Our heart
beats
embedded.

車載用マイクロコントローラの
ソフトウェア開発における5つの課題

車載用マイクロコントローラの
ソフトウェア開発に携わるお客様へのご提案

Empowering Tomorrow's Automotive Software
www.etas.com

ETAS

概要

マイクロコントローラは自動車のリアルタイム制御と機能安全を担い、自動車開発において重要な役割を果たしています。しかし、マイクロコントローラ用ソフトウェアの開発プロセスは、一般的には最優先で最適化されているわけではありません。市場では、バックグラウンドで稼働する組み込み機能よりも、差別化を図りやすく、目新しい技術を取り入れたアプリケーションに、より多くの注目が集まります。ソフトウェアデファインドビークルを目指す取り組みが加速すると、基礎的な開発業務が見落とされがちになります。ただし、自動車メーカー様やサプライヤー様にとっては、マイクロコントローラ開発プロセスの最適化こそが、将来の成功を約束する要因なのです。

当社ETASでは、開発プロセスの最適化を推進する際にお客様が直面するであろう課題を「困難なインテグレーション」、「複雑な適合」、「長期間に及ぶテスト」、「スケーラビリティと柔軟性の限界」、「サイバーセキュリティの包括的な要件」の5つに分類し、分析を行いました。このホワイトペーパーでは具体的なプロセスをV字モデルに当てはめ、5つの課題について詳しく解説し、そのソリューションを提案しています。提案したソリューションが、お客様の開発プロセスの効率向上、作業負担の軽減、より高度なセキュリティ基準の実装、そして最終的には、日々進化する市場への迅速な対応につながることを願っております。

目次

1. はじめに

安全性、リアルタイム性、リソースの最適化といった特性はエレクトロニックコントロールユニット（ECU）、特にマイクロコントローラに大きく依存しており、これが自動車のさまざまな機能の根幹をなしています。しかし自動車業界は未曽有の変革期を迎え、自動車メーカー様やサプライヤー様へのイノベーション圧力が高まる中、より新しい機能が優先され、ECUのソフトウェア開発プロセスが軽視される傾向が見られます。その一方で、ECUにはさまざまな機能拡張の可能性があります。その一例としては、さらなる自動化、さらなる効率化、そしてセキュリティの限界克服などが挙げられます。機能要求の高まりとイノベーションの速度向上が必須と考えられる中、こういった根幹部分のプロセスの改善を怠ると、最終的には開発のボトルネックとなりかねません。

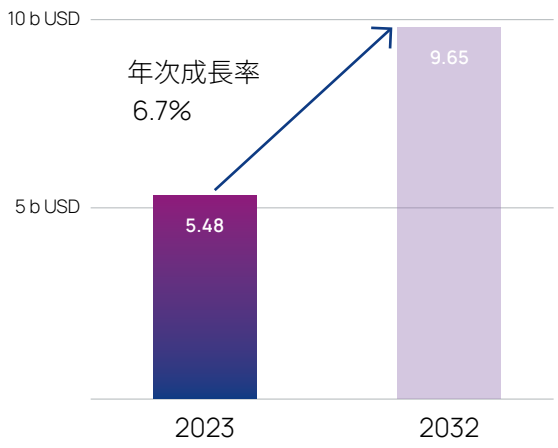
自動車開発では、この根幹部分に多くの時間を割く必要があります。自動車のコードはその大半がマイクロコントローラを対象としており、ユーザーに認知されないルーチンの機能を実行しています。目立たない機能ではありますが、組み込みソフトウェアは自動車において極めて重要な役割を担っています。自動車コンポーネントに不具合が生じれば、深刻な事故につながるおそれもあります。自動車安全基準は常に進化していますが、この安全基準に関する要求はますます厳しいものとなっています。ルーチンの機能とはいえ、組み込みマイクロコントローラシステムに向けたソフトウェア開発は極めて重要であり、その傾向は今後も続くでしょう。

このホワイトペーパーは、自動車メーカー様やサプライヤー様による、最高水準のマイクロコントローラ用ソフトウェア開発に向けた自社プロセスの再編成を支援するものとなっています。ここではまず、車載用マイクロコントローラの重要性に関する基本情報と、ECUの世代交代に向けた新規ソフトウェア開発に必要となる手順について説明します。続いて開発プロセスにおける5つの主要な課題と、想定されるソリューションについて深く掘り下げます。自動車メーカー様が、必ずしも等しくすべての課題に直面しているわけではありません。中には、すべての手順を社内では実施していないメーカー様やサプライヤー様もいらっしゃいます。デジタル化の成熟度、既存ツールの構成、レガシーソフトウェアの割合、ハードウェアの編成なども、各社によってさまざまです。その場合でも、大局的な見通し、共通の課題、および想定される対応策を把握することは、自動車業界の各企業様にとって必ず有益なものとなるでしょう。

2. 車載用マイクロコントローラの世界について

自動車における多くの制御および通信機能は、ECUの中核をなすさまざまなマイクロコントローラによって実施されており、これにはセンサーインターフェースと制御に向けた費用効果の高い8ビットのものもあれば、現代のインフォテインメントディスプレイや自動車の動力系に向けた32ビットのものもあります。標準的な自動車に搭載されるこういった電子コンポーネントの数は40を越えており、ハイエンドの車種では最大150にも及びます。これによって、例えばエアバッグ、ABS、ESC、エンジン制御、タイヤの空気圧、バッテリーの状態などといった機能の信頼性が確保されています。こういった機能はすべてバックグラウンドで実施されており、ドライバーには認識されません。
現在では、E/Eアーキテクチャのイノベーションサイクルの短縮化と、自動車の車種や機能のさらなる増加に、自動車メーカー様やサプライヤー様が対応を迫られるようになりました。したがって、マイクロコントローラをベースとしたソフトウェア開発もまた、より一層の効率化が求められます。ハードウェアとソフトウェアのデカップリング、車載コンピュータの一元化、車載クラウドコンピューティングなどといった、ソフトウェアデファインドビークルに関連するトレンドは、従来のECUアプローチに取って代わるものとみなされている傾向にあります。実際には、リアルタイム性とASIL-Dまでの機能安全を確保する唯一の手段である組み込みシステムは、従来と変わらず重要な役割を果たしていくことになります。そのため、自動車メーカー様やサプライヤー様は、ECU用ソフトウェアに向けた自社の開発プロセスを、速度、柔軟性、セキュリティに関する新たな市場要件に適応させなければなりません。

図1：マイクロコントローラ世界市場の成長率



2023年に54億8千万米ドルと推定される車載マイクロコントローラのグローバル市場の規模は、毎年6.7%の成長率が見込まれ、2032年には96億5千万米ドルに達するとみられています。

Source: https://www.fortunebusinessinsights.com/de/markt-f-r-mikrocontroller-f-r-die-automobilindustrie-104084

組み込みマイクロコントローラシステムのソフトウェア開発は新しいものではなく、また、自動車業界に限ったものでもありません。こういったコンポーネントは、歯ブラシから掃除機に至るまで、あらゆる電子機器や日用品に導入されています。では自動車の場合、その開発プロセスはなぜ特別なものになるのでしょうか。最も重要な点は、自動車が道路に出る、まさにその瞬間から高い安全性と信頼性が要求されることであり、その次に、数十年間にわたる保守性が重視されることです。さらに、パワートレイン制御系や自動車の動力系のような多くのシステムでは、信頼性の高いリアルタイム性が極めて重要となります。そのため、電子制御システムは、自動車や環境内で展開されている物理的なプロセスに対し、遅れをとることは絶対に許されません。

世代交代ごとのマイクロコントローラのプログラム設定は、自動車メーカー様やサプライヤー様にとっては繰り返し実施されるルーチンの作業ではありますが、法令・法規による高度な要求にプロセスを適合させつつ、自社のグローバルなネットワーク内でそれを可能な限り効率化する必要があります。特に、現状の要件に対して部分的にのみ適合しているツールや方法を用いた、伝統的なプロセスチェーンを利用している企業にとっては特に大きな課題となります。

解説：

ECU、MCU、MPU、VCUの連携

車載システムで使用されるエレクトロニックコントロールユニット（ECU）は特化型のコンピュータであり、自動車の固有の機能を管理します。このECU中のマイクロコントローラ（MCU）は、エンジン制御やセンサー管理のようなリアルタイム性の機能を取り扱います。より高度なECUにみられるマイクロプロセッサ（MPU）は、インフォテインメントやADASなどの複雑なアプリケーションに対してより優れた処理能力を発揮し、多くの場合、Linuxのようなオペレーティングシステムで稼働します。車載コンピュータはこれらのECUを連携させ、車載ネットワーク全体におけるシームレスな通信と運用を可能にします。

3. 車載ECUのソフトウェア開発プロセス

E/Eアーキテクチャは、より一元化されたアプローチに向けて再編成されつつありますが、マイクロコントローラをベースにしたECUは残り続けるでしょう。組み込みソフトウェアの一般的な開発プロセスは、理論上はそのまま継続させることができます。ただし、ソフトウェアデファインドビークル（SDV）の台頭や、機能性を求める市場の要求を満たすために、開発サイクルは急激に加速していくでしょう。従来のアプローチでは、この速度に追いつくことはできません。しかし、従来のアプローチは一夜のうちに置き換えられるものでもありません。通常、最適なソリューションを導くには、革新的なソリューションを用いたツールと試行錯誤を繰り返すプロセスが必要です。

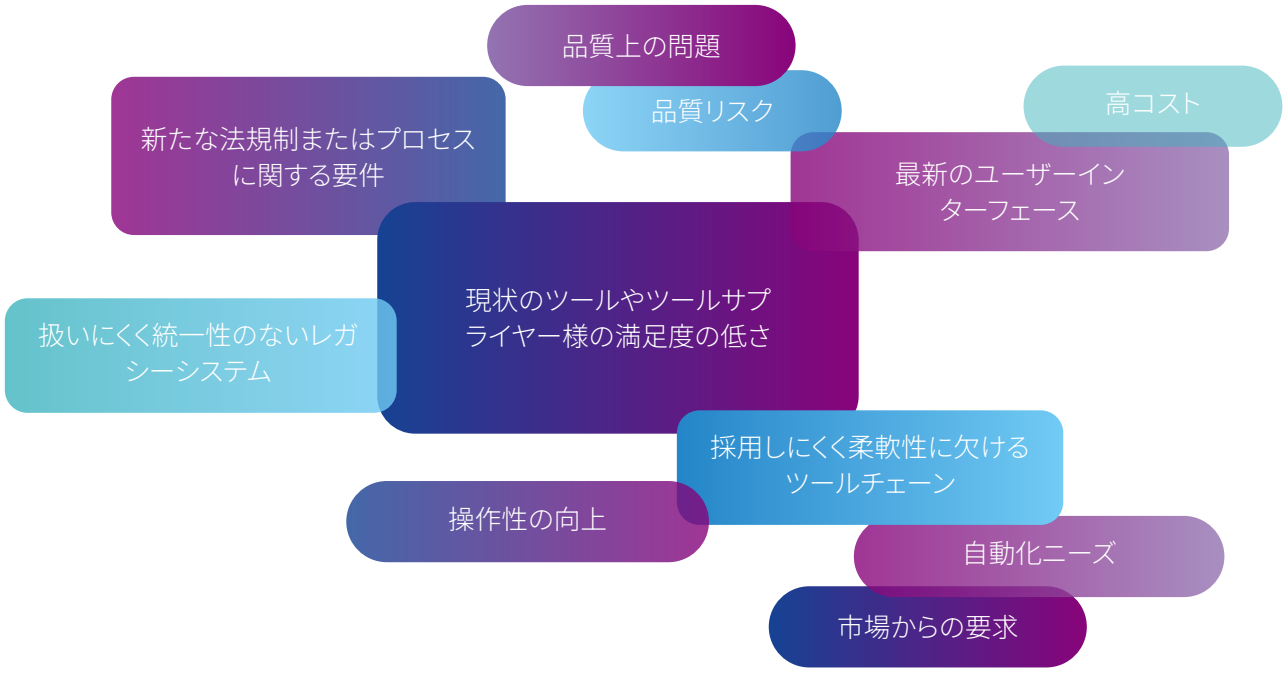
まずは、プロセスそのものの出発点を把握し、そのプロセス内のさまざまな手順を分離することが重要です。

3.1 ECU世代交代の理由

具体的な課題を深く考察する前に、そもそもなぜ新たなECU用ソフトウェアを開発しなければならないのか、そしてそのプロセスはなぜ長期間に及び、複雑になるのかを考えてみましょう。きっかけとなるのは、ある世代から次の世代へのECUの切り替わりであり、これにはソフトウェアの実質的な再構築が伴います。新たなソフトウェアのプログラミングを始めるには、事前準備としてツールチェーンの編成に多くの労力を費やす必要があります。ECUには新たな世代ごとに独自の開発プロセスがあり、専門領域と連携モデルによって、このプロセスも異なります。対応するツールチェーンは来るべき世代に向けて個別に定義する必要があり、多数の変更点が盛り込まれることになります。

ある世代から次の世代への移行時にこういった変更が必要となるのには、多くの理由があります。（図2）。その一部は、品質上の問題や、現状のツールやツールサプライヤー様の満足度の低さなどといった内的要因です。外的要因には、新たな法規制またはプロセスに関する要件（例えば、ASPICEやISO 26262 ASIL-Dなど）が挙げられます。また、市場からの要求、ユーザビリティの向上や最先端のユーザーインターフェースへの期待といったものも、要因のひとつとなります。世代交代では、高コスト、品質リスク、および整合性のないレガシー（適応が難しく、柔軟性のない）ツールチェーンの編成といった要因が主な障害となります。それに対し、世代交代で期待できることは、主にプロセス内のさらなる自動化です。

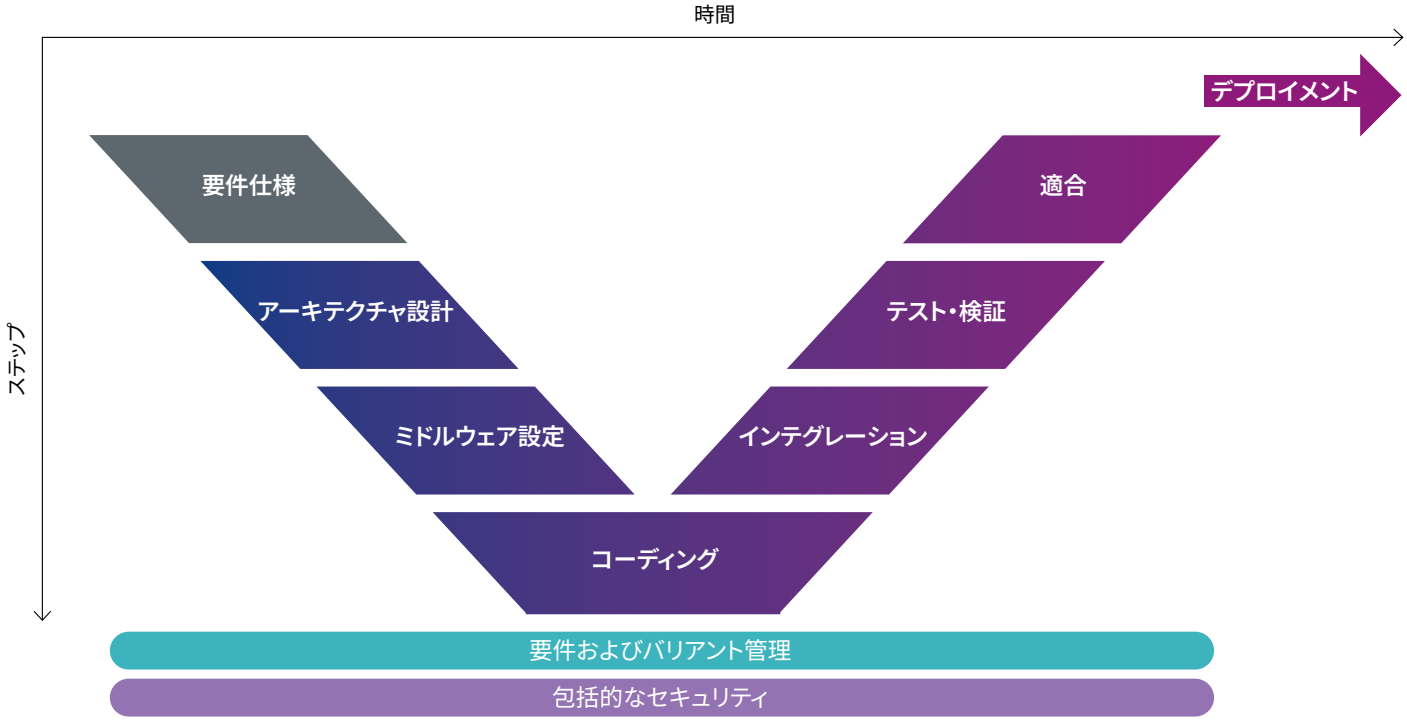
Figure 2: Reasons and obstacles for ECU generation changes



3.2 開発プロセスの主な手順

組み込みソフトウェアの開発プロセスはV字モデルで説明するのが一般的です（V字モデルにはさまざまなバリエーションが存在します）。これまでの実績から、当社では各フェーズを次の主要な作業手順に分類しました。これらは必ずしも交互に実施されるわけではありません。並行して進められる手順もあれば、何度か実施しなければならないものもあります。要件および変更の管理はあらゆる作業手順で必要となり、数回にわたって繰り返し実施されます。

図3：車載ソフトウェア開発のV字モデル



このV字モデルは、ECUソフトウェア開発プロセス向けのものであり、ソフトウェア開発は2つの大きな部分に分けられる。

アーキテクチャ設計

ECUの一部として、マイクロコントローラはエンジンや自動車の動力系の制御などといった特定の連結機能を担当します。したがって、極めて詳細な要件カタログが欠かせません。まず初めに、すべての機能、制約、ハードウェア要素、ミドルウェア、およびこれらの相互依存性を特定します。ハードウェアおよびソフトウェアコンポーネントは、この前半フェーズで必ず厳格な自動車規格に適合していなければならず、細部にまで及ぶ入念さが求められます。

ミドルウェア設定

ミドルウェアとは、オペレーティングシステムやハードウェアと、ECUのアプリケーションソフトウェアとの間のソフトウェアレイヤーを表します。これは橋渡しの機能を果たし、自動車のソフトウェアアーキテクチャ内における種々のコンポーネント間での通信およびデータ交換を促進させます。アプリケーションとハードウェア関連システムとをデカップリングすることで、ミドルウェアはそれらの開発、保守、およびアップグレードの個別実施を可能とします。ミドルウェアが仲介役として機能することで、土台となるハードウェアが抽象化され、ソフトウェアコンポーネント用に標準化されたインターフェースによる、シームレスな相互作用が可能となります。

ECUの脆弱性と適合に関する問題は、不正アクセス、データ侵害、およびシステム不具合に至るおそれがあり、自動車の安全性と信頼性が損なわれる危険があるため、言うまでもなく、こういった仲介役は最高レベルのセキュリティ基準に適合していなければなりません。厳格な規格にこのレベルで適合させるには、広範なテストと検証が必要となり、開発プロセスに複雑さと時間が加わることになります。そのため、成熟度（ISO 26262 ASIL-D適合）、将来の方向性（ベンダーロックインの防止）、およびサイバーセキュリティを担うミドルウェアの選択は極めて重要となります。継続的な更新を伴う、堅牢かつ高度な設定が可能なミドルウェアのソリューションでなければ、進化を続ける脅威からECUを守ることはできず、また、あらゆる（変化を続ける）規則の要件を満たすことはできません。

コーディング

アーキテクチャが定義され、すべてのコンポーネントがその固有の要件とともに特定されると、デベロッパーはシステム設計を機能ソフトウェアに落とし込みます。要求される機能はすでに標準プロセスに含まれています。この段階で、ソフトウェアデベロッパーは新たな（補足的な）機能を書き込み、エラーを除去し、そして既存の機能を最適化または拡張します。ここでの課題は、プロセスの設定において既存の機能が書き換えの必要なく実装されるようにすることです。これには高度な機能安全、サイバーセキュリティ、およびコーディング効率を維持するのみならず、高度な再利用性が必要となります。

インテグレーション

この時点ですべてのワークストリームが集められ、ミドルウェア設定、アーキテクチャ設計からの情報、アプリケーションソフトウェア、および事前適合データがコードとして構築され、マイクロコントローラに送信することができます。

大きな課題は最適化にあります。つまり、自動車機能のパフォーマンスと新規ハードウェアの能力とでバランスをとり、独自の効率目標と環境条件を満たすことです。ここでは、効率的なシステムリソースの使用量とリアルタイム運用とに向けて、ミドルウェアを微調整します。もちろん、安全性と適合性も極めて重要であり、厳格な安全基準と規則の各要件に適合するには、やはり厳格なテストと認証プロセスを実施する必要があります。

テスト・検証

テスト・検証フェーズでは、ECUの機能が、安全性、パフォーマンス、および信頼性に関するあらゆる要件と仕様に適合していることを確認します。広範なテストを通じて、発生のおそれがある問題や不具合を特定します。自動車に多くのソフトウェアが搭載されている場合、このプロセスは非常に複雑かつ冗長になるおそれがあります。そのため、コストを最小化し、パフォーマンスを最大化するためには、短い期間で可能な限り多くのテストを実施することが極めて重要となります。

適合

適合では、各パラメータと各フィールドにデータを入力し、ソフトウェアの挙動を物理的なシステムに適応させます。アプリケーションの中には数千ものパラメータを備えるものもあり、それらが相互に影響し合います。また、パラメータは独自のパフォーマンス目標、環境条件、および基準に適合しなければなりません。また、条件が更新されれば、パラメータ等を再設定する必要があります（排気量基準値が変更された場合など）。

デプロイメント

ソフトウェアが「使用可能」段階に至ると、最終承認プロセスに移行し、対象となるすべてのECUに送信されます。

4. 自動車メーカー様とサプライヤー様における主な課題

前章で説明したプロセスの全体像を把握すると、お客様の抱える課題がどのプロセスで発生しているかを特定することができます。これには、1つの手順のみに関連するものもあれば、プロセスそのもの、あるいはある世代から別の世代への移行に当てはまるものもあります。当社は多くのお客様との関わりを通じて得た知識と経験をもとに、5つの主要な課題を特定しました。

4.1 困難なインテグレーション

車載システムが開発されるのは必ずと言って良いほど「ブラウンフィールド状態」、つまり既存のソフトウェア開発環境上です。そのため、インテグレーション時に多くのレガシーの問題が発生することで、開発プロセスが複雑化してしまいます。種々のコンポーネント間の多様かつ複雑な通信インターフェースによるシームレスな相互運用性を確保するには、込み入った設定と広範なテストが求められます。さらに、インテグレーション作業を手動プロセスに頼ると、エラー率が高まり、開発サイクルが長引き、人件費が高騰します。

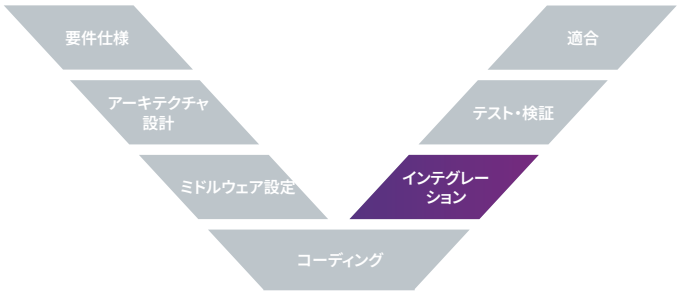


図4：V字モデルにおける「インテグレーション」ステップ

課題解決のアプローチ

段階的なモダナイゼーション：
レガシーコンポーネントを最新の相当品で徐々に更新し、後方互換性を確保することで、確立されたプロセスの全変更や高額な投資コストのリスクを負うことなく、新規システムへの円滑な移行およびインテグレーションを促進することができます。

インテグレーション作業の自動化：
自動化されたツールおよびスクリプトを実装し、頻発するインテグレーション作業に対処することで、エラー率が低減され、開発サイクルが加速します。

ミドルウェアソリューションの導入：
最高水準のミドルウェアを使用することで、インターフェースの複雑さを抽象化および管理し、一貫した通信レイヤを構築することで、インテグレーションの作業を簡略化できます。また、標準化された通信プロトコルを採用および徹底することで、インターフェース設定を簡略化し、種々のコンポーネント間における相互運用性を確保できます。

4.2 複雑な適合

適合は長期間に及ぶため、開発サイクルの間延び、プロジェクトタイムラインの遅れ、コストの増加、および効率の限界が発生します。適合パラメータの把握の難しさから設定時にエラーが発生するおそれもあるため、テストの繰り返しと、再適合のサイクルが余儀なくされます。パラメータが誤って設定されると、そのマイクロコントローラが基準に不適合となり、さらなる改訂や、コストのかかるリコールに至るおそれがあります。さらに、プロジェクトが複雑な場合は、いまだに手動で実施されている可能性のある従来のソフトウェアドキュメンテーションが限界に達し、エラーが生じる可能性も高まります。

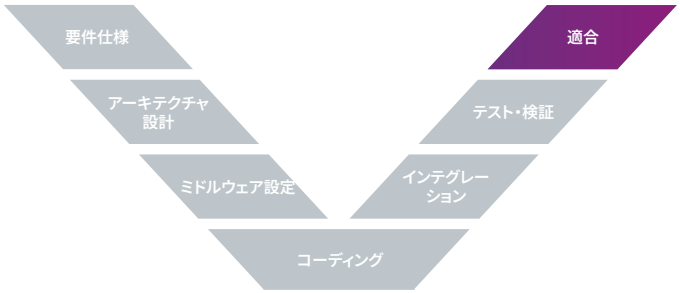


図5:V字モデルにおける「適合」ステップ

課題解決のアプローチ

ドキュメンテーションの改善と知識の共有:
包括的な文書を作成し、そこに各適合パラメータ、その用途、他のパラメータとの相互依存性、およびシステムへの影響を明確に記載することが大切です。ドキュメンテーションの自動化ソリューションを実践することで、エラーを削減し、プロセスの効率化が図れます。

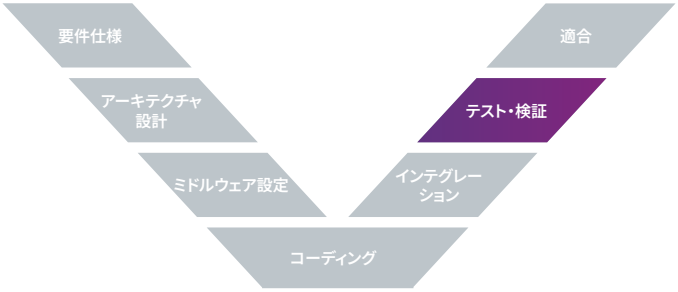
ユーザーフレンドリーなインターフェース:
各ユーザーにとって必要なパラメータとオプションのみが表示されるようなソリューションを利用することで、適合プロセスを簡略化することができます。スライダーやダイナミックチャートなどのインタラクティブな表示を使うと、適合設定値の把握および調節が容易になります。

自動化と標準化:
ルーチンの適合作業に対処するには自動化ツールの使用が有効です。これによって手動作業の削減、適合プロセスの合理化が実現し、さらに、種々のプロジェクト間とチーム間における一貫性と信頼性が確保されます。

シミュレーションとモデリング:
シミュレーションおよびモデリングツールを使用して、実際のハードウェアでは起こりえない結果を排除しつつ適合の変更点をテストできる、仮想テスト環境の構築を推奨します。各適合設定値の影響を可視化し、より良好な意思決定を実施できます。さらに、適合パラメータがシステムモデルに組み込まれた、モデルベースの設計アプローチを適用することで、操作とテストが格段に容易になります。

4.3 長期間に及ぶテストとデバッグ

ソフトウェアが複雑であるほどテストの必要性は高まりますが、テストの長期化がイノベーションにおけるボトルネックとなります。そのため、長期的視野のテストプロセスでは、信頼性と安全性を伴った効率化が重要です。さらに、種々のモジュールとシステムとの間の相互作用からテストシナリオが複雑化するおそれがあり、それを管理および実行するのが困難となります。また、テストを始めるどころか、それ以前に、インテグレーションプロセスが長期化する可能性もあります。その上、組み込みマイクロコントローラは処理能力、メモリ、およびストレージに限界がある場合が多く、それによってハードウェア上で直接的に実施できるテストの種類と範囲が限られるおそれがあります。したがって、上記のような組み込みシステムの制約においては、従来のデバッグテストツールが不適切となる可能性があります。



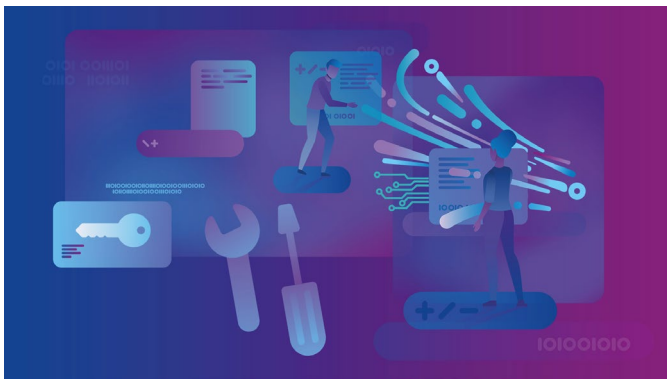


図6：V字モデルにおける「テスト・検証」ステップ

課題解決のアプローチ

モデルベース開発：
シミュレーションツールを使用して、仮想環境でシステムのモデル化とテストを実施することを推奨します。これにより、プログラム設定のバグが直ちに検出され、プログラマーにフィードバックされるため、ハードウェアベースのテスト時間が削減されます。

高度なデバッグツール：
リアルタイムデバッグ用のインサーキットエミュレータ（ICE）、およびトレースプロファイルツールは、実行フロー、パフォーマンスのボトルネック、およびメモリ使用量を分析し、効率的なデバッグと最適化を支援します。

SiLテスト：
仮想環境（フロントローディング）でテストを実施すれば、早い段階でのエラー検出が可能になります。また、並列化とタイムラプステストによっても、仮想環境でのテストが加速されます。

継続的インテグレーション／継続的デプロイメント：
CI/CDパイプラインを構築することで、ビルド、テスト、およびデプロイメントプロセスを自動化して、手動作業を削減し、ループの繰り返しを加速させることができます。自動化されたテストでは問題が早期に検出され、開発におけるバグフィックスの時間とコストが削減されます。

ユニットテストおよびテスト駆動開発：
TDDプラクティスの採用により、コーディング前にテストが実施され、テストカバレッジが高まります。インテグレーション前にコンポーネントのユニットテストを設計および実行すれば、問題の切り分けと解決が簡略化されます。

コード生成：
モデルからコードを自動生成することで、一貫性を確保し、ヒューマンエラーの可能性を減らすことができます。

自動化フレームワーク：
複雑なテストシナリオをスクリプトで自動化すれば、一貫性を確保し、手動作業を削減できます。また、コードカバレッジツールを使用して、すべての実行パスとエッジケースを網羅することが可能です。

並列および継続的テスト：
変更がある場合は、必ずそのシステムの継続的テストを実施することを推奨します。複数のテストリグや仮想環境を利用することでテストを並行して実施することで、テスト時間を削減できます。また、効率化とボトルネックの削減のためにハードウェアのリソース割り当てを最適化することも重要です。

4.4 スケーラビリティと柔軟性の限界

堅牢かつモノリシックなソフトウェア設計では修正と拡張が制限されており、更新と新機能の追加が複雑化します。自社システムの場合や標準化が不十分な場合は、ベンダーロックインや複雑化に陥り、新機能やサードパーティー製コンポーネントとのインテグレーションでエラーが発生しがちです。また、ソフトウェア設計においてモジュール化が不十分な場合は、効率的なコードの再利用が阻害され、新機能のシームレスなインテグレーションが妨げられて、システム全体の適応性に影響が及びます。

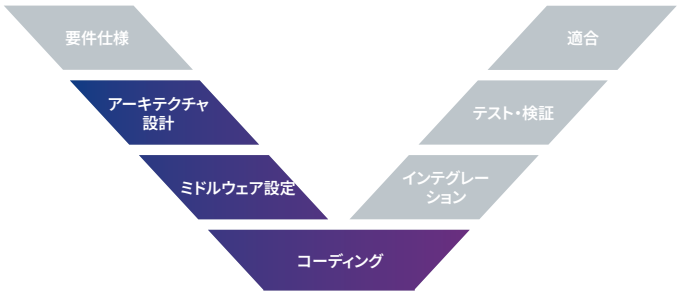
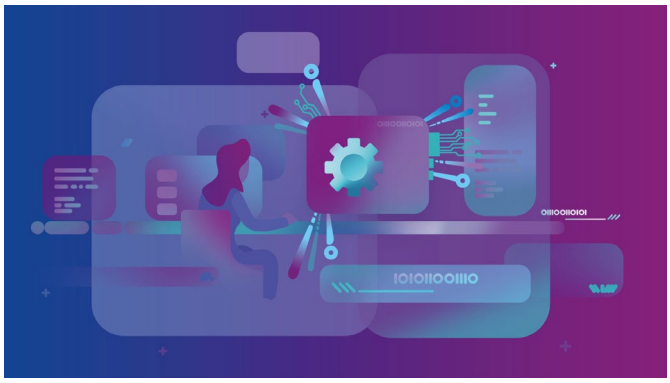


図7:V字モデルにおける「アーキテクチャ設計」「ミドルウェア設定」「コーディング」ステップ



課題解決のアプローチ

モジュラー設計とインターフェースの標準化:
モジュール式かつコンポーネントベースのアーキテクチャと標準化された通信プロトコルを実装して、モノリシックな構造を分解し、更新と拡張を容易にすることが重要です。

オープンスタンダードと相互運用性:
相互運用性を考慮しつつ、オープンスタンダードと設計システムをツールチェーン内に実装することで、ベンダーロックインを防止し、新機能またはサードパーティー製コンポーネントのインテグレーションが簡略化されます。一般的な基準に適合しておらず、それによりベンダーロックインを引き起こすおそれがあるサプライヤーの採用は避けることをお勧めします。

ミドルウェアと抽象化レイヤー:
ミドルウェアと抽象化レイヤーを利用すれば、アプリケーションロジック同士およびそれを特定ハードウェアからデカップリングし、更新とスケーラビリティの向上が容易になります。

モジュール化とコードの再利用:
高度なモジュール化と再利用可能なコンポーネントを利用してソフトウェアを設計することで、新機能のシームレスなインテグレーションを促進して、システム全体の適応性が高まります。

4.5 サイバーセキュリティの包括的な要件

マルウェア、ハッキング、不正アクセスなどのサイバー脅威が進化・多様化するにつれ、ECUシステムのセキュリティを確保する作業が複雑になってきています。残念なことに、マイクロコントローラの計算リソースには限界があり、パフォーマンスへの影響なく包括的なセキュリティ対策を実践することは困難です。さらに、さまざまな分野と業界における、厳格かつ常に変化するサイバーセキュリティの規則と基準を遵守すると、開発プロセスがさらに複雑化します。

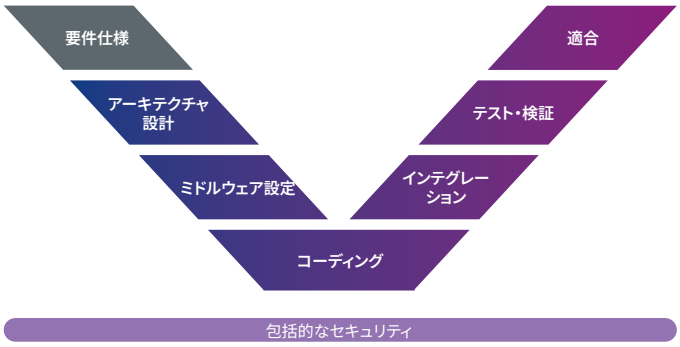
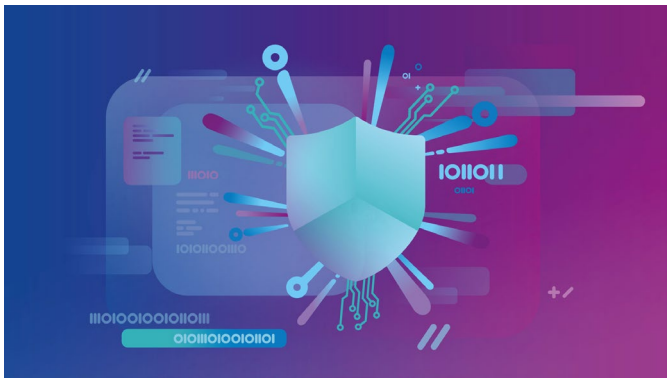


図8：V字モデル全体にかかわる包括的なセキュリティ要件



課題解決のアプローチ

多層セキュリティアーキテクチャ:
多様な脅威ランドスケープには、多層セキュリティのアプローチで包括的に対処することが有効です。これには暗号化、認証、侵入検知、およびセキュアブートのメカニズムが含まれます。

リソース制限環境のセキュリティ:
計算リソースの限られたマイクロコンピュータに特化した、軽負荷かつ効率的なセキュリティプロトコルを利用すれば、パフォーマンスを損なわずに堅牢な保護を確保できます。

制への適応:
変化を続けるサイバーセキュリティの規制を継続的に監視し、これらに適応することが大切です。また、開発ライフサイクルに適合要件を盛り込み、定期監査を実施して規制への準拠を確実にする必要があります。

5. まとめ

E/Eアーキテクチャ内の最小コンポーネントであるにも関わらず、機能別のECUからドメイン別のECUに至るまで、あらゆるアーキテクチャにおいてマイクロコントローラは極めて重要な役割を担っています。マイクロコントローラのソフトウェア開発の向上に投資することで、今後数十年においても、自動車市場における競争力が維持されます。目指すべきは速度の向上、コストの削減、およびセキュリティの保証ですが、レガシーや、自動車のさらなる複雑化にも考慮する必要があります。この実現には確立されたプロセスの再検討が必ず伴い、それにはこのホワイトペーパーで提示した課題の段階的な解決が有用です。経験豊富なパートナーによる洗練されたソリューションを上手に取り入れることで、自動車メーカー様やサプライヤー様は自社の組み込みソフトウェア開発を最適化することができ、将来的な（市場の）要求と（法的）規制に対する柔軟性が備わります。強固な土台を完成させることにより、グローバルな自動車産業における長期的な競争力を形成することができます。

ETASについて

1994年に設立されたETAS GmbHは、ボッシュ・グループの完全子会社であり、欧州、北米、南米、アジアに海外営業拠点を展開しています。

ETASは、ソフトウェア開発、車載ミドルウェア、データ収集および処理、サイバーセキュリティの分野で、ソフトウェアデファインドビークルを実現するための包括的なソリューションを提供しています。
車載サイバーセキュリティの分野では、業界のパイオニアとして各種オンボードおよびオフボードソフトウェア製品を取り揃えているほか、お客様がサイバーセキュリティ関連の複雑さをコントロールし、サイバーリスクを軽減し、ビジネスの可能性を最大限に引き出すためのプロフェッショナルサービスもご提供しています。

ETASの自動車セキュリティソリューションは、すでに世界中の何百万台もの車両システムに導入され、ソフトウェアデファインドビークルのサイバーセキュリティ標準を確立しています。

ETASのソリューションについて

30年以上の実績を誇る大手自動車サプライヤーとして、ETASは幅広いソリューションポートフォリオを擁しています。お客様のニーズと市場の要件に合わせて組み合わせやカスタマイズが可能です。以下にECUソフトウェア開発プロセスに特化した組み合わせ例をご紹介します。

ETAS AUTOSARソリューション: 簡単、安全、かつセキュアなソフトウェア統合

当社は、自動車メーカー様およびサプライヤー様向けに、ソフトウェアとハードウェアのシームレスな統合と、サイバー脅威からの包括的な保護のためのソリューションバンドルを提供しています。さらに、当社の仮想テストソリューションを使えば、開発ライフサイクルの早い段階で統合を行うことで、テスト時間を短縮し、開発作業の速度を上げることができます。

ETAS RTA-CAR (Classic AUTOSAR) は、
Classic AUTOSAR
ソフトウェアコンポーネントの開発と構成に使用されるツールセットであり、ECUの効率的な統合、コード生成、検証を可能にします。
+
ESCRYPT CycurHSMは、
ECUのセキュアブート、車載通信、ECUコンポーネントの保護、フラッシュを保証する革新的で柔軟なHSMセキュリティ ファームウェアです。
+
ESCRYPT CycurLIB は、
組み込みシステムの要件に特化した暗号化ライブラリです。
+
仮想 ECUは、
開発の初期段階で複雑なテストを実行します。

ETAS 仮想化環境ソリューション: クラウドベースのソフトウェアイン ザループテスト

ETASは、クラウドでのSILテスト用のモジュール式ツールとサービスを提供しています。仮想ECUの作成、デバッグ、事前適合、およびクラウドで並列実行される仮想アーティファクトの統合とシミュレーションが可能になります。大規模なテストケースであってもテストと反復処理を実行できるため、効率とシミュレーション速度が大幅に向上します。クラウドベースのソリューションは拡張可能なため、ソフトウェアの継続的な統合とデプロイをサポートします。これは、現代の開発プロセスにとって特に重要です。

ETAS VECU-BUILDER は、
ソフトウェアインザループ (SIL) セットアップで自動車用マイクロコントローラソフトウェアの検証と妥当性確認を行うための仮想ECUを生成するツールです
+
COSYM (システムのCOシミュレーション) は、
開発の初期段階でシステム レベルでソフトウェアをテストおよび妥当性確認するための強力なシミュレーションおよび統合プラットフォームです。
+
ETAS MODEL-SYMULATOR を使用すると、
クラウドでシミュレーションを実行できます。
+
ESCRYPT CycurFUZZ は、
最新の規制と標準を満たすのに役立つ最先端のファズテストツールです。

ETAS の適合およびドキュメンテーションソリューション: データを価値ある情報に変える

ETASのシームレスなツールカップリングソリューションは、データの検索、分析、および転送を自動化します。複数の製品を1つのバンドルに統合することで、ECUの適合と診断における効率、精度、およびチームコラボレーションを強化します。

ETAS EHANDBOOK は、
ECUソフトウェア機能のロジックをさまざまな抽象レベルでインタラクティブかつグラフィカルに表示するドキュメンテーションツールです。
+
INCA (Integrated Calibration and Application Tool) には、
自動車用電子システムの適合、診断、および検証用の柔軟なツールが含まれています

リアルタイム適合検証ソリューション: データを深く読み取る

ETASのソリューションバンドルは、適合のリアルタイム検証と迅速なフィードバックを容易にし、迅速な調整と高品質の結果を可能にします。この統合アプローチにより、遅延の最小化、チームのコラボレーションの強化、開発サイクルの加速、ECUプロジェクトの成功が促進され、プロジェクトの効率が保証されます。

ETAS EHANDBOOK
を使用すると、詳細なECU ドキュメントにすぐにアクセスできます。
+
ETAS MDA (測定データ アナライザ) は、
データの分析と視覚化を簡素化し、傾向と問題を明らかにして正確なキャリブレーションと最適化されたECUパフォーマンスを実現します。
+
ETAS EATB (ETAS Analytics Toolbox) は、
テストを自動化および合理化し、包括的な洞察と情報に基づいた意思決定のための強力な分析およびレポート ツールを提供します。
+
ETAS ASCMO (キャリブレーション、モデリング、最適化のための高度なシミュレーション)
を使用すると、ECU の動作の詳細なシミュレーションと高度なキャリブレーションが可能になり、パフォーマンスと効率が向上します。

連絡先

Anthony Esteban
Customer Chief Engineer, ETAS
Get in touch on LinkedIn

Contact Form