組み込みソフトウェア開発における自動テスト環境構築による開発効率化

HMI ソリューション事業本部 ソフト基盤開発部 第一開発室 水野 創太
第五開発室 大舩 剣士朗
第一開発室 棟本 剛

1. 取組背景

昨今、ソフトウェア開発では「継続的インテグレーション・継続的デリバリー（Continuous Integration/ Continuous Delivery、以降 CI/CD という）」と呼ばれる開発手法がよく用いられます。CI/CDとは**図 1**のようにソフトウェアの変更（コミット）からビルド→テスト→デプロイ→リリースといった一連のプロセスを自動化することによって、開発効率や品質を向上させる手法です。

当社では、この CI/CD をカーナビゲーションシステム(以下、カ-ナビという) の組み込みソフトウェア開発に用いています。しかし、実際の製品上での動作を確認するには、試験用のプロダクトボードに最新ソースコードでビルドしたソフトウェアを書き込んでテストをする必要があります。このような、組み込みソフトウェア開発特有の問題により、一般的な CI/CD フローでは自動化できないプロセス（音出力の確認及び映像出力の確認など）が存在しました。今回、我々は、この課題を解決した自動テスト環境を構築し、不具合発見の早期化による品質向上と開発効率の向上を実現しました。この自動テスト環境構築の取組について紹介します。

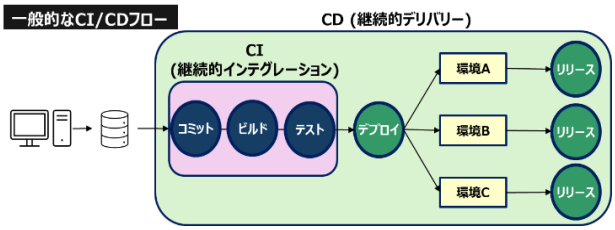

図 1 一般的な CI/CD フロー

2. 取組内容（システム構成）

図 2 に自動テスト環境を含めた当社のカーナビの組み込みソフトウェア開発の CI/CD のフローを示します。機種・仕向けごとに異なるテスト環境に適応するため、一般的な CI/CD フローからテスト工程の位置を変更しています。

(1) 毎日午前 0 時に最新状態のソースコードを使ってソフトウェアのビルドを実施
(2) 国別の仕向けや機種ごとの複数の異なる環境で自動テストを実施

今回 CI/CD プロセスを”Jenkins”というツールを利用した環境構築を行いました。Jenkins は 1,000 を超える豊富なプラグインが提供されており、CI/CD の目的に応じて様々な機能を活用でき、ソフトウェア開発現場で利用率の高い、”Git[*(1)]”や”AWS[*(2)]”といったツール類との連携ができるといった利点があるため採用しました。

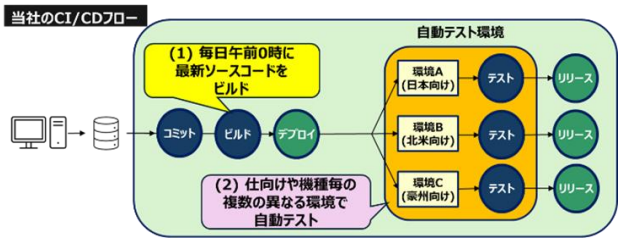

図 2 カーナビソフトウェア開発 CI/CD フロー

図 3 に自動テスト環境の構成を示します。AWS サービスを利用して、クラウド上に構築した Jenkins をインストールした Master サーバーからテスト環境用に用意した Windows PC を Slave（agent）としてコントロールすることで事前に設定したテストを実行します。今回、自動テスト環境の構築にあたって、拡張性を重視したシステム構成にしました。**図 3** の自動テスト環境は機材を用意すれば、環境数を自由に増やすことができます。カーナビのソフトウェア開発では、国別の仕向けや機種毎の複数の製品を同時に開発しています。**図 3** の

自動テスト環境は機材を用意すれば、環境数を自由に増やすことができ、複数の製品を同時にテストできるため、効率化が図れます。

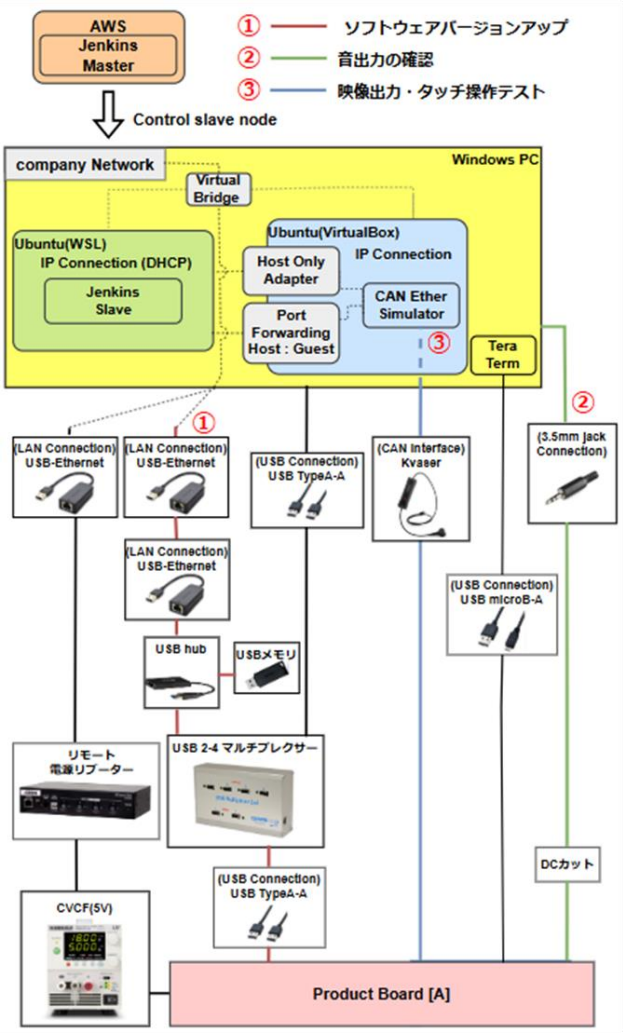

図 3 自動テスト環境のシステム構成

3. 取組内容（テストの自動化）

自動テストで実施する主なテスト項目の内容を紹介します。

3.1 ソフトウェアバージョンアップ

毎日午前 0 時にビルドした最新ソフトの動作確認テストを行います。そのためにはプロダクトボードに書き込まれたソフトウェアを最新バージョンに更新する必要があります。そのバージョンアップの実施と成否確認、正常動作確認の自動化を行います。

プロダクトボードのソフトウェア更新はビルドで作成したバージョンアップ用ファイルを USB メモリに入れてターゲットボードに接続することで実行されます。データの Windows PCとターゲットボードをUSB-Ethernet*(3) で LAN 接続し、ターゲットボードに接続した USB メモリに、バージョンアップファイルを転送することでソフトウェアをバージョンアップします。（**図 3** の①のデータ送信経路）

3.2 音声出力テスト

重要なテスト項目として、音声出力テスト（音声ファイル再生時に想定どおりの音声が出力されているかどうかの確認）があります。これはカーナビソフトウェアの機能として、音楽再生や、ラジオ、カーナビ案内音声などの音を再生する機能が多く存在するからです。

通常はスピーカーに接続して音が出るか確認しますが、今回は**図 3** の②のデータ送信経路を使い、テストの自動化環境の仕組みを構築しました（**図 4**）。テストの流れは以下のようになります。

(1) 音声読み上げサービス「VOICEVOX」を利用して事前に用意したテキストⒶから音声ファイルを作成し、ボードに転送
(2) ボード上で音声ファイルを再生し、出力された音情報を PC に取り込み、録音してデータ化
(3) 音声認識 AI サービス「Whisper」を使って文字起こしをしたテキストⒷを作成
(4) テキストⒶとテキストⒷを比較し、文字列が一致しているかを確認

これにより想定どおりの音声が出力されているかを確認することができます。

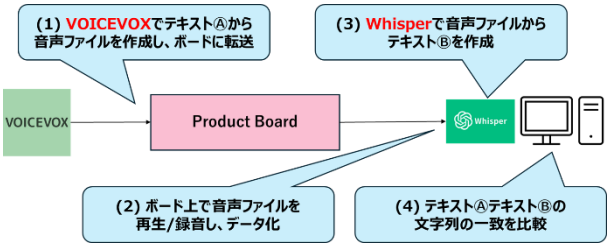

図 4 音声出力テストの仕組み

3.3 映像出力・タッチ操作テスト

他の重要なテスト項目として、映像出力テスト（ディスプレイに想定どおりの映像が出力されているかどうか）、ディスプレイタッチテスト（ディスプレイタッチ時に想定どおりの画面遷移が行われているかどうか）があります。これはカーナビソフトウェアの機能として、地図情報の表示やテレビの視聴など、ディスプレイに情報を表示して利用する機能が多く存在するからです。

映像出力テストやディスプレイタッチテストでは、製品開発用に用意された CAN 情報のシミュレーション用ツールを利用します。本ツールは、CAN 情報のシミュレーションだけでなく、**図 5** のようにディスプレイに表示している映像のリアルタイム映像をキャプチャーし、ツール UI 上にキャプチャーした画面をクリックすることで実際にディスプレイをタッチした際と同じ操作をボード側に反映することができます。（**図 3** の③のデータ送信経路）

本ツールを利用して、特定操作を実行した際にキャプチャーした映像と、事前に用意した正解画像を OpenCV の「SSIM（Structural Similarity Index, 構造類似度指数）」を用いて一致率を比較します。これによりディスプレイに想定どおりの映像が出力されているか、ディスプレイタッチ時に想定どおりの画面遷移をしているかの確認ができます。

図 5 ツールを活用した画面遷移

3.4 その他テスト項目

3.1～3.3 節でご紹介したテスト以外にも、複数のテストを行っています。**表 1** に実施しているテスト項目の一覧を紹介します。**表 1** には本取組前後でのテストの自動化状況も記載しています。

本取組前は、最新ビルドソフトへのバージョンアップなどの基本的な内容は実施できていましたが、音声確認や映像出力の確認といったカーナビソフトウェアの重要機能の確認はできていませんでした。本取組によりテスト項目は大きく拡張できました。

特に、本取組前では自動化できていなかった「音声出力テスト」や「映像出力テスト」といったテストの自動化が実現しました。これにより、CI/CD プロセスによる「マージ～リリース」までの全工程が自動化できるようになり、ソフトリリース作業にかかる工数を削減することができました。

表 1 テスト項目一覧

項目	テスト内容	本取組前	本取組後
Sub/LAN/SoC バージョンアップ	各種マイコンのバージョンアップが成功すること	○	○
バージョン情報確認テスト	書き込みソフトのバージョンの情報の取得	○	○
サービス起動確認テスト	Systemd サービスが全て正常起動していること		○
CPU 使用率確認テスト	高負荷動作時に CPU 使用率が 70%以下であること		○
インストールパッケージ確認テスト	インストールされている ipk 一覧の取得	○	○
パーティションマップ確認テスト	パーティション情報一覧の取得		○
HAL API テスト	各 HAL の API テストが PASS すること	○	○
映像出力テスト	ディスプレイに想定どおりの映像が出力されること		○
ディスプレイタッチテスト	ディスプレイタッチ時に想定どおりの画面遷移が行われること		○
音声出力テスト	音声ファイル再生時に想定どおりの音声が出力されること		○
CAN 送受信テスト	CAN 送受信が成功すること		○
adb 接続テスト	adb 接続が可能であること		○

４．まとめ･今後の展開

今回構築した自動テスト環境によって、組み込みソフトウェア開発のテストが自動化できるようになりました。これにより、不具合発見の早期化による品質向上、開発工数の削減が期待できます。しかし、現状では自動化を実施できているテストは基礎的な最低限の機能のテストに限られているため、複雑な機能や特殊な機材を用いたテストについては、いまだに開発担当者によって手動で実施しています。今後、各開発担当者が手動で実施しているテスト項目の自動化を進め、さらなる開発工数の削減を図っていきます。
また、ソフトウェア開発は AI 技術をはじめとする新技術の広がりによって、さらなる作業の自動化やそれに伴う品質の向上が進むと考えられます。組み込みソフトウェア開発でも、それらの技術を積極的に取り入れ、開発技術のさらなる向上および新技術の検討を継続していきます。

商標

脚注

*(1) Git
ソースコード等のファイルのバージョンを追跡する分散型バージョン管理システムとして現代のソフトウェア開発で用いられるツール

*(2) AWS
Amazon が提供するクラウドサービス。コンピューティング、ストレージ、データベースなど多様なサービスが提供されており、環境のクラウド化による可用性や柔軟性を目的にソフトウェア開発やサービス提供に使用が広がっている

*(3) USB-Ethernet
USB 接続ポートしか持たない機器で LAN 接続を行うために、USB 接続とイーサーネット接続を中継するアダプタ