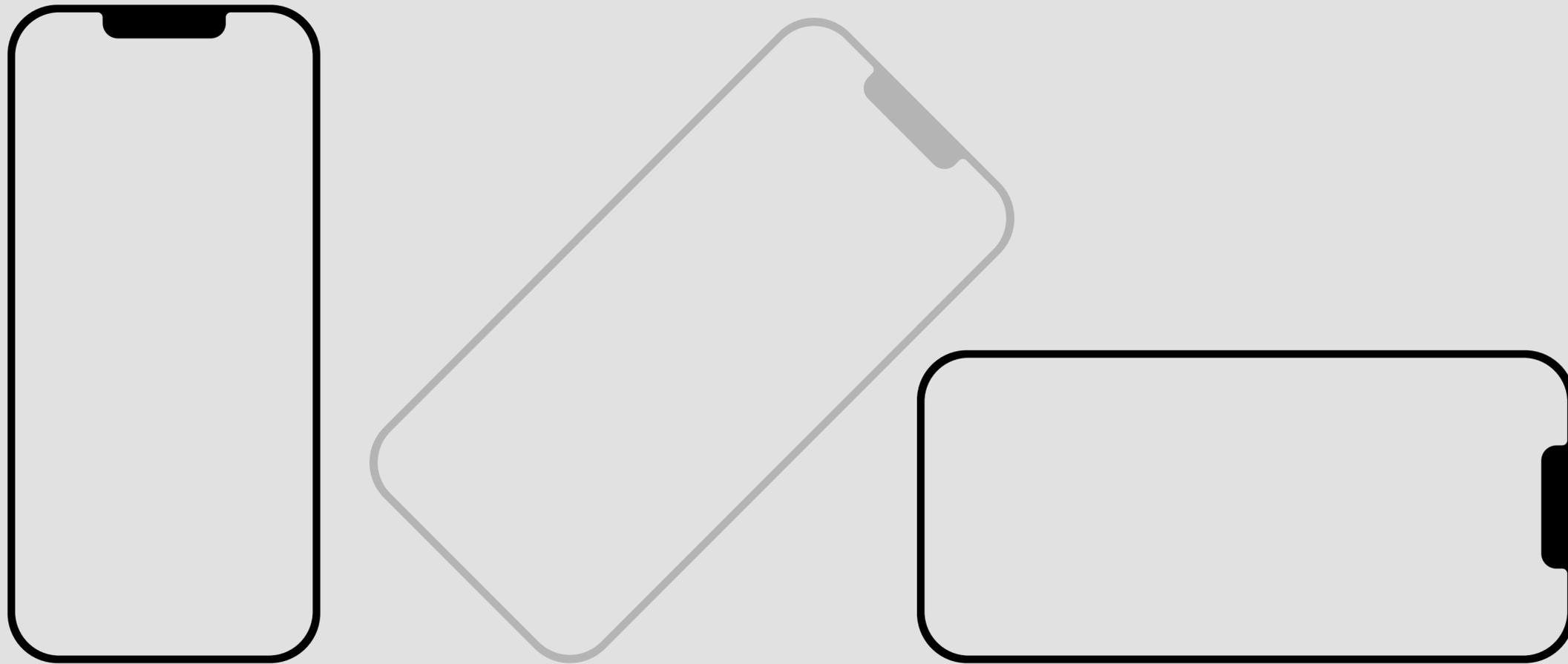


# ソフトウェアデファインドビークルと ソフトウェア時代のリーダー

ソフトウェア複雑性を克服しソフトウェア主導の変革を牽引する  
大手OEM向け戦略プレイブック



モバイル端末で読んでいますか？  
**横向きにする**と読みやすくなります。



## はじめに

本プレイブックは、最大の課題であるソフトウェアの複雑性を、最強の競争優位へと転換する方法を示します。

成熟した OEM のリーダーにとって、いま核心となる課題は、新規参入企業と競い続けることではなく、製品の本質となった膨大なソフトウェア複雑性をいかに制御するかにあります。

変革は「実施すべきかどうか」ではなく、「どのように実現するか」の段階に入りました。本プレイブックは、この現実の中で勝ち抜くために必要な体制を構築するためのロードマップを提示します。

### このプレイブックについて

製造業を変革してきた自社の卓越したエンジニアリング力を、ソフトウェア開発という新たな分野にも生かすことで、成熟した自動車メーカーをソフトウェア主導企業へと転換していくための戦略フレームワークです。それは、これまでの強みを手放すのではなく、その強みを新たな形で発揮することで実現されます。

### 主要項目

#### 主要課題の提示

「1億行の技術的負債」が収益性、イノベーション速度、そして競争ポジションに影響を与える主要な変動要因となっている理由と、それを管理するために求められる戦略的アプローチ。

#### プラットフォームエンジニアリングの体系

分散していたソフトウェア開発を、統合された能力へと再構築するための考え方。標準化が開発スピードを高める要因と、ゴールデンパスと統合ワークフローによる品質の標準化。

#### ソフトウェア基盤の構築

アーキテクチャ検証からコンプライアンスの自動化まで、安全な SDV 開発に求められる技術スタック。

#### 未来のビジネス機会に向けたトランスフォーメーション・ロードマップ

プラットフォームを自社で保有することで、AI 安全性コンプライアンスなど、変化し続ける市場要求に対応できるようになる理由。組織変革を段階的に進めるための4ステップの指南と、すでに移行を進めた OEM が成果を上げている実証済みの戦略。

### 対象読者

小さな改善ではソフトウェア格差を埋められないことを認識しているシニア技術専門家やエンジニアリングリーダー。本書は、組織がソフトウェアを「どう捉え、どう構築し、どう提供するか」を体系的に変革する意思を持つ意思決定者に向けた内容です。

既存の強みを生かしながら戦略的なトランスフォーメーションを進めるリーダーになりましょう。自動車産業はすでに大きな変革を経験しています。本プレイブックは、どのようにして「変革を起こす側」へと転じられるのかを示します。

## 目次

### 主要な用語

#### はじめに

OEM各社がソフトウェア会社となる時代

### 1. レガシー型アプローチはSDV時代でも依然として通用せず

産業から学ぶ

### 2. ハードウェア企業のソフトウェア原則

プラットフォームエンジニアリングだけが成果を出せる理由

データが証明する成果

プラットフォームエンジニアリングが実現するもの

### 3. 品質・安全性・セキュリティが築く基盤

安全な開発環境を構築するために必要なスタック

05

06

07

09

13

14

16

16

17

18

20

品質向上が生む継続的な価値

20

基盤から革新へ——現場で実証された成功事例

21

### 4. プラットフォームとコードベースの掌握がイノベーションを生む

22

アーキテクチャ管理でAIとADASを極める

24

Android Automotive の真価を引き出す

25

データドリブンなユーザー体験の創造

25

顧客成功事例 - Elektrobit

26

### 5. 進むべき道 – 変革を加速する4つの実践ステップ

28

ソフトウェア主導シフトの投資対効果

31

### まとめ

32

次のステップ

34

# 主要な用語

SDV の変革を進めるうえで重要となる主要な概念をあらかじめ整理しておきましょう。

## ソフトウェア定義ドビークル (SDV: Software Defined Vehicle)

車両の中核となる機能、特長、性能が固定的なハードウェアではなく、**ソフトウェアによって決定され、継続的に強化されるよう設計された車両**のこと。

「コンピューターに車輪が付いたもの」と捉えるほうが、「コンピューターが搭載された車」と考えるよりも本質を捉えています。

## プラットフォームオーナーシップ

車両の機能、ユーザー体験、そして将来拡張される能力を規定する**重要なソフトウェアレイヤーを自らの管理下に置くこと**。プラットフォームオーナーシップは「すべてを自社で開発することだけを意味するわけではなく、段階的な広がりの中に位置づけられる概念です。

サプライヤーに依存することなく、自社の判断でソフトウェアスタックを統合し、改修し、発展させられるアーキテクチャ上の主導権を持つことも含まれます。

## プラットフォームエンジニアリング

ソフトウェア開発を戦略的に進めるためのアプローチで、**あらゆるツールチェーンやワークフローを「ゴールデンパス」として体系化**し、開発者が必要な基盤を一から構築することなくセルフサービスで開発できるようにする考え方。社内の開発基盤を「ひとつのプロダクト」として扱う点に特徴があります。

## エンジニアリングプラットフォーム

プラットフォームエンジニアリングによって構築される具体的なソリューションで、**開発を加速するための統合されたツール群、ワークフロー、標準をまとめた仕組み**。アプリケーションが動作するハードウェア／ソフトウェアプラットフォームとは別物なので、混同しないように注しましょう。

## 技術的負債

**将来の変更をますます困難かつ高コストにしてしまう、コード内に蓄積された複雑性や安易な回避策**のこと。自動車分野では、数百万行規模のレガシー C/C++ コードが集積するミドルウェア層にその典型が見られ、コンプライアンスや安全規制の対応を一段と複雑にしています。

## アーキテクチャ検証

**ソフトウェアの実装が意図した設計どおりになっているかを継続的に確認**し、「アーキテクチャの劣化」によって統合トラブルや安全・コンプライアンス上のリスクが発生することを防ぐ取り組み。

はじめに

# OEM各社が ソフトウェア会社となる時代

ソフトウェア企業への変革を成し遂げた自動車メーカーが、今後10年のモビリティイノベーションを方向づける存在になります。一方、その取り組みを先送りする組織は、ためらわずに前進した競合から該当する機能をライセンス調達せざるを得なくなるでしょう。

いま問うべき本質的な問いは次のとおりです——車両の未来をコントロールするのは、エンジニアなのか、それとも膨れあがったコードベースの複雑性なのか。

昨今の車両には1億5,000万行を超えるコードが搭載されており、今後10年でさらに10倍規模に拡大すると予測されています。参考として、NASAのアポロ11号はおよそ14万5,000行のコードで月面着陸を実現しました。現在の車両は、人類を宇宙へ送り出したシステムをはるかに超えるソフトウェア複雑性を内包していることになります。

この現実には、自動車産業の姿を根本的に変えてしまいました。Tesla、BYD、吉利（Geely）などの新興OEMは、車両を迅速で反復可能な「ソフトウェアプラットフォーム」として扱っていますが、大手OEMはどこもレガシーアーキテクチャと複雑なサプライチェーンがもたらす制約に苦しんでいます。

プラットフォームオーナーシップ = 製品の未来を握ること

プラットフォームのオーナーシップはスペクトラムの中に存在するものです。重要なのはすべてを自前で作るのではなく、顧客体験を規定し、イノベーションを可能にする「重要レイヤ」をコントロールすることです。異なるベンダーやサプライヤーのコンポーネントを活用していても、アーキテクチャと統合を自分で掌握していれば、「自分のプラットフォーム」を持っていると言えます。その上に独自のユーザーエクスペリエンスを構築できるからです。すべてがどのようにつながり、進化していくかを自分で握っているとき、競争優位性そのものを所有していると言えるのです。

## 高品質なソフトウェアをスピード感を持って出荷できる力が 新たな競争軸になる

2024年、**Recall Masters** は報告書の中で、2,800万台以上の車両がリコールの影響を受け、そのうち174件のキャンペーン、合計1,380万台の車両が、ソフトウェアおよび電子システムの故障に直接起因していたことを明らかにしました。<sup>1</sup>

リリース後に発見される欠陥は、開発段階で見つかった場合と比べて修正コストが最大100倍に膨れ上がるため、ソフトウェア品質はもはやエンジニアリング上の懸念にとどまらず、リーダーシップレベルの事業リスクとなっています。

この変革が起きているかどうかは問題なのではありません。それを「制する側」になるのか、「制される側」になるのかが問われているのです。本プレイブックは、主導権を取り戻し、隠れた負債を競争優位に転換するための戦略的ロードマップを提供します。

<sup>1</sup>Recall Masters State of Recalls 2024 [➤](#)



# 1

レガシー型アプローチは  
SDV時代でも依然として通用せず

# レガシー型アプローチはSDV時代でも依然として通用せず

何年も前から聞き続けてきたことでしょう。レガシー型アプローチは機能しない、と。業界のリーダーたちは長年、旧来のやり方ではソフトウェアデファインドの時代に対応しきれないことを認識してきました。

しかし、問題を知っているだけでは何も解決しません。多くの大手 OEM にとって、旧来のやり方は今もなお日常の現実であり、「1億行の技術的負債」として知られる、イノベーションを引きずる恒常的な重しになっています。この問題は、新しいプロジェクトを行うたびに作業を複雑にし、コストを押し上げる継続的な課題です。それは、長年蓄積してきた技術的複雑性だけでなく、チームが日々強いられているフラストレーションの象徴でもあります。

この技術負債は、レガシー車両アーキテクチャに根ざした断片化された基盤から生じています。たとえば、複数のサプライヤーから提供される、数十もの独立したサイロ化された ECU に縛られた基盤と向き合わざるを得ません。この断片化によって、SDV の中核能力である「車両全体のソフトウェア更新」がほぼ不可能になっているのです。

こうした状況は、問題を次々と連鎖的に引き起こします。

「レガシーアーキテクチャこそが根本的な問題です。1台の車両には約60～90個ものサイロ化された ECU が搭載されており、それぞれが異なるティア1サプライヤー製のブラックボックスとして動作し、遅く帯域の狭いネットワーク上で互いに通信しようとしています。」

OEM にとって、この断片化されたモデルは、現代の SDV メーカーが得意とする高速な車両全体の FOTA や SOTA 更新をほぼ実現不可能にしてしまいます。

そして悪循環が生まれます。新しい機能を追加する際、従来のやり方ではサイロ化された ECU をさらに追加するのが常套手段となってきました。しかし、それを積み重ねるたびに、コードベース全体は巨大化し、複雑化し、モデルイヤーごとの更新はさらに困難になっていくのです。」



Miao Luo

Qt Group

テクノロジー戦略ディレクター

## コストの増大

現在、車両リコールの大部分はソフトウェア問題が原因となっており、断片化されたアーキテクチャはこうした問題の発生率を高めるだけでなく、修正コストもさらに押し上げています。また、世界の自動車サイバーセキュリティ市場規模は2024年に35億米ドルと評価されており、2025年から2034年にかけて年平均成長率（CAGR）11.6%で拡大すると予測されています。<sup>2</sup>

## 将来の収益を阻む要因

アーキテクチャを十分に制御できていない環境では、信頼性の高い OTA（Over-the-Air）アップデートの展開が極めて複雑になります。**Tesla** は2012年に OTA を先駆的に導入しましたが、それから13年経った今でも、多くの自動車メーカーにとって「依然として未知の領域のままです」と、調査会社 TechInsights の自動車部門バイスプレジデントである Ian Riches は述べています。<sup>3</sup>

<sup>2</sup>GM Insights Automotive Cybersecurity Market [➤](#)

<sup>3</sup>MotorTrend - The Slow Roll of Automotive OTA Updates [➤](#)

”  
現在はソフトウェアの問題が  
自動車リコールの  
相当数を占める  
”

## 開発速度を鈍化させる要因

技術負債は「イノベーションの重石」を生み出し、単純な更新でさえ、文書化されていない依存関係や統合の悪夢に阻まれてしまいます。チームは顧客価値を生み出すよりも、レガシー制約の管理に時間を取られるようになります。その結果、競争上の大きな格差が生まれます。SDV の先進企業が数週間でアップデートを出荷できる一方、レガシー環境のチームは同じ変更にも6~12か月を要するのです。このスピードの遅さは単なる不便さでは済まず、やがてセキュリティリスクへと発展します。重大な脆弱性が明らかになった際、アジャイルな競合は数日で FOTA (Firmware-Over-The-Air) により全車両をパッチ適用できますが、断片化されたアーキテクチャを抱えるメーカーでは、顧客が数か月間も危険にさらされることになるのです。

クリーンで適切に管理されたアーキテクチャがなければ、OEM は SDV 時代の特徴であるサービスベースの収益モデルを活用することができません。



## 産業から学ぶ

新しいプログラムが開始されるたびに、この技術負債は指数関数的に増大していきます。特に、ADAS や AI 機能などの最新システムをレガシーミドルウェアと結びつけようとする際に、統合の課題が一気に増幅します。この現実を受け、多くの先進的な OEM は自社のアプローチを再評価し始めています。

たとえば **Ford** は、自社の高度な電気アーキテクチャの習熟に、ソフトウェア開発の焦点を再び当てるといった戦略的決断を下しました。これは、持続的なイノベーションを実現するには、まず基盤となるプラットフォームを自らの手で確立する必要があるという業界の認識が広がっていることを示すものです。この変化は、既存 OEM が自らの強みと長年のエンジニアリング知見を生かしつつ、ソフトウェア基盤の近代化を戦略的に進めていることの好例と言えます。

[4 CNBC - Ford kills project to develop Tesla-like electronic brain ↗](#)

```
import Quick3DAssets.LaneAssist 1.0
import Quick3DAssets.DangerArrow 1.0
import QtQuick.Timeline
import Data 1.0 as Data

// Turn signal state
property bool signaling: laneAssist.lane < 0
property bool turning: "Turning" === state

property alias lanelines: lanelines
property alias timelineAnimationAdas: timelineAnimationAdas

environment: sceneEnvironment

// Lighting modes
Keys.onDigit1Pressed: dynamicLighting.state = ""
Keys.onDigit2Pressed: dynamicLighting.state = "Day"
Keys.onDigit3Pressed: dynamicLighting.state = "Dusk"

// Weather modes
Keys.onDigit4Pressed: currentWeather = 0
Keys.onDigit5Pressed: currentWeather = 1
Keys.onDigit6Pressed: currentWeather = 2
Keys.onDigit7Pressed: currentWeather = 3
```

# 2

## ハードウェア企業の ソフトウェア原則

# ハードウェア企業の ソフトウェア原則

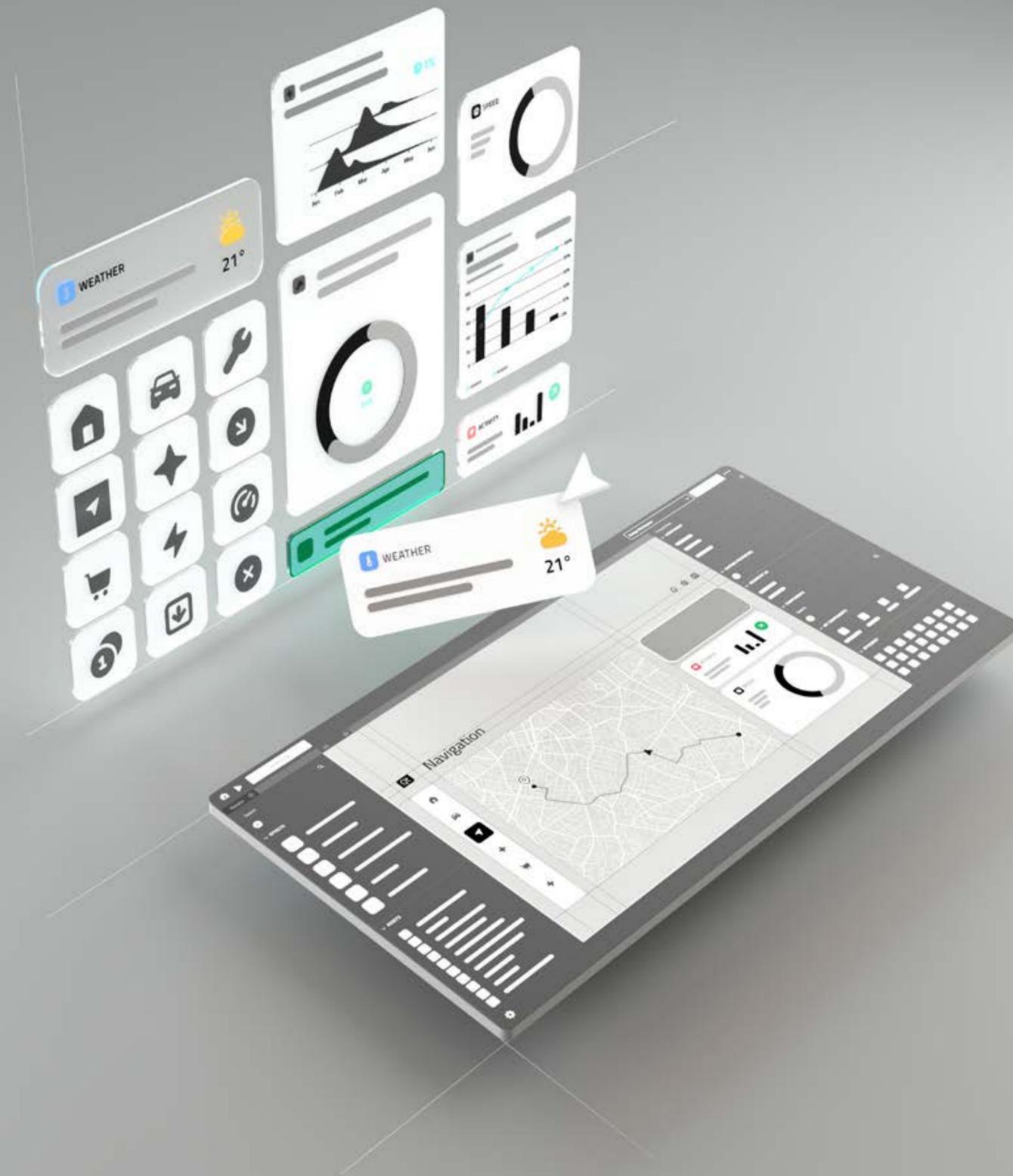
OEM各社は、ハードウェア時代のプロセスのままでは競争力を維持するのに苦心しています。価値の源泉がソフトウェアへと移るとき、必要となるのは、開発アプローチ全体を再編成することで成り立つ新たな原則です。

何十年にもわたり、既存の OEM は製造の原則を磨き上げてきました。ソフトウェア主導の企業は、同じ厳格なマインドセットをコードにも適用し、品質とアーキテクチャを一体の、連続したプロセスとして扱います。

この転換が必要なのは、ソフトウェアがハードウェアとは本質的に異なる、避けられない三つの現実のもとで動いているためです。

- ソフトウェアは常に変化する
- ソフトウェアの問題は加算ではなく乗算で増える
- ソフトウェアは適切に制御すれば継続的な改善を可能にする

これらの構造的な課題は、個別のツールだけでは乗り越えることができません。その重力から抜け出すには、この新しい現実のために設計された原則、すなわちプラットフォームエンジニアリングへ戦略的にコミットすることが必要です。



## プラットフォームエンジニアリングだけが成果を出せる理由

プラットフォームエンジニアリングは、車両の組立ラインで培った原則を、新しいタイプの工場——すなわち「ソフトウェア工場」に適用するものだと考えてください。

物理的な組立ラインが標準化されたツールと再現性のあるプロセスに依存しているのと同じように、エンジニアリングプラットフォームは、開発者に標準化されたツール、テンプレート、API、そしてベストプラクティスを提供します。目指すところは同じです。開発を効率化し、複雑さを抑え、生産性を高めることです。これにより、チームは毎回土台から作り直す必要なく、信頼性が高く一貫した環境の中で、革新的な機能を素早く構築できるようになります。

プラットフォームエンジニアリングは、単なるツールや技術ではなく「原則」です。かつてリーマン生産方式が自動車の作り方を変革したように、プラットフォームエンジニアリングはソフトウェアの作り方を変革します。

[プラットフォームエンジニアリングについて詳しく見る](#) ➔

## データが証明する成果

プラットフォームエンジニアリング戦略の採用は、現代の自動車ソフトウェア開発が抱える課題に対して最も効果的な解決策です。このアプローチをデータで検証するため、Qt Group は2024年に **Forrester Consulting** に市場調査を委託し、この転換がもたらす大きなビジネス効果が確認されました。この調査では、300名以上の組み込みソフトウェアのリーダーおよび実務者を対象にしており、その結果、いくつかの重要な利点が明らかになりました。

- **開発効率が 59% 向上**  
多くの回答者が最も重要な運用上の改善として指摘。
- **コンプライアンスとセキュリティの強化**  
厳格な規制要件への適合と堅牢なセキュリティの両立。
- **保守コストの削減**  
ソフトウェアプラットフォームの長期的な保有コストの低減。

この調査では、自動車業界に特に重要となる利点として、開発者、デザイナー、テスターを含む組み込みソフトウェア開発チーム間の連携が向上することも明らかになりました。

これらの結果を総合すると、プラットフォーム戦略が明確な競争優位性をもたらし、企業がより速く、より安全にイノベーションを進められるようになることが示されています。

## プラットフォームエンジニアリングが実現するもの

従来のアプローチが品質、開発、デプロイメントを別々の課題として扱ってきたのに対し、プラットフォームエンジニアリングは、統合されたエンジニアリングプラットフォームによって、これらを一体的に捉える包括的なアプローチを実現します。その結果、次のようなことが可能になります。

### 標準化は硬直化ではない

ユースケースの **80%** を「ゴールデンプス」に沿って進められるようチームを支援しつつ、イノベーションのための柔軟性を維持します。

### 品質は継続的

開発ワークフローの中でコンプライアンスを継続的に担保できるツールを含んだ、エンドToEndのソフトウェア開発ソリューションを採用します。

### アーキテクチャは常にクリーン

実装が設計どおりであることを継続的に検証するツールを提供し、技術的負債を生むソフトウェアの劣化を防ぎます。

### デフォルトで再利用可能

一度作ったコンポーネントを複数のプログラムで活用できるようになり、**35%**の企業が市場投入までの時間が短縮したと報告しています。

断片的なツールから統合されたプラットフォームへの移行は、品質やアーキテクチャの標準を迂回できなくすることで、「1億行の技術的負債」の根本原因に直接アプローチします。

プラットフォームエンジニアリングの原則についてより詳細はこちらをご覧ください。  
**Laying Golden Paths: How Executives and Engineers Can Align on Platform Engineering.** ↗

# 3

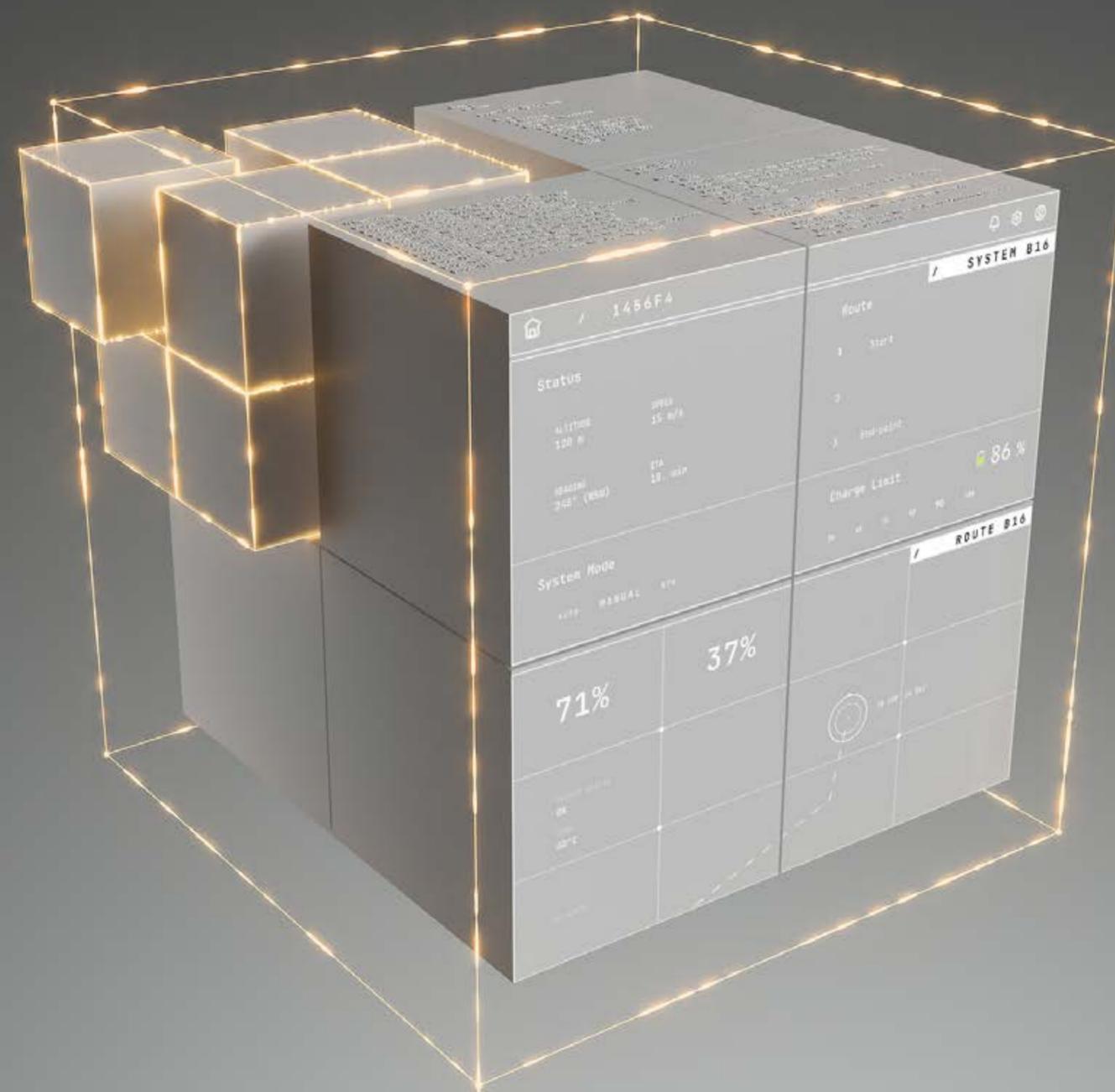
品質・安全性・セキュリティが  
築く基盤

# 品質・安全性・セキュリティが 築く基盤

プラットフォームエンジニアリングは、品質、安全性、セキュリティを開発ライフサイクル全体に組み込むことで、ソフトウェアを第一とする技術分野の模範となります。このアプローチは SDV にとって不可欠であり、自社プラットフォームへのより深い主体的な関与へとつながります。

## 品質はアーキテクチャから始まる

効果的なプラットフォームには、優れたコードだけでは不十分であり、ソフトウェア開発と品質保証を一体化し、連続的なプロセスとして統合するエンドトゥエンドのソリューションが求められます。これはアーキテクチャ上の整合性から始まり、最新の統合ツール群によって実現されます。



## 安全な開発環境を構築するために必要なスタック

最新の SDV 開発には、堅牢な開発ツールチェーンとシームレスに統合された品質保証ソリューションという2つの柱を基盤とした安全なプラットフォームが不可欠です。

### 開発ツールチェーン

- **ユニットテストフレームワーク**でCI環境にて高速ユニットテストを実行
- **プロファイラー**でUIの応答性と滑らかさを最適化
- **スタティックビルド**は高信頼性・安全重視のシステムに最適で、厳格な制御・セキュリティ・認証性を実現
- 詳細な **SBOM** によりソフトウェア依存関係を確認し、脆弱性発生時も迅速に対応
- **長期サポート**でAPIの安定性を長期間保証

### 統合された品質保証

- **アーキテクチャ検証**で設計の逸脱を防止
- MISRA、ISO 26262、AUTOSAR などの**組み込み系コンプライアンス**
- MC/DC サポートに対応する**包括的なコードカバレッジ分析**
- **統合テスト**でGUI、静的解析、パフォーマンステストを一括実施

## 品質向上が生む継続的な価値

品質が開発プロセスに組み込まれていると、後から追加するよりもメリットが積み重なります。

- 不具合は早期発見で、後からの修正コストを大幅に削減
- コンプライアンス文書は手作業なしで自動生成
- 開発者はすぐにフィードバックを受け取れるため、レビューの待ち時間が不要
- アーキテクチャは常にクリーンで保守しやすい状態を維持

プラットフォームエンジニアリングの移行期間において、**43%の組織がセキュリティとコンプライアンスを最大の課題としています**。特に安全性が求められる自動車用途では、開発チームが最初のコードから機器の廃棄までライフサイクル全体の責任を担うことが重要となります。

自動車ソフトウェアテストについてはこちらをご覧ください  
**Automotive Software Testing: Ensuring Quality in the Age of SDVs.** ↗

## 基盤から革新へ——現場で実証された成功事例

ソフトウェア主導の企業へと転換するには、新しいツールだけでなく、根本的な意識改革が求められます。かつてはハードウェアが車の本質を決め、ソフトウェアはその補助に過ぎませんでしたが、今やソフトウェアが機能性を規定し、ハードウェアがその実現を支える役割へと変わっています。

この変革には、ソフトウェアを中心に設計・構築・検証するという徹底した姿勢が不可欠です。プラットフォームの主導権を持つことで、変革は具体的かつ実践的なものとなり、もはや理論上の話ではありません。業界のリーダーたちは、すでにその可能性を現実で証明しています。

Qt World Summit 2025 では、**Harman International** が、ソフトウェアスタックを自社で所有することで、ブランドが求める品質と安全性を維持しつつ、差別化された体験を提供できることを示しました。

Harman 事例のプレゼンテーションを視聴する [➤](#)

自社プラットフォームを UI フレームワークからレンダリングエンジンまで一貫して管理することで、自動車メーカーがどのような重要なメリットを得られるかを示す好例には次のようなものがあります。

- **妥協なきカスタマイズ**: 安全基準を守りつつ、ハードウェアに合わせてフレームワークを最適化
- **将来を見据えたアーキテクチャ**: ベンダーに縛られることなく、長期的な進化を自分たちでコントロール
- **シームレスな統合**: 独自のイノベーションとパートナーのソリューションをつなげる
- **データに基づく機能強化**: 実際の車両データをもとに、継続的に品質を向上

### 重要なポイント

プラットフォームの主導権を握ることで、SDV の成功を左右する迅速なイノベーションサイクルが可能となります。土台を自ら管理すれば、革新のスピードも自在にコントロールできます。

# 4

プラットフォームとコードベースの  
掌握がイノベーションを生む

# プラットフォームと コードベースの掌握が イノベーションを生む

プラットフォームとコードベースの管理は、現在の課題を解決し、未来の可能性を広げます。アーキテクチャの主導権が革新的な能力を生み出す、重要な3つの分野について見ていきましょう。



## アーキテクチャ管理で AI と ADAS を極める

自動運転技術に注目が集まる中、AI は車載ソフトウェアのあらゆる領域に革新をもたらし、その複雑性を飛躍的に高めています。開発チームにとって、これは安全性やコンプライアンスの見落としにつながり、厳格な安全規制のもとで品質保証が極めて困難になるという課題を生み出しています。

例えば、AI や ADAS の機能は GPU による高速処理コードに大きく依存しており、従来の静的解析ツールでは解析できません。そのため、現在の品質管理ツールで確認できないコードの機能安全性を証明することが困難となっています。

ご存知でしょうか。**ガートナー社**によると、2027年までにサイバー攻撃の17% が生成 AI を活用したものになると予測されています。

プラットフォームレベルでの管理やコードの完全な可視化がなければ、新たな脅威にどう立ち向かうことができるでしょうか？

深層コード解析を伴うアーキテクチャ管理により、これらの課題は解決できます。AI 搭載システムの ISO 26262 適合性を証明し、ADAS のアップデートもアーキテクチャの一貫性を保ちつつ実施、さらに安定性を損なうことなく機械学習モデルの改良が可能となります。

<sup>5</sup> Gartner Forecasts Global Information Security Spending to Grow 15% in 2025 [➔](#)

複雑なシステムで ISO 26262 の適合性に苦戦していますか？  
[こちらのガイドをご覧ください](#)



### 重要ポイント

AI は自動車イノベーションの次なるステージですが、安全性と法規制順守を両立させるには、プラットフォーム全体を統括するアーキテクチャ管理が不可欠です。

## Android Automotive の真価を引き出す

Android Automotive OS が OEM 各社に広く採用される中、標準化されたプラットフォーム上で実装品質とブランド独自性をどう確保するかが新たな課題となっています。

多くの場合この課題は、サプライチェーンにおける「ブラックボックス」モデルに起因します。貴重な知的財産を守るため、サプライヤーはソースコードを公開せず、コンパイル済みバイナリとしてアプリケーションを提供します。これは業界で一般的な慣習ですが、最終的に多数の統合システム全体で一貫した高品質なユーザー体験を保証する責任は、OEM にあります。

プラットフォームを自社で持つことで、統合に関する課題を解決するための次のようなツールが手に入ります。

- ソースコードがなくても動作を検証できる包括的なテストフレームワーク。<sup>6</sup>
- サプライヤー各社が提供するコンポーネント間で統一されたユーザー体験。
- 標準プラットフォーム上で実現するブランドの差別化。
- システム全体を再構築することなく部品を柔軟に入れ替えられるアーキテクチャ。

### 重要なポイント

プラットフォームの所有権を持つことで、品質とブランドの一貫性を確保する手段が得られ、標準的な Android Automotive OS を独自の競争力へと変えることができます。

## データドリブンなユーザー体験の創造

SDV 時代における究極の競争優位性は、実際のデータに基づき絶えず進化する、卓越したユーザー体験です。このデータサイクルは、OTA (Over-the-Air) アップデートとプラットフォーム制御によって実現し、将来の収益やブランドの信頼につながる原動力となります。

OTA 市場は、2032 年までに 130 億ドル超に成長すると予測されていますが、このチャンスを手にできるのは、データを自在に管理できるアーキテクチャを備えた OEM だけです。<sup>7</sup>

しっかりとしたプラットフォーム基盤があれば、車両は学習するシステムへと進化します。これにより、次のようなことが可能になります。

- 車両の反応性を高める**パフォーマンス向上**の導入。
- 実際の HMI 利用状況をもとに**将来の機能開発**に活かす。
- ユーザー体験全体を向上させる**予測保守**などの新サービスの提供。

### 重要なポイント

アーキテクチャのコントロールにより、リアルなデータを活用した継続的な改善サイクルが実現し、ユーザー体験を常に進化させ、新たな OTA による収益機会を獲得できます。

<sup>6</sup> Qt - Android Automotive 向け UI/GUI テストの簡素化とスケール化 [➤](#)

<sup>7</sup> GM Insights - Automotive Over-The-Air (OTA) Update Market Size [➤](#)

顧客成功事例

 Elektrobit Axivion

# Elektrobitが アーキテクチャ管理で 開発を加速した方法

品質を損なわずにスピードを追求するために

**Elektrobit** の EB Street Director ナビゲーションソフトウェアは、**Audi**、**Porsche**、**Mercedes-Benz**、そして **Volkswagen** などの車両に搭載されています。これらのプレミアムブランドは、卓越したパフォーマンスと迅速な機能提供を求めています。

100名以上の開発者が世界各地から一つのモジュール型プラットフォームで共同作業する中、**Elektrobit** は「高いスピードで新機能を追加しながら設計の一貫性を保つ」という典型的な課題に直面しました。新しい開発者はシステム理解に数週間かかり、機能追加のたびに既存の動作への影響リスクが生じます。

継続的なアーキテクチャ検証による課題解決

**Elektrobit** は、自社のコードがUMLベースの設計通りになっているかを自動で常時チェックできる、独自のアーキテクチャ検証ソリューションを導入しました。違反があれば、統合テストの何ヶ月も前にすぐ発見できます。

この透明性によって開発プロセスが大きく変わりました。チームはスピードと保守性のバランスをしっかりと見極めて判断できます。開発者自身が自分のコードが全体にどう影響するかを把握でき、新しいメンバーもドキュメントよりツールを使って早く馴染めるようになりました。

## 顧客成功事例

### 成果：構造化がスピードを生む

この変革は、明確な効果をもたらしました。

- 機能の迅速にリリース可能—変更による逸脱を即時に把握できるため
- 新規開発者の早期戦力化(数週間の生産性向上)
- 実際のアーキテクチャへの影響に基づく正確なプロジェクト見積もり
- 進化を続けても保守性を維持するモジュール型アーキテクチャ

Elektrobitは、アーキテクチャの原則が開発速度を高めることを実証しました。明確な境界設定でテストしやすくなり、継続的な検証で技術的負債を防ぎつつ、自動車業界の安全基準も守られています。

#### 重要ポイント

Elektrobitは、このプレイブックの中核となる原則を体現しています。アーキテクチャを制御することは、イノベーションのスピードを握ること。分散チームやレガシーシステムを抱えるOEMにとって、継続的なアーキテクチャ検証により複雑さが整理され、品質が自動的に確保され、イノベーションも繰り返し実現できるようになります。

[導入事例を読む](#) ➔



# 5

## 進むべき道 – 変革を加速する4つの実践ステップ

# 進むべき道 – 変革を加速する 4つの実践ステップ

成功のために、これまでの投資を手放す必要はありません。プラットフォームの変革は段階的に進めることで、各ステージで価値を生み出すことが可能です。

段階ごとの内容は、常に自社の目標に合わせて調整されるべきですが、成果を上げている変革は確立されたパターンに従っています。このロードマップを示すため、**テクノロジー戦略ディレクター Miao Luo** に、4つの重要なステップについて知見を聞きました。

## ステップ4

成果を拡大して共有

## ステップ3

オーナーシップのためのスキルと文化を醸成

## ステップ2

プロトタイプ組織の構築

## ステップ1

長期的な経営コミットメントの確保

### ステップ1

#### 長期的な 経営コミットメントの確保

変革には揺るぎないリーダーシップが求められます。5年以上のロードマップ、確保された予算、そして専任の SDV 推進者がトップダウンで変化を導き、業務に支障をきたさず推進します。

「長期的なコミットメントの欠如は変革の最大の障壁です。多くの既存 OEM はコスト増加の初期段階で挫折してきました。私にとって真の変革とは、困難な道を確認を持って乗り越えること——その苦労が会社のより良い未来につながると深く信じる姿勢を意味します。」

### ステップ2

#### プロトタイプ組織の構築

リーダーの賛同を得たら、次のステップは SDV のコンセプトを基幹生産を変えずに試すことです。

「少人数で部門横断型の特命チームを編成し、既存の車両モデルに新しい SDV 機能を搭載しましょう。まずはソフトウェア OTA (SOTA) から始め、次にファームウェア OTA (FOTA) に進めます。現実の環境でコンセプトを証明する、リスクの低い商用パイロットを目指すことが重要です。」

### ステップ3

#### オーナーシップのための スキルと文化を醸成

変革を成功させるためには、現有の人材を活かし、新たなマインドセットを育むことが不可欠です。

「アジャイル開発の研修を実施し、ツールを標準化し、SDV マインドセットを育てましょう。パイロットプログラムからのフィードバックをもとに改善を重ねることで、既存人材の力を最大限に引き出し、専門性を高められます。何よりも、現場の開発チームが自律的に判断し、実行できる環境を整えることが不可欠です。オーナーシップの文化は個人から始まるもので、社内報のローガンだけでは根付きません。」

### ステップ4

#### 成果を拡大して社内外に共有

モデルの有効性が証明された後は、ステークホルダーの自信を維持しつつ、ビジネス全体の転換へと進みます。

「実証された SDV 機能を複数モデルに拡大し、新たなプロセスを本格生産ラインへ取り入れましょう。これらの成功実績は、社内チームへ、そして社外の投資家・顧客へもしっかり発信することで、勢いを維持し続けることが重要です。」



— Miao Luo  
Qt Group テクノロジー戦略ディレクター

## ソフトウェア主導シフトの投資対効果

前述の通り、ソフトウェア主導の変革を導入・実施することのビジネス的根拠は、明確なデータによって裏付けられています。Qt Group 委託による2024年の Forrester Consulting の調査を引用すると、業界リーダーたちは、最新のプラットフォーム志向の手法導入がもたらす影響について質問を受けました。

これらの企業がこれまでに得た、または今後得られると見込まれる主要な効果は以下のとおりです。

このコンセプトに関する詳細はこちらをご覧ください。

[What Big Firms Can Learn from Smaller Teams About Platform Engineering.](#) ↗

50%

新しい機能の市場投入期間が短縮

68%

エンドユーザー体験が向上

57%

組み込みソフト全体でブランドのアイデンティティと一貫性の強化

49%

不具合率の低下。特に自動車製造業や自動車関連産業の回答者で顕著な傾向。

この優れた成果は、プラットフォーム主導の戦略が、大規模な組織にも小規模チームのような俊敏性をもたらし、それをエンタープライズ規模で実現できるというシンプルな原則から生まれています。

まとめ

# まとめ— 今こそ選択すべき時

自動車業界のルールは、ソフトウェアによって一新されました。競争優位性は、アーキテクチャの主導権と継続的な品質という2つの柱のもと、優れたソフトウェアを生み出す原則から生まれます。

このプレイブック全体を通じて、イノベーションを阻む主な壁として「1億行の技術的負債」に向き合ってきました。プラットフォームの主導権とコードベースの管理こそが戦略的解決策であり、専門家による変革へのロードマップも提示しています。

前進するために明白かつ喫緊の選択が必要です。

選択肢1:現状維持で受け身に徹する

複雑さを断片的で手間のかかる手法で管理し続けると、ソフトウェアのリコールによる利益低下や、セキュリティ問題によるブランド価値の損失、機敏な競合他社が自社では実現できない新機能で市場を奪うのを目の当たりにすることになります。

選択肢2:技術基盤を自社で掌握し、未来を切り開く

モビリティの未来を自ら描きましょう。車両を作って終わりの製品から、顧客に新たな価値を生み続ける進化するプラットフォームへと生まれ変わらせます。単発の取引モデルから、継続的に価値を提供する関係へとシフトし、イノベーションの速さを自ら決めるテクノロジーリーダーになることができます。

この変革を成し遂げた OEM こそが、次の 10 年のモビリティを牽引します。決断を先送りすれば、ためらわず進んだ競合企業から、技術のライセンス供与を受ける立場に甘んじることになるでしょう。

# 次のステップ

自動車業界は大きく変革しました。車はソフトウェアで定義され、データが収益を生み出します。プラットフォーム主導権の有無が先駆者と追随者の分界点となります。

今のプラットフォーム成熟度を診断し、変革のロードマップを描く戦略的コンサルティングをご検討ください。

## コンサルティングの予約

[qt.io/contact-us](https://qt.io/contact-us)



# Qt Group について

**Qt Group** (Nasdaq Helsinki: QTCOM) はグローバルなソフトウェア企業です。産業界のリーダーと **150万人** を超える世界中の開発者が信頼を置き、ユーザーに愛されるアプリケーションやスマートデバイスを作成しています。

UI デザインやソフトウェア開発から品質管理と導入まで、製品開発サイクル全体を通してお客様の生産性向上を支援します。Qt Group のお客様は **70以上の業界**で **180か国**以上に広がっています。Qt Group の従業員数は世界で約 **900名**、2024年の売り上げは **2億910万ユーロ**でした。

詳細は Web サイトをご覧ください。 [www.qt.io/ja-jp/](http://www.qt.io/ja-jp/)