SIEMENS
Ingenuity for life

Siemens Digital Industries Software

車載組み込みソフトウェア開発のための統合型プラットフォームの構築

エグゼクティブ・サマリー

今日の自動車を支える機能や装備にはソフトウェアアプリケーションが大きな役割を果たしており、それにともなって車載組み込みソフトウェアが急速に高難度化しています。一方、ソフトウェアの重要性が高まるにつれて、ハードウェア、ソフトウェア、および物理システム間のやり取りも複雑化してきました。従来のソフトウェア開発手法では、今日の自動車のサイバー - 物理領域間のインターフェースの数や複雑さを管理できません。競争に勝ち抜くためには、ソフトウェア開発のプロセスを進化させて、ソフトウェア開発と物理システム開発をつなぐ共通のデジタル・スレッドを確立する必要があります。

Piyush Karkare
オートモーティブ・インダストリー・ソリューション　グローバルディレクター

目次

最新の自動車は、パワフルな演算ユニット、センサー、アクチュエーター、機械システム、電気機械システムからなる複雑なシステムであり、今日では、機能と装備の多くが電気的に実現されています。ユーザーは、最新自動車で提供される多くの装備をそのまま楽しみつつも、シームレスで直感的な乗車体験を求め (図1)、車載システムを管理するソフトウェアでその多くを実現する傾向が強まってきました。ソフトウェアはまた、コネクテッドカー、電気自動車、自動運転車の構想を実現して、収益に結び付けたいと企業にとっては決定的に重要です。車載ソフトウェアは高度化と複雑化の一途を辿り、その開発、検証、評価がますます難しくなってきました。

製品エンジニアリングとソフトウェアエンジニアリングは元来、別々の開発サイクルで進められます。エンジニアはソフトウェア開発を個別に管理し、通常は、あらかじめ決められたチェックポイントでソフトウェアとハードウェアのインターフェースを検証するだけです。しかし、自動車の機能性と接続性が増すにつれ、ソフトウェアと物理システム間の相互作用が複雑になったため、現行のプロセス、ツール、手法の弱点が顕著になってきました。自動車システムの包括的な視点を持たずにソフトウェア開発作業を進めると、ハードウェア適合性、ソフトウェア品質、プログラム責任追跡性に問題が生じます。

未来の自動車開発に向けての技術競争を勝ち抜くためには、ソフトウェア開発のプロセスを今すぐにでも進化させなくてはなりません。ソフトウェアシステムと物理システムを結ぶデジタル・ツインこそが、スマートなコネクテッドカーの複雑性を制御する唯一の手段と言えるでしょう。ソフトウェアシステムとハードウェアシステム双方の挙動をクローズド・ループで表示できるため、製品ライフサイクル全体を通して継続的な検証が可能です。堅牢なデジタル・スレッドは、車載ソフトウェア機能が自動車に完全に適合していること、すべての関連タスクと成果物を期限内に準備できることを示す証です。デジタル・スレッドはまた、変更が誰によって、どのように、なぜ行われたのかを追跡するうえでも有効です。

図1: 今日の消費者は、自動運転車であろうとなかろうと、自動車にシームレスなユーザー体験を期待する

課題と機能中心アプローチ

車載ソフトウェア開発は、共通の車載機能セット開発に取り組む数々の組織の複数の設計領域が関与する複雑な環境で進められます (図2)。車載ソフトウェア機能は、自動車レベルまたはプラットフォームレベルの相互作用を定義した要件を起点として、その後、必要な電気接続性とネットワーク通信機能を追加して改良されます。

アプリケーションの設計・構築段階では、それぞれのアプリケーションが大きなシステムの一部であることを忘れがちです。1つまたは複数の自動車性能を駆動するタスクを実行するため、アプリケーションが個別の車載機能を呼び出します。1つのアプリケーションで複数の自動車性能を実現することもあれば、1つの性能を得るために、複数のECUを使用する複数のアプリケーションが必要となることもあります。

後者の一例と言える車線維持支援機能ソフトウェアは、センサー、多数のプロセッサー、ステアリングアクチュエーターと連携し、運転者に処理内容を伝えなければなりません。組み込みソフトウェアアプリケーションの要件はもともと、自動車レベルのシステム設計・開発プロセスに由来しており、組み込みソフトウェアアプリケーションをシステム全体の枠組みで捉えたものです。アプリケーション開発プロセス全般を通して、システム全体の視点を持つことが非常に重要です。

通常、自動車レベルのソフトウェア機能開発を自動車OEMが担い、細かな制御アルゴリズムと組み込みソフトウェアアプリケーションはサプライヤーが開発することがほとんどです。しかし、競争の激化を受けて、OEMがインフォテイメント、ADAS、パワートレインなどの重要機能を社内開発することが増えてきました。

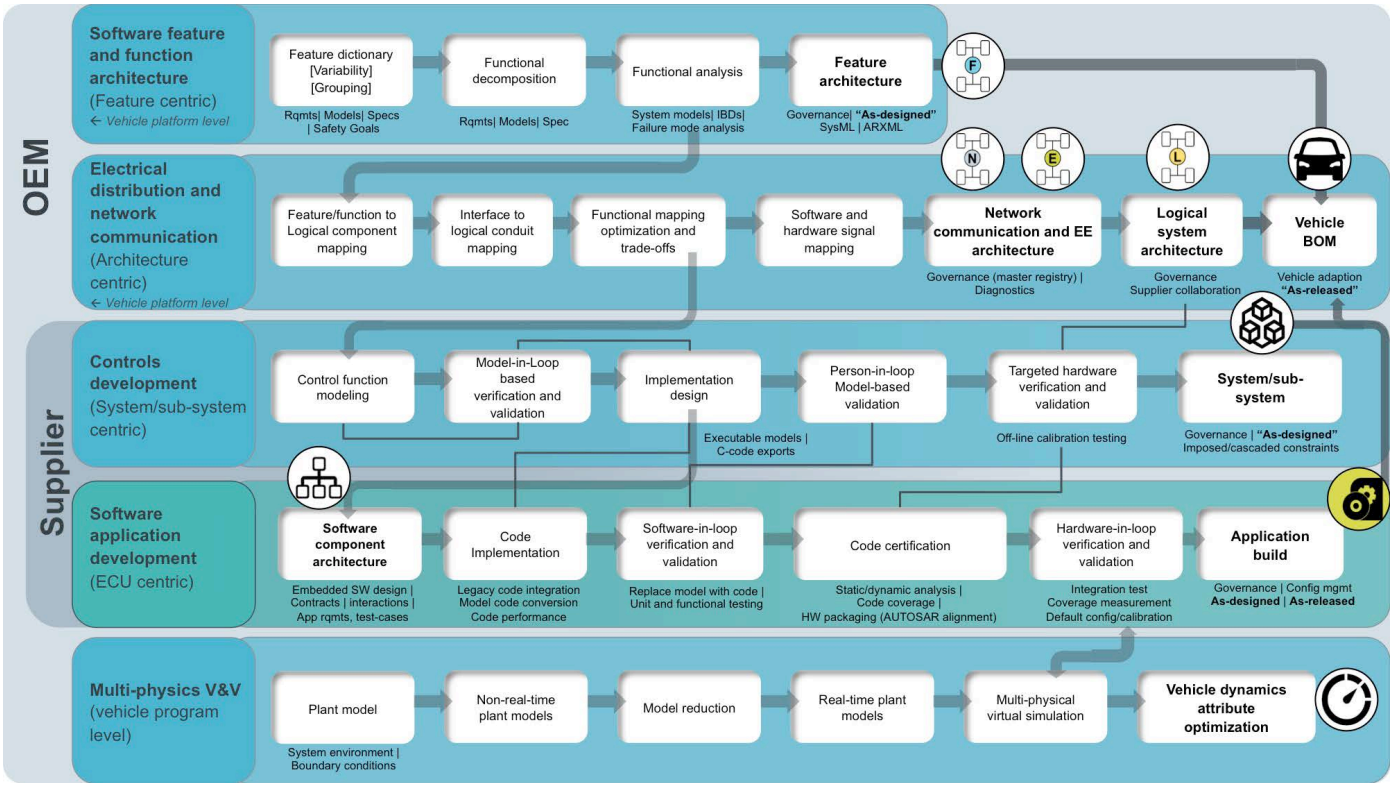

図2: 複数のエンジニアリング領域にまたがる多数の組織が関与する高度に複雑なタスクである車載組込みソフトウェア開発

車載ソフトウェアが期待どおりの機能を発揮することを検証するために必要な制御アルゴリズムは、具体的な機能動作とソフトウェアコンポーネント間の相互作用を要約したものです。ソフトウェアコンポーネントは、サイバー‐物理システムを機能させるソフトウエアアプリケーションとしてコンパイルされます。ソフトウェアアプリケーションはプラットフォームが共通の全車種に適合するよう設計されるのが普通です。

実装と操作の複雑化を受けて、各サプライヤーの責任範囲がECUのコンテンツでサポートする機能全体ではなく、自身で開発したコンテンツのみに限られるようになってきました。サプライヤーとOEMのエンジニアリング部門はいずれも、ECU (ハードウェア) 中心のアプローチを取ることが多く、実装される機能に対する注意がおろそかになりがちです。ハードウェア中心のアプローチでは、ソフトウェアやハードウェアのコンポーネントの成熟度に応じて、ソフトウェア機能を持続的に検証することになるため、非常に難易度が高く、時間もかかります。検証と妥当性評価に時間がかかることから、機能レベルのバグの発見が終盤までずれ込むことも珍しくありません。この結果、コストのかかる終盤での変更を招き、テスト期間が短くなり、製品や機能のリリースに遅れが生じます。分散して実装される機能の増加と複雑化のために、問題が見逃されたり、発見が遅れる可能性が飛躍的に高まっています。

一方、機能中心のソフトウェア定義アプローチであれば、混沌とする車載組み込みソフトウェア定義が整理されるので、システムエンジニアは、ソフトウェア機能 (装備アーキテクチャー) の分解を定義、管理し、ソフトウェアコンポーネントを論理的な電子コンポーネント (ECU、スイッチ、アクチュエーターなどの論理アーキテクチャー) に割り当て、再利用することに専念できます。また、電子接続、電源供給、グランド、ヒューズをソフトウェア機能の要件に基づいて最適化する (電子アーキテクチャー) ことも、ネットワーク通信機能を最適化し、セキュリティ保護する (ネットワークアーキテクチャー) ことも可能です。

自動車開発プログラムが立ち上がると、ソフトウエア機能と論理電気コンポーネントの間にある既存のソフトウェア-ハードウェア関係に基づき、それぞれの車載機能に必要な電気コンポーネントが特定されます。論理コンポーネントに物理部品番号を割り当てておくことにより、開発プログラムのなかでソフトウェア機能がどのように設計、使用されるのかをエンドツーエンドで追跡できます。OEMやサプライヤーで行ったすべての作業を追跡することは膨大な難作業です。効果的なコラボレーション環境で作業を進めることができれば、自動車に対する適合性と責任追跡性を備えた完全なソフトウェア機能をエンジニアが提供しやすくなります。

車載組み込みソフトウェア開発のための統合型プラットフォーム

機能中心のアプローチを効率的に実装するには、世界中のサプライヤーから納品され、いくつもの車載ECUに分散配置される複雑なソフトウェア機能を設計、作成、追跡、改良するためのロバスト性、安全性、広いアクセス性に優れた手法が必要です。

これには、車載組み込みソフトウェア開発のための統合型プラットフォームが不可欠です。そのプラットフォーム上で、多種多様なツール群で行っている作業をすべて調整して、ハードウェアおよびシステム構成の制約に対応し、検証および妥当性確認 (V&V) が完了したバージョンを完成させます。OEMやサプライヤーは、ツールチェーンのエコシステム全体のデータフローを結合、協調させ、プロセスや方法、ツールを最適に統合できます。

多様なツールセットを繋いで、組み込みソフトウェアアプリケーションの定義から計画、開発、配布にいたるライフサイクル全体のあらゆる作業を調整する統合型プラットフォームは、多くのエンジニア間の有機的なコラボレーションを推進し、トレーサビリティーを確保し、有効データを再利用しやすくします。

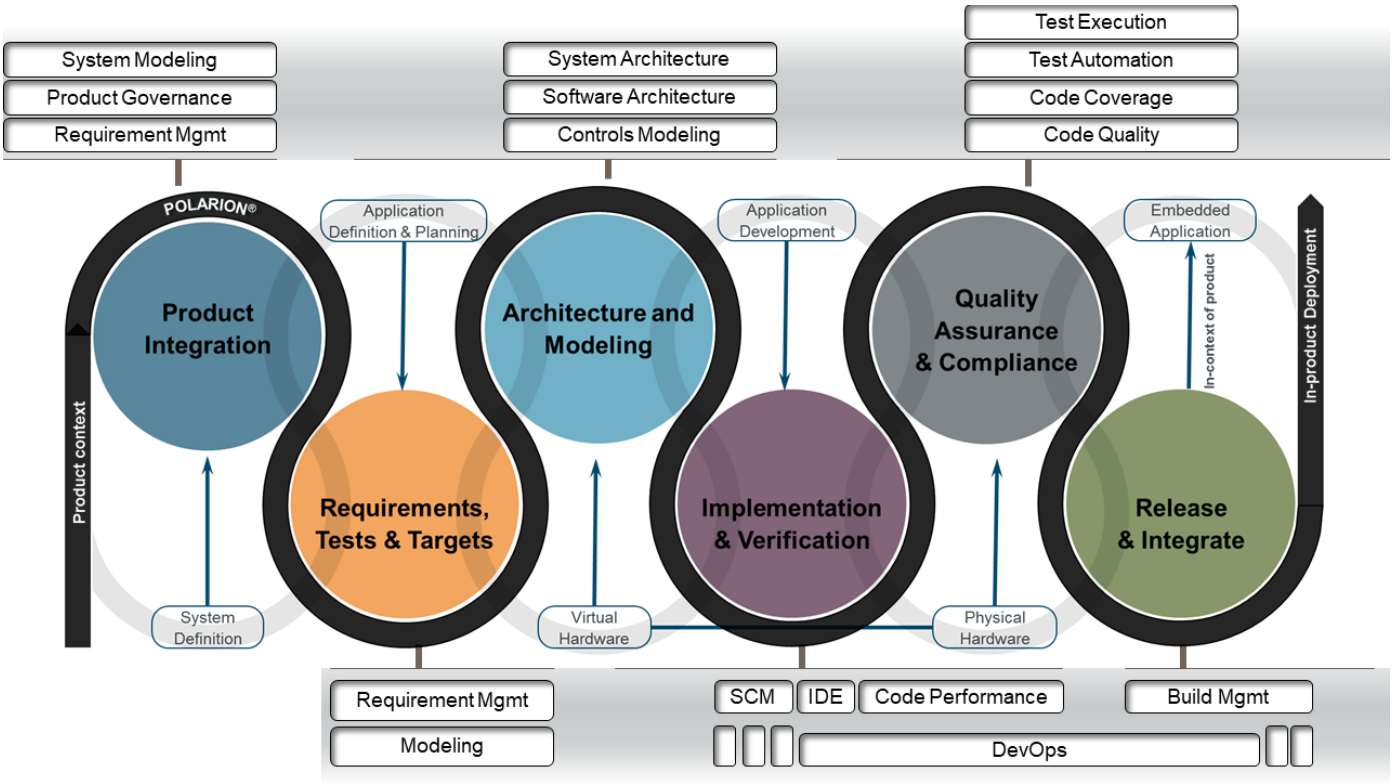

図3: ハードウェアとシステムの制約に基づいて、検証および妥当性確認が完了したアプリケーションのバージョンを完成させるには、車載組み込みソフトウェア開発のための統合型プラットフォームが必要

組み込みソフトウェア開発の3つのプロセス

車載組込みソフトウェアアプリケーション開発を調整する作業は、次の3つの段階に分けられます (図4)。

- アプリケーションの定義と計画
- アプリケーションの開発と品質保証 (QA)
- アプリケーションの提供と監視

ここで重要なことは、さまざまな組織に属する多くの関係者がそれぞれ別のプラットフォームを使用して進める作業の全体的な一貫性と相互適合性を常に保つことですが、統合型ソフトウェア開発プラットフォームであれば、それぞれのプロセスとそこで進める作業を単一のデジタル・スレッドにまとめることができます。その後、組織ごとにアジャイル手法などの手法を選んで、決められた予算と期限を守って堅牢な組込みソフトウェアアプリケーションを開発します。

組み込みアプリケーション
定義と計画

組み込みアプリケーション
開発と品質保証 (QA)

組み込みアプリケーション
提供と監視

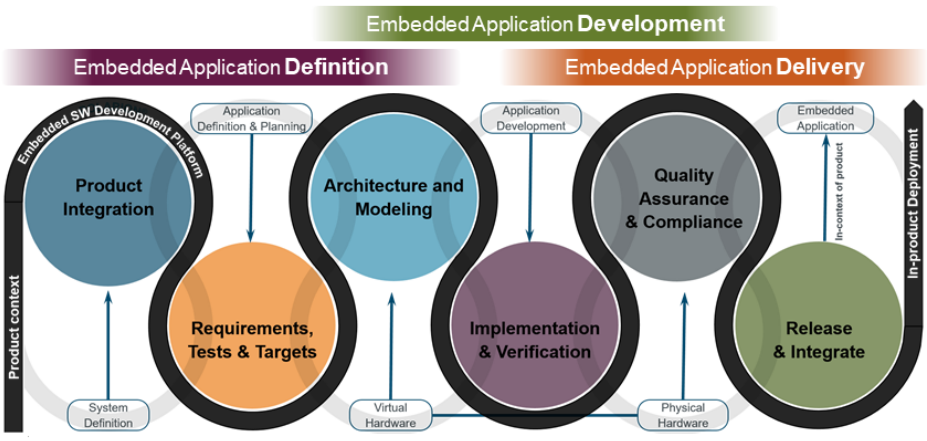

図4: 機能中心の車載組込みアプリケーション開発の3段階

アプリケーションの定義と計画

高度なソフトウェア開発調整ツールであるPolarionは、システムレベルの製品定義を使用、追跡しながら、システムレベルの変更管理とアプリケーション開発を直接連携させ、プロジェクトと全体的なシステムを同期させます。ここで言うシステムは、自動車レベルの機能抽象度で生成されたものです。

システムレベルの定義とハードウェア制約をシステムレベルの要件および仕様として定義したら、ハードウェアとソフトウェアの要件、ノイズ要因、故障モード影響解析 (FMEA)、テストケース、設計目標の組み合わせをソフトウェアエンジニアが分解して、組み込みソフトウェアアプリケーションレベルで定義と計画をスタートさせます。計画段階では、必要な複数のツールセットを使用して、モデリング、コード生成、テスト、組み立て製造などのタスクを割り振ることができます。ソフトウェアエンジニアは、システムレベルの定義と制約条件を参照しながらほかのシステムユーザーと協業でき、システムレベルの変更が発生したときに通知を受けてソフトウェアレベルの影響を評価することが可能です。

後続の変更を追加したり、詳細なソフトウェア要件を作成したりすると、それを契機として、ソフトウェアアーキテクチャーとモデルが変更されます。その後、ソフトウェア要件に基づき、ソフトウェアエンジニアが制御アルゴリズムに適合するようにモデルを更新します。ソフトウェアエンジニアは、コードレベルの変更に進む前に、詳細なモデルインザループ(MiL) 試験を実施して、アプリケーションレベルとシステムレベルで期待どおりの結果が得られることを検証します。こうすることで、プロセス早期からシステムレベルの方向性と足並みをそろえ、エンジニアが重篤なリスク、問題、欠陥を発見できるようになります。

アプリケーション開発と品質保証 (QA)

ソフトウェアコンポーネントのアーキテクチャー設計とモデリングのプロセスを通じて、コンポーネントの相互作用が期待どおりの機能性を達成していることを検証する必要があります。検証と妥当性評価によってモデルの堅牢性と完全性を高めつつ、ソフトウェアインザループ (SiL) 試験でコードの改変をテストし、ハードウェアインザループ (HiL) 試験に進みます。自動車モデルレベルで更新を加えたら、再度、試験を実施し、自動車性能レベルでの一貫性、相互適合性、全体的な責任追跡性の保証に必要なレポートとオーディットトレイルを生成します。

このようなモデルベースのアプローチは、プロセスを加速させるだけでなく、ソフトウェアが意図したとおりに動作し、設計上のハザードがないことを保証するSOTIF (意図した機能の安全性) 手法の確実な実装にも役立ちます。SOTIF手法は、標準的な機能安全性と、故障が起こるという前提で安全目標を確立するSTPA (システム理論に基づくプロセス解析) アプローチを補完するものです。このように手法を組み合わせることで、かつてない強固な自動車組み込みソフトウェア・アプリケーションの開発を支援しています。単一のソフトウェア環境内で標準的なプロセス記述と必要な作業成果物を紐づける機能もあるため、プロセスの適合性、責任追跡性、一貫性を確保し、データのライフサイクルを連携・管理する作業はすべてツールに任せ、エンジニアは設計タスクに専念できます。

アプリケーションの提供と監視

組み込みアプリケーションは最終的に、ソフトウェアアセンブリに組み込まれた後で、最終的な自動車のBoM (部品表) に追加されなければなりません。その前にまずは、提供用のアプリケーションを準備し、提供したアプリケーションを監視する基盤を構築する必要があります。基盤には、自動車アーキテクチャーの枠組みのどこでどのようにアプリケーションが使用されるのかを持続的に監視する機能も欠かせません。

ここで紹介する統合型ソフトウェア開発プラットフォームは、アプリケーション開発、配布、監視のすべての段階で複数のツールを連携させることで、コード性能データとテストカバレッジデータを提供するとともに、複数の手法とコーディング基準 (MISRA-Cなど) を整合させます。それに加えて、コードの適切な更新と必要なQAプロセスを実施する連携的なコラボレーション環境もサポートします。

最終的には、必要な機能が達成されていること、システムが安全に動作することを保証するため、仮想 / 物理ハードウェアと周辺機器を使用してアプリケーションを検証および妥当性評価しなければなりません。検証によって、組み込みソフトウェアアプリケーションの完全性、適合性、車両システムレベルの要件と仕様に対する遵守性を保証できます。

ソフトウェアエンジニアは、車両バリエーションに適したソフトウェアビルドを提供する必要があります。車両プラットフォームは、標準装備、オプション装備、コンポーネント、組み込みハードウェアを組み合わせた多様な車両バリエーションに展開されます。ソフトウェア設計領域は、対象のバリエーションに応じてビルドが変わるため、非常に複雑です。ソフトウェアエンジニアは、車両バリエーションごとに異なるハードウェアそれぞれに対し、完全に適合するソフトウェアを確実に提供しなければなりません。通常は、ソフトウェアの構成ツール、校正ツール、ブートローダーをECUごとにパッケージ化する方法を取っています。このパッケージは、出荷後の自動車を顧客がアップデートする際に非常に重要です (図5)。

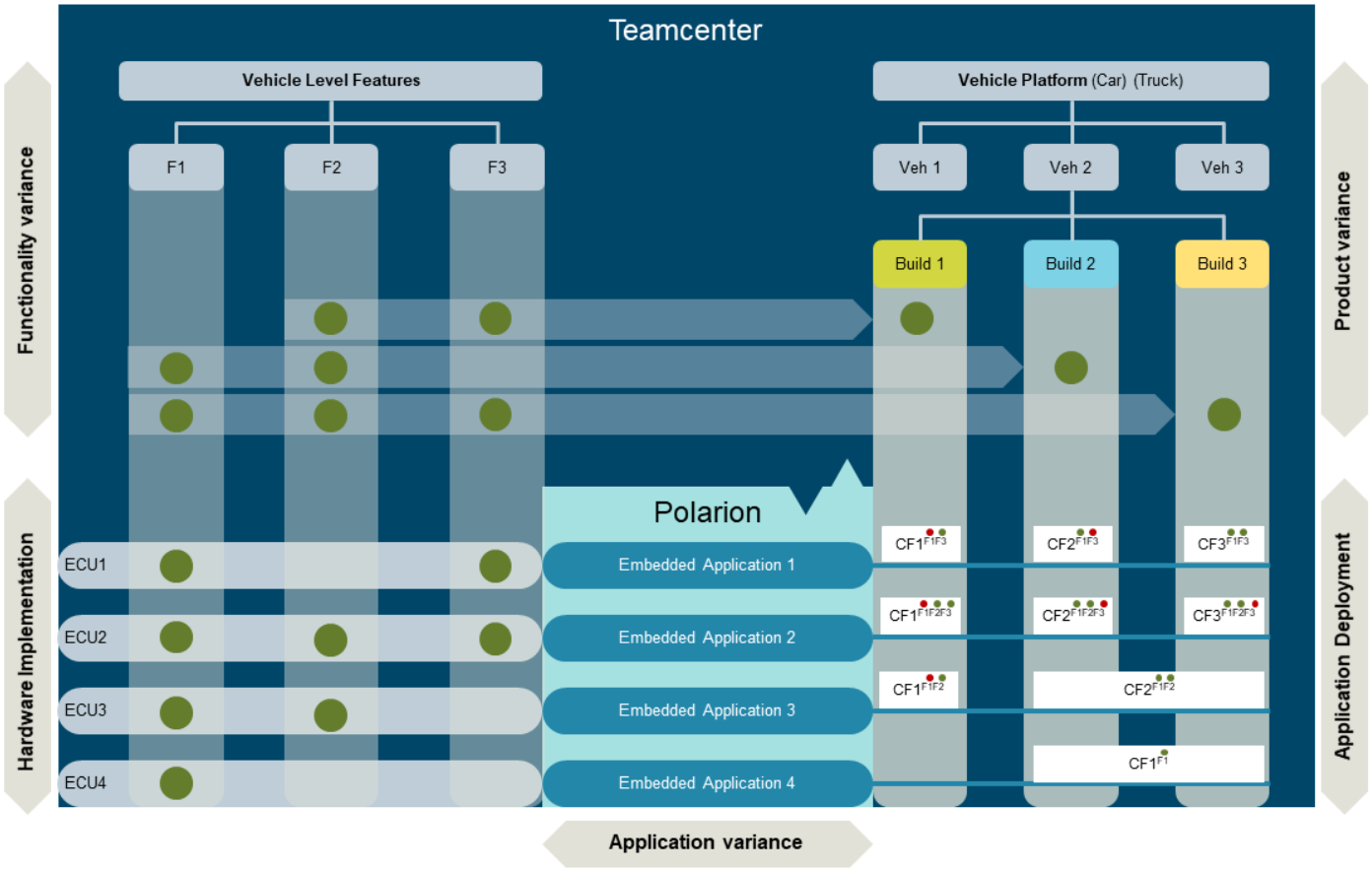

図5: 自動車の機能装備、車両ビルド、アプリケーションビルドの管理と構成

車載組込みソフトウェアにとって重要なもう1つの要素は、アプリケーション構成管理です。企業が組み込みオペレーティング・システム (OS) を開発するようになって以来、ソフトウェアアプリケーション設計がそれを動かすハードウェアに依存することが少なくなりました。スマートフォンのアプリケーションやOSと同様です。アプリケーションの適合性は主として、ハードウェアではなく、OSに左右されます。こうしたことから、車載組み込みソフトウェアアプリケーションと組み込みシステムレベルのソフトウェア (リアルタイムOS、ベースソフトウェアなど)、基本ハードウェアのバリエーション、車両バリエーションとの間の相互適合性が複雑化し、管理するのが困難になりました (図6)。

この複雑性を管理するには、アプリケーションライフサイクル管理 (ALM) と製品ライフサイクル管理 (PLM) を組み合わせた強力なソリューションが不可欠です。ハードウェアとソフトウェアを統合的に開発できるプラットフォームを使用すると、OEMは設計時 (開発向け)、出図時 (エンジニアリング向け)、製造時 (車両組み立て工場向け)、保守整備時 (OTAまたは販売店での更新向け) のそれぞれのソフトウェアビルドに応じて、基本プロセスと基盤を構築およびサポートできます。

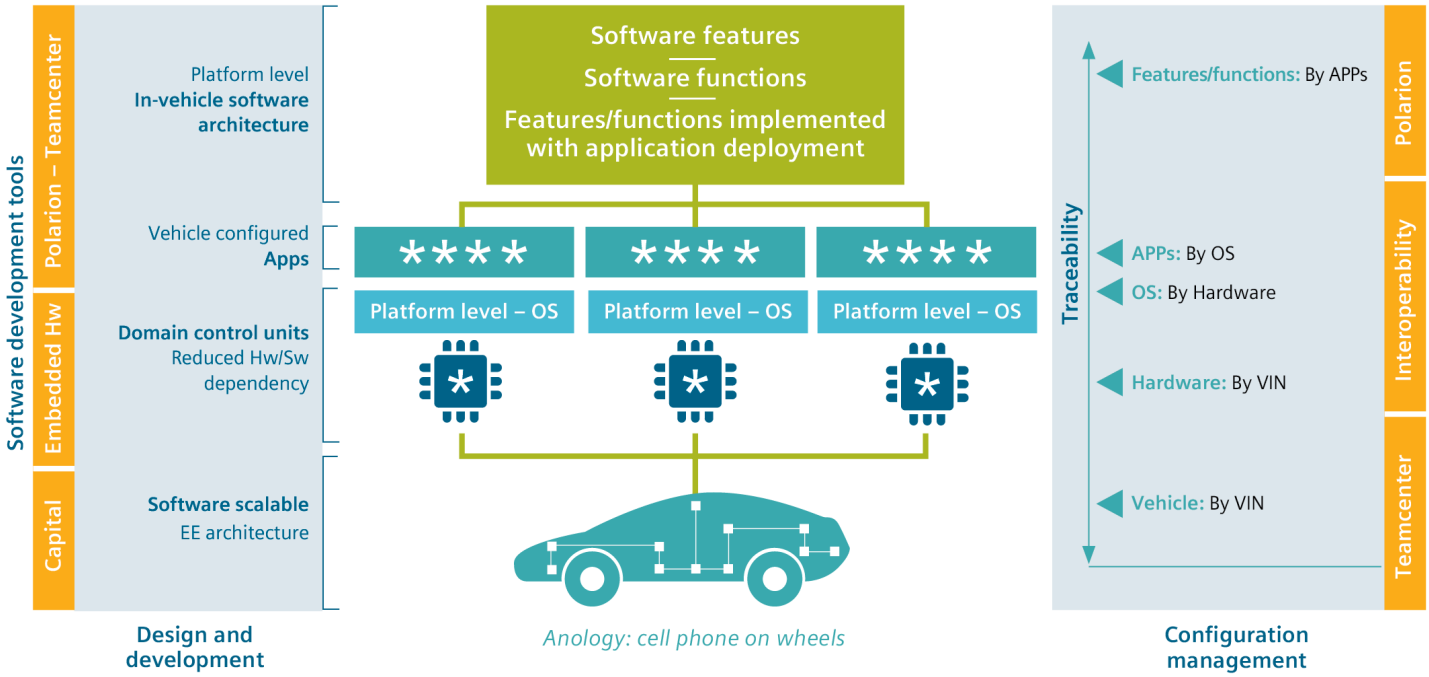

図6: スマートフォンのアプリケーション・エコシステム同様、車載ソフトウェアのハードウェア依存度が下がるにつれて、車載組み込みアプリケーションの構成管理がますます複雑化

アプリケーション開発フローの例

ここでは、ECUに実装するアプリケーションを新規開発するフローを例に取り、手順を説明します。開発するアプリケーションはアダプティブ・クルーズ・コントロール (ACCM)、開発プロットフォームとしてPolarionを使用しました (図7)。システムレベルの抽象度で自動緊急ブレーキ (AEB) とアダプティブ・クルーズ・コントロール (ACCM) の2つの機能を統合させます。まず、新しいシステム要件と仕様をACCMおよび関係するすべてのサブシステムにカスケード展開します。

その後、ACCMに対する大幅なソフトウェア変更のほか、周辺機器にも影響があるため、ACCM外部の非ソフトウェアにも変更を加える必要が生じます。ACCMへの変更内容には、新しいシステム要件、ハードウェア仕様、FMEA、安全要件が含まれることもあります。ACCMソフトウェアエンジニアは、ACCMソフトウェアを改変する過程でシステム全体を考慮に入れ、持続的に連携しなければなりません。ソフトウェア開発調整ツールと製品ライフサイクル管理 (PLM) ソフトウェアを接続することで、複数のチームや抽象度を持続的に繋ぎます。特に、機能レベルの変更管理とPLM管理の連携が際立っています。

システムレベルの変更に対するスプリント作業を通じて、ソフトウエア固有の要件、テストケース、テスト手法を必要に応じて変更および更新します。ソフトウェア管理ツール内のデータ生成物に基づき、変更の影響を受けそうな作業項目にはフラグが付けられるので、変更を確定する前に詳細にレビューできます。AEBとACCの両方を更新するための設計変更要求 (ECR) の大半にはあらかじめ、フラグがついています。PLMとソフトウェア開発調整ソリューションは、こうして変更中のデーター貫性を維持します。

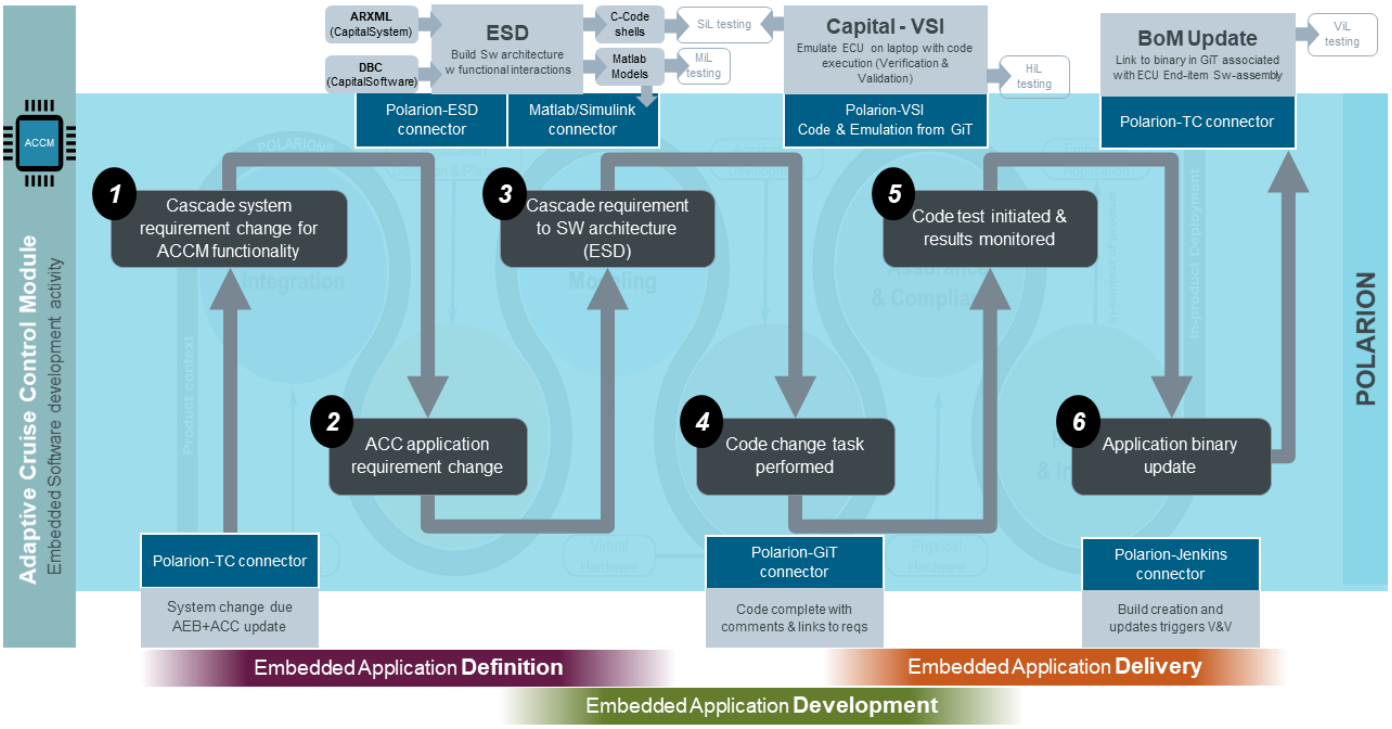

図7: 統合型ソフトウェア開発プラットフォームの機能を使ったアプリケーション開発フローの例

アジャイルプロセスでもシステムレベルの変更に対処できます。ソフトウェア要件に対する更新は、Simcenter Embedded Software DesignerまたはMATLAB/Simulinkのソフトウェアコンポーネントアーキテクチャーとモデルに対する更新を駆動します。システムレベルの変更を下流に展開するときに、影響を受けそうなモデルにフラグが付けられるなど、統合型ソフトウェア開発プラットフォームで調整が行われます。アーキテクチャーレベルの変更とモデルレベルの変更が確定する (電装システム設計ツールから電子システムとCANバスネットワーク通信の変更をインポートする) と、モデルインザループ (MiL) 試験のためのモデル (MatlabまたはC言語) がESDからエクスポートされます。変更のレベルと規模によっては、更新後のアーキテクチャーに基づいて機能インターフェースを検証するため、変更をコードに適用する前に、予備的なSiL試験を実施することもできます。

機能インターフェースの検証を終えたら、スプリントを開始してコード変更を実装します。コード変更は、元々の要件 (手順2) とアーキテクチャー変更に紐づけが可能なため、変更の正確性とシステムレベル要件に対する完全な追跡性が確保されます。マーケティング上の理由による変更案件や保証対象の修理案件を例に取って考えてみましょう。ボトムアップの追跡機能は、複数の車両開発プログラムに共通する変更を評価し、(何度も更新する必要なく) 1回の更新で機能を改良できるため、大きなコスト削減につながります。

アジャイル手法のスプリント作業では、更新後のコードは仮想ハードウェア上のSiL試験または物理ハードウェアを使ったHiL (ハードウェアインザループ)試験のいずれかで検証します。ハードウェア制約やシステムレベル制約とソフトウェアビルドを継続的に統合しながらアジャイル手法で試験します。このソフトウェア開発プラットフォームには、コード変更がハードウェアとソフトウェアの相互作用に与える影響をシステムレベルでテストする機能があります。また、全体的な試験結果とプロジェクトのダッシュボードを使用して、検証済みの堅牢なソフトウェアアプリケーションの展開準備が整っているかどうかをエンジニアが確認できます。

「出図時」BoMに含める最終ソフトウェアアセンブリは、PLMツールからの設計変更通知 (ECN) に基づいて更新されます。ECNには、構成済みの自動車構造におけるECU部品番号が記載されるので、最新のソフトウェアアセンブリを最終的な車両アセンブリに確実に提供できます。その後、最新のソフトウェアアセンブリはECUフラッシングを行う販売店の組み立て工場に送られます。

車載組み込みソフトウェア開発の統合

アーキテクチャー定義ツール、モデリングツール、試験ツールを統合したソフトウェア / 製品開発エコシステムは、車載組込みソフトウェア設計のための統合プラットフォームです。このプラットフォームは、追跡機能を持ち、IP再利用を促す協業環境を提供するため、非適合性をプロセス早期に検知することで、開発時間とコストを削減します。この結果、過度の繰り返し作業を行うことなく、安全性、堅牢性、セキュリティ性に優れた製品を構築できます。

統合型の車載組込みソフトウェアプラットフォームは、エンジニアリングビューから直接、設計データ、要件、試験結果をオンデマンドかつリアルタイムで確認する機能をサポートし、複雑なソフトウェアアプリケーション開発調整の基盤です。また、強固なデジタル・スレッドの構築にも役立ち、点在する組織とチーム間を有機的かつ直感的に連携できるため、すべての利害関係者の生産性を著しく高めます。

それだけでなく、Polarionのようなソリューションであれば、エンタープライズアジャイルやSAFeをはじめとする大規模実装プロジェクトのテンプレートのほか、ISO 26262、A-SPICE、CMMIなどの標準規格に準拠したテンプレートが付属しており、アプリケーション展開が迅速化します。テンプレートを顧客に応じてカスタマイズできるため、顧客にも採用しやすく、高い費用対効果を達成します (図8)。

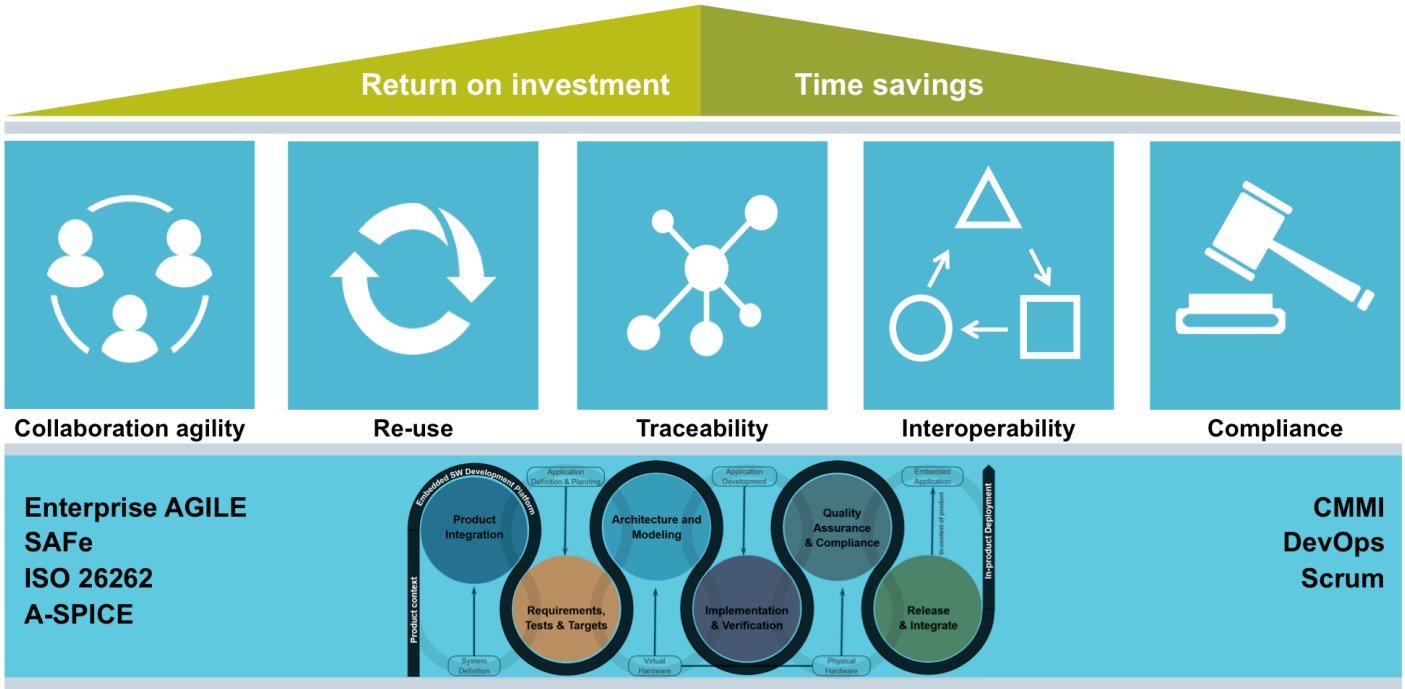

図8: 追跡性、相互運用性、コンプライアンス遵守性とコラボレーションが可能な統合型組み込みソフトウェア開発プラットフォーム。標準テンプレートを使用すると、高い費用対効果を獲得し、開発期間を短縮

エンジニアリングコスト削減と作業の効率化を目指して複雑なエンジニアリング環境を導入した企業は、OEMであろうとサプライヤーであろうと、できるだけ早期に投資を回収したいと思うものです。ソフトウェアの開発と調整を統合したプラットフォームは、カスタマイズ可能な協業環境を提供しており、容易かつコスト効果高くアジャイルワークフローを導入できます。再利用しやすい環境であるため、実証済みコンポーネントをできる限り再利用することで、多くのバリエーションをより迅速に作り上げ、差別化した製品開発と高い収益性を達成します。それだけではありません。リソース、時間、コストを抑えつつ、監査、調査、レビューに必要なトレーサビリティーを提供します。作業生成物の追跡はツールに任せ、エンジニアはエンジニアリングタスクに専念できます。

競争の激しい世界中の市場を対象とする自動車の競争力を高めるうえで、車載ソフトウェアがますます重要なものになってきました。ADASや高性能インフォテイメントシステムなど、購入の決め手になり得る刺激的で便利な機能の実現には多くの先端ソフトウェアが使用されています。一方、市場の競争に打ち勝つには、これまで以上に厳しいスケジュールを達成しなければなりません。また、ソフトウェアの品質と信頼性に寄せられる高い期待にも応える必要があります。ソフトウェアが自動車と乗り手を繋ぐ中心的な役割を果たすようになると、システム遅延、グリッチ、ユーザー・インターフェースの使いにくさは、これまで以上に厳しく評価されます。ソフトウェア開発のための統合型コラボレーション環境のなかでも、追跡とIP再利用の機能が組み込まれているものには、かけがえのない有用性があります。

シーメンスデジタルインダストリーズソフトウェア

本社
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

アメリカ
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

ヨーロッパ
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

アジア / 太平洋
Unit 901-902, 9/F
Tower B, Manulife Financial Centre
223-231 Wai Yip Street, Kwun Tong
Kowloon, Hong Kong
+852 2230 3333

シーメンスデジタルインダストリーズソフトウェアについて

Siemens Digital Industries Softwareは、Siemens Digital Industriesのビジネスユニットです。製造業がイノベーションを実現するための新たな機会を創出し、産業のデジタル・トランスフォーメーションを牽引するソフトウェアソリューションを提供して世界をリードするグローバルプロバイダーです。米国テキサス州プラノを本拠地とし、これまで世界140,000社以上のお客さまにサービスを提供しています。シーメンスデジタルインダストリーズソフトウェアは、あらゆる規模のお客様と協働して、アイデアの実現方法、製品の実現方法、稼働中の製品および設備資産の活用や状況把握の方法を変革できるよう支援しています。製品やサービスに関する詳細はsiemens.com/plmをご覧ください。

siemens.com/plm

78394-82939-C6-JA 11/20 LOC