知見の融合と共創によって，進化した統合ソリューションを提供する循環型 デジタル・エンジニアリング企業へ変革し，多様化する社会課題の解決に貢献することを目指している。そのためには様々なデータを集約・分析し潜在的な顧客要求を把握することと，要求を製品やサービスなどの形で早期に具現化することが重要である。このように，新製品・新サービスをいち早く市場に投入し，市場からのフィードバックを受けて迅速に適応することは顧客価値の最大化につながる。

これらの理由から，開発のリードタイムを短くすることは，重要な事業課題の一つになっている。しかし，当社グループ内の多くのソフトウエア開発現場では改善が進んでいない現状がある。これは，ソフトウエア開発業務の負荷が高く，自動化等の改善や新技術の導入に向けた時間確保が困難なためである。また，一度自動化等の改善の仕組みを導入してもそれを保守・維持する負担が重く，開発環境が陳腐化してしまっている現場もある。これらの課題を解決するため，当社グループでの統一したソフトウエア開発基盤MelSecOpsをコーポレート組織で構築し当社グループ全体に提供することで，ソフトウエア開発のリードタイム短縮を図る。

本稿では，MelSecOpsで提供する各種サービスの概要及びソフトウエア開発現場への適用事例を述べる。

2. MelSecOpsの概要

この章では，MelSecOpsが提供する各種サービスを述べる。MelSecOpsは当社グループでの統一したソフトウエア開発基盤であり，DevOps，生成AI，セキュリティーの三つのサービスで構成される。これらのサービスには，当社グループ独自のノウハウや工夫を織り込んでいる。

2. 1節では，サービスの全体像を述べる。**2. 2節**～**2. 4節**では，それぞれDevOps，生成AI，セキュリティーについての取組みを述べる。

2.1 全　体　像

図1にMelSecOpsが目指すソフトウエア開発の姿を示す。MelSecOpsでは，ソフトウエア開発の各種作業を支援するツールを提供することで，開発リードタイムの短縮を狙っている。支援対象は設計・実装・テスト等のエンジニアリングプロセスに限らず，プロジェクト管理を支援するツールも提供する。例えば，進捗・課題・品質の可視化や生成AIによる支援を通じて，プロジェクト状況のタイムリーな把握や迅速な意思決定をサポートする。

これらの支援を実現するためのMelSecOpsのシステム構成を**図2**に示す。MelSecOpsはクラウドベースの開発基盤で

あり，リポジトリー，DevOps(チケット管理，ダッシュボード，パイプライン)，生成AIやセキュリティーツールから成る。さらに，オンプレミス環境にあるツールとの連携やナレッジベースのような他環境との連携も担う。クラウドベースにした主な理由は，サーバー等の準備が不要なためユーザー(グループ内のソフトウエア開発部門)に迅速に環境を提供できること，従量課金のため初期投資を小さく始められることの二点である。各ツールは個々に独立して利用可能であり，ツールの価値を更に高めるために，ツール間の連携機能も開発予定である。具体例は各項目で述べる。

2. 2節以降で，各サービスとそれを実現するツールの詳細を述べる。

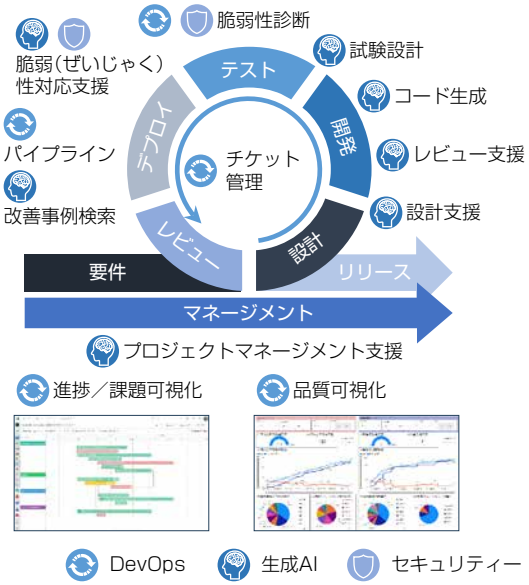


図1－MelSecOpsが目指すソフトウエア開発の姿

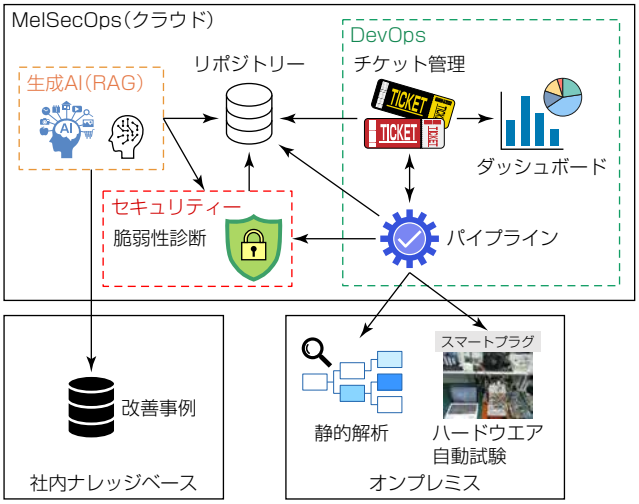


図2－MelSecOpsのシステム構成(概念レベル)

2. 2　DevOps

DevOpsでは次のツールを含むSaaS(Software as a Service)(注2)を提供する。また，開発部門がSaaSを利用するに当たって，初期設定等の環境導入のハードルが高いという事情から，標準・推奨設定やサンプルを併せて整備した。

(注2)　サービス提供事業者のサーバーでソフトウエアが動作し，そこにユーザーがアクセスして利用する形態のアプリケーション

2. 2. 1　チケット管理

タスクや障害対応の進捗や工数などを記録するツールである。当社グループではタスク管理や問題管理に表計算ソフトウエアを利用している開発現場が多く，プロジェクト管理が非効率であるという問題があった。チケット管理システムを活用することで，プロジェクト全体と詳細なタスクの進捗把握の両立が容易になる，変更履歴を追跡できる，変更時に通知を受信できる，などの利点が得られる。一方，チケット管理システムを新規に導入するには，チケットの設定(入力項目やステータスの遷移など)の検討が必要であり，敷居が高いという課題があった。そこでMelSecOpsでは，システム自体に加えて，チケットの定義，権限設定，進捗把握のための表示設定(例：遅延の警告表示)などを含む標準設定及び設定を適用するための補助ツール・手順書を提供する。これらは，当社グループのソフトウエア開発現場でのベストプラクティス等のノウハウを基に作成している。例えば，障害対応のチケットには検出工程，混入工程，混入原因，手戻り(対応)工数など，ソフトウエアの品質評価や改善施策検討のために開発現場でよく用いられている情報を入力項目に含めた。同様に，レビューのチケットにはレビュー対象の分量とレビュー時間，その子チケットであるレビュー指摘対応のチケットには指摘への対応工数の入力項目を設けた。

2. 2. 2　ダッシュボード

チケット管理システムに入力した情報を集計し，プロジェクトの進捗状況やソフトウエアの品質情報を可視化するツールである。チケット管理システムでも一部の集計や可視化は可能だが，プロジェクト管理のための十分な機能は備わって

いなかった。そこで，当社グループのソフトウエア開発でよく使用される指標・図表を自動で集計・表示する仕組みを構築した。**2. 2. 1項**のチケット管理システムで述べた標準設定を利用している場合，カスタム不要でそのまま利用できる形で提供する。例えば，レビュー指摘や障害による手戻り工数，テスト密度・障害密度，障害の検出・解消件数のデイリーでの変化，タスクの進捗を予定と実績で比較するガントチャートなどの集計が可能である。

2. 2. 3　パイプライン

ビルド，テスト，デプロイ等の定型的な作業を自動化するツールである。自動化が進んでいない開発現場や，自動化環境は存在するがサーバー等の保守が負担になっている開発現場を対象に，クラウドベースの自動化の仕組みを提供する。パイプラインを利用するためには専用のスクリプトを記述する必要があり，開発現場で導入する際の障壁になり得るという課題があった。そこで，パイプラインで自動実行することが多いと見込んだ処理について，スクリプトのカタログを提供する。カタログでは，再利用しやすい粒度でスクリプトのテンプレート(部品)とその利用方法をまとめた。ユーザーが自動化したい内容に応じて，カタログに掲載した部品を選択し組み合わせることで容易にパイプラインを構築できるようにして，導入の敷居を下げた。カタログに掲載した部品の例を**表 1** に示す。

表 1 -パイプライン部品例

パイプライン部品名	内容
ビルド	アプリケーションをビルドする
単体試験	単体試験を実施する
コード静的解析	コードの静的解析を実施する
成果物リポジトリー登録	成果物をリポジトリー登録する
デプロイ	成果物をデプロイする

パイプラインは，開発現場が持つ既存の自動化環境と統合する機能を持つ。**2. 1節**で述べたとおりMelSecOpsはクラウドベースでサービスを提供するが，開発現場が持つ自動化環境はオンプレミスに構築することが多い。既存の自動化環境をクラウドに移行するためには追加の開発コストを要するため，オンプレミスの環境をそのまま利用できることに対するニーズは高い。また，ハードウエアを伴う自動試験環境や，ノードロックのライセンス(特定の機器での使用が許可されるライセンス形態)で持っているツールなど，クラウドへの移行が困難な環境も存在する。このような既存の環境を有効活用するために，クラウドとオンプレミスを相互に接続し連携可能な機能を持つツールを選定した。

また，ソースコードリポジトリーと連携し，プルリクエスト(あるブランチでの変更内容をほかのブランチに反映(マージ)する要求)を発行する際にパイプラインの実行を強制する機能も持つ。これは，大人数の開発では各開発者によるソースコードの変更をマージする際に問題が発生する可能性が高く，マージによる誤りの混入を防ぐためである。この機能を活用することで，開発規模と品質確保を両立しやすくする。

これらのほかに将来的には，パイプラインを用いてチケット管理システムに登録したチケットの情報を定期的に生成AIで分析することで，プロジェクトの健全性チェックや遅延・リスクの兆候把握に活用することを構想している。

2. 3　生成AI

ソフトウエア開発の効率を劇的に改善するために，生成AIの活用ニーズは急速に高まっている。特に，長期間継続する事業が多い当社グループでは，これまでの製品・サービス開発で蓄積したナレッジを有効活用できるRAG(Retrieval-Augmented Generation：検索と生成AIを組み合わせて回答精度を向上させる技術)の需要が多くある。一方，各組織に生成AIを用いたアプリケーションを開発できるエンジニアが必ずしもいるわけではないため，組織ごとに開発することは非効率である。そこで，MelSecOpsでは生成AIを活用する標準化したアプリケーションを開発し(**表2**)，プロンプトの調整やRAGで参照するデータベースの作成だけユーザーごとにカスタマイズすることで，即座に生成AIを用いたアプリケーションを開発組織に導入できる仕組みを構築した。これによって，生成AIを活用したアプリケーションの開発スピードを大幅に向上させることに成功した。

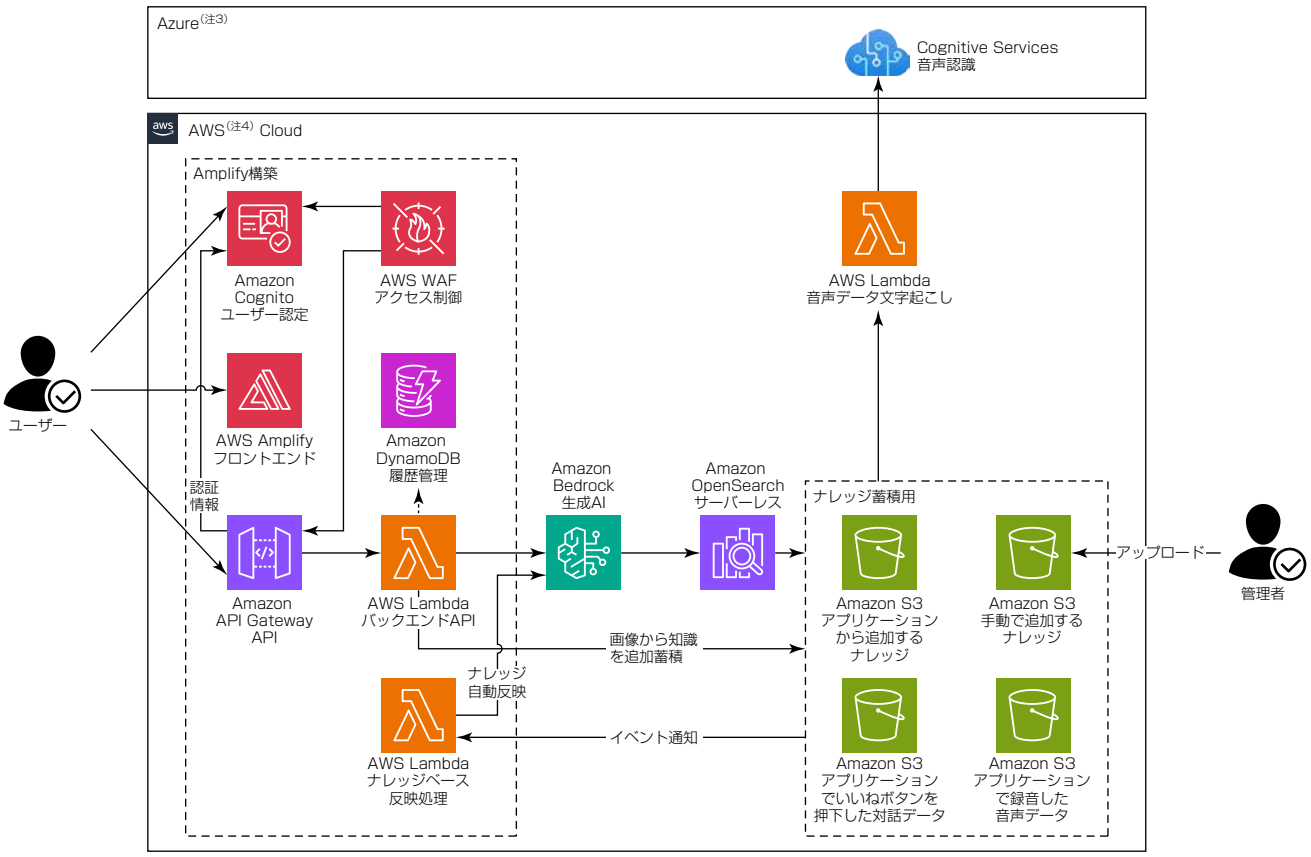
生成AIを活用したアプリケーションの中で，仕様問合せツールのアーキテクチャーを示す(**図3**)。このツールは，RAG技術を使用しており，RAG技術を用いるアプリケーションの基本構成は同様のアーキテクチャーを採用している。RAG技術を活用したアプリケーション(以下“RAGアプリ”という。)の特徴として，精度向上のために二つの機能を設け

ている。一つ目はナレッジアップロードである。この機能は，ユーザーがアプリメニューから個別にナレッジを追加できるようにすることで，不足しているナレッジをユーザーが追加し，精度を向上させることを可能にするものである（**図４**）。二つ目は“いいね機能”である。この機能は，RAGアプリへの問合せ結果に対して“いいね”ボタンで応答できるもので，このユーザーの応答情報を次回の問合せ時に活用することで，有効な回答を優先的に生成することを可能にしている。

また，改善事例を格納したグループ内ナレッジベースを生成AIで検索するツールも持つ。当社グループでは，数多くあるソフトウエア開発現場の知見を有効活用するために，開発でのベストプラクティスをグループ内に広く共有するWebサイトを持っている。しかし，サイトの検索性が問題で利用者が伸び悩んでいた。そこで，探したい情報をプロンプトで入力すると当該Webサイト内を検索し，蓄積したナレッジの中から類似している情報を提示するRAGアプリを構築した。RAGアプリの公開を含めたリニューアルの結果，サイトの利用者数・訪問回数共に約２倍に増加した。

表２-MelSecOpsが提供する生成AIを活用したツールの一覧（開発中，開発予定のものを含む）

ツール名	内容	効果
作業見積り	過去案件の工数実績等を参考に，作業工数を見積もる	・プロジェクトの遅延リスクの低減 ・再計画のリスク低減
仕様問合せ	仕様書やマニュアルに記載されている文章の要約。問合せへ自動応答する	・ドキュメント確認の工数削減 ・問合せ対応の工数削減
設計書検索	開発内容から変更が必要になる対象機能を記載した設計書をデータベースから検索する	・設計検討の工数削減 ・ナレッジ共有の効率化
レビュー支援	過去のレビュー指摘内容を基に，ドキュメント（設計書，仕様書）をレビューする	・レビューの工数削減 ・不具合の再発防止
試験設計	仕様書を基に，試験ケース・試験手順を生成する	・試験設計の工数削減 ・テストカバレッジ（網羅率）の向上
脆弱性対応支援	ソースコードを解析し，解析の結果から脆弱性に対するソースコードの修正案を提案する	・脆弱性対応の工数削減
プロジェクトマネージメント支援	過去のプロジェクト実績・事例と現在のプロジェクトの状況を基に，リスクを抽出する	・リスクの早期発見 ・遅延の防止
改善事例検索	当社グループのWebサイトを対象に，ユーザーにマッチするソフトウエア開発の改善事例を検索する	・良好事例の展開促進



（注３）　Azureは，Microsoft Corp.の登録商標である。
（注４）　AWSは，Amazon Technologies，Inc.の登録商標である。

図３-仕様問合せツールのアーキテクチャー

図４-仕様問合せツールの画面例

2.4　セキュリティー

近年，サイバー攻撃の多様化，ネットワークに接続する組み込み機器の増加，サイバーセキュリティーに関する各国の法令・規制の整備などに伴って，製品のセキュリティー確保のニーズが高まっている。そこでMelSecOpsでは，脆弱性診断ツールを先に述べたパイプラインで自動実行する機能を提供する。当該ツールでは，ソースコード，コンテナの設定，依存関係のあるオープンソース等に脆弱性が含まれていないかを自動で確認できる。パイプラインを活用しソフトウエアのビルドやテストと同時に脆弱性を診断することで問題を早期に発見できるため，セキュアな製品・サービスの提供につなげることができる。また，チケット管理システム及びダッシュボードと連携し，脆弱性への対応状況を管理することも可能である。

さらに，脆弱性診断ツールによる指摘内容を生成AIに入力し，ソースコードの修正案を提示することで脆弱性対応を効率化できないか技術検証中である。脆弱性診断ツールが出力する解析結果であるSARIF(Static Analysis Results Interchange Format)ファイルに含まれる膨大な情報は，人間が理解して修正対応をするのに工数がかかるという問題がある。そこで，ソースコードとSARIFファイルを特定のプロンプトとともに生成AIに入力することで，指摘内容に補助的な情報を付与して理解を助けたり，ソースコードの修正案を提示したりできないかを検討している(**図５**)。現在，プロトタイプによる検証が完了し，特定の事例に対して情報の付与や修正案の提示が可能であることが分かった。今後は，実製品のソースコードを対象に実証した上で，グループ内で広く活用できる汎用的なツール化を目指す。

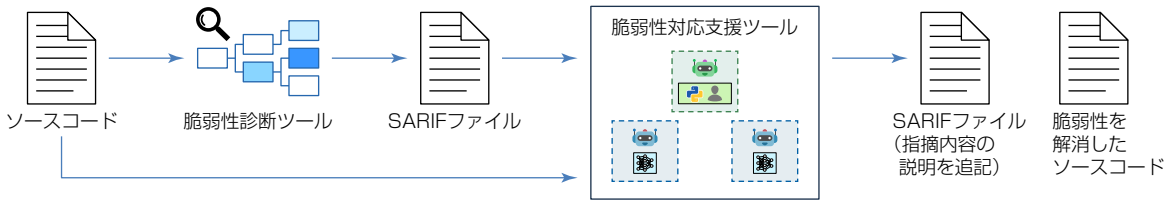


図５-生成AIを活用した脆弱性対応支援ツール

3.　導入支援サービス

この章では，当社グループの開発現場で，開発効率が低下することなくスムーズにMelSecOpsを導入できるサービスについて述べる。

3.1　サービス概要

当社グループのソフトウエア開発では，各製作所で独自の開発プロセスやルールが存在する。このため，MelSecOpsを現場主導で導入すると，現場の負担が大きくなり，導入に時間を要することが予想された。そこで，導入を効率的に進めるために，開発現場のプロセスや課題を個別に分析し，効果を迅速かつ最大限に享受できる形で導入する専属チームを

作って，専属チームによる導入支援サービスを提供した。

また，MelSecOpsは，DevOps，生成AI，セキュリティーの主要ツールを全て導入する必要はなく，個別にDevOpsだけや生成AIだけなど，取捨選択して導入することも可能にしている。この導入ポリシーの意図は，MelSecOpsを全て導入しようとすると，開発現場で混乱が生じて，導入までの準備や検討に多くの時間を要することが考えられるためである。このようにツールの部分導入を可能にすることで，利用価値をスピーディーに判断できるようにして，段階的に導入範囲や導入ツールを拡大することを狙いとしている。

3. 2節以降で，二つの製作所にそれぞれ必要なツールを導入した際のサービス事例について述べる。

3. 2　DevOps導入事例

この節では，MelSecOpsの主要サービスであるDevOpsを導入した事例について述べる。この導入事例の開発現場では，元々ウォーターフォール型の開発プロセスを採用しており，顧客の開発要求に迅速に対応できず，市場への製品リリースに時間がかかるという課題を抱えていた。この課題を解決するために，DevOpsを導入したものである。

DevOpsの導入に際して，現行の開発プロセスを詳細に分析し，DevOpsの効果を最大限に享受するために，開発全体はウォーターフォール開発を維持しつつ，各開発項目に対する設計から実装及び単体試験はアジャイル開発を採用するハイブリッドな開発プロセスを設計した。このハイブリッドな開発プロセスを実現するために，各開発フェーズでのコードの構成管理と実行されるパイプラインを定義した“ブランチ戦略”を設計した(**図6**)。このブランチ戦略によって，個別の開発要求ごとの実装作業を並列して進めることが可能になり，ソフトウエア全体を組み合わせた総合試験の評価フェーズでも，機能ごとに品質を担保したソフトウエアを集約することで効率的な開発が可能になった。

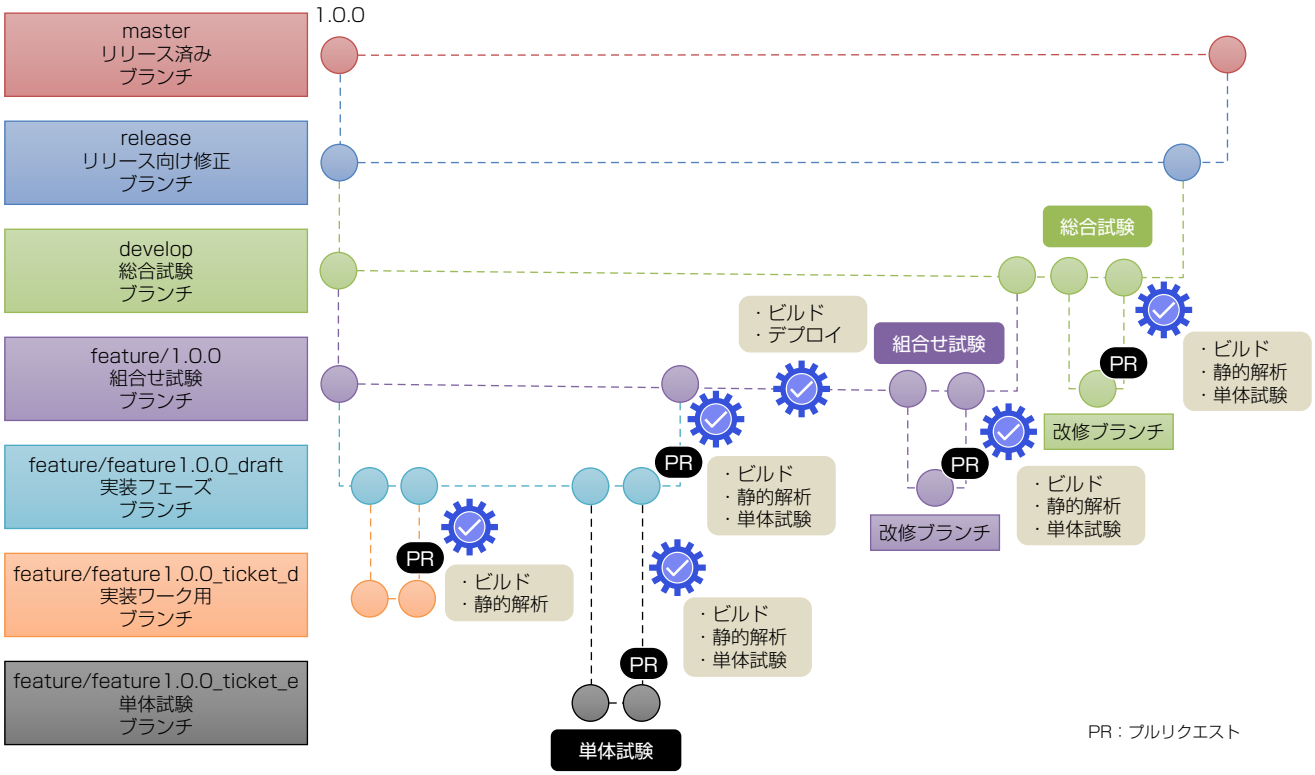


図6-開発プロセスに応じたブランチ戦略

このブランチ戦略を支援するため，**2. 2節**で述べたパイプライン部品を組み合わせることで当該製作所向けのパイプラインを整備した。例えば，コードの修正をトリガーにビルド，コードの静的解析，単体試験を実行するパイプラインを準備している。

これによって，開発チームはDevOps上に開発環境を移行し，開発要求事項をチケットで管理し，ソフトウエア開発でのコードのビルド，静的解析，単体試験をパイプラインで自動実行することで，開発工程を効率化した。

3.3　生成AI導入事例

この節では，MelSecOpsの主要サービスである生成AIを導入した事例について述べる。この導入事例は，組み込み機器開発で，制御仕様を変更する際に，複数の制御仕様書から変更内容及び影響を受ける制御仕様を抽出するのに時間がかかっていた問題を解決するものである。この問題を解決するために，**表２**に示す生成AIを活用した仕様問合せツールで制御仕様書を横断的に検索するRAGアプリを導入した。

一般的に，RAGアプリの導入では，導入後に現場で活用可能な精度が出ないことが課題になるため，導入支援サービスでは業務で活用可能なレベルまで精度を向上させている。具体的には，RAGアプリが参照するデータベースに格納されるファイルなどの情報をそのまま格納するのではなく，生成AIが目的に応じて扱いやすい構造化データに前処理を行ってから格納する。構造化データにする前処理の一例として，RAGアプリが参照するPDF(Portable Document Format)ファイルはOCR(Optical Character Recognition)(注5)で処理した上でマークダウン形式に変換し，さらに参照するデータの目的に応じてチャンクサイズを調整することで，生成AIが回答を導出しやすくする(**図７**)。これによって，RAGアプリの正答率が50%以上改善した。さらに，ユーザーインタビューによる定性評価と，RAGAS(RAG Assessment System)による定量評価によって，プロンプトチューニングを繰り返すことで，検索の正答率を当初の50%から70%以上に向上させた。

(注５)　画像から文字を認識しテキストデータに変換する技術

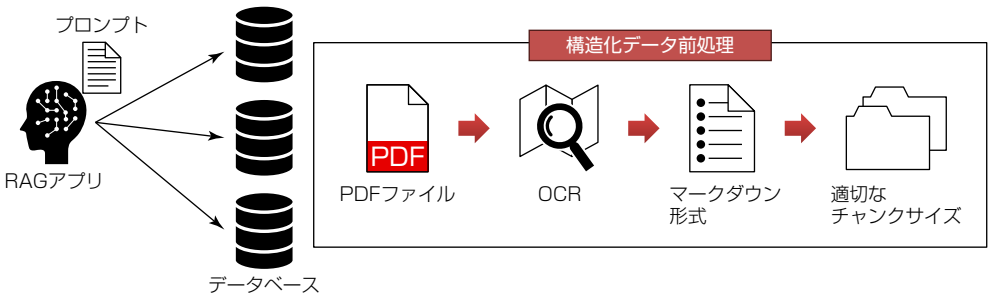


図７-構造化データ前処理例

4. む　す　び

当社グループでの統一したソフトウエア開発基盤であるMelSecOpsについて述べた。MelSecOpsではソフトウエア開発の各種作業を支援するツール群を提供することで，ソフトウエア開発のリードタイムを短縮できる。

今後は，提供するツールの拡充及び既存ツールの機能追加を通じて，更なるリードタイムの短縮を狙う。また，MelSecOpsが開発現場に普及するためには，効果的なツールを提供するだけではなく，導入をサポートすることも重要である。そのため，**３章**で述べたような，開発現場と伴走し，課題把握，改善の提案，開発の特性に合わせたMelSecOpsの設定・カスタマイズ等を行う導入サービスを拡充する。