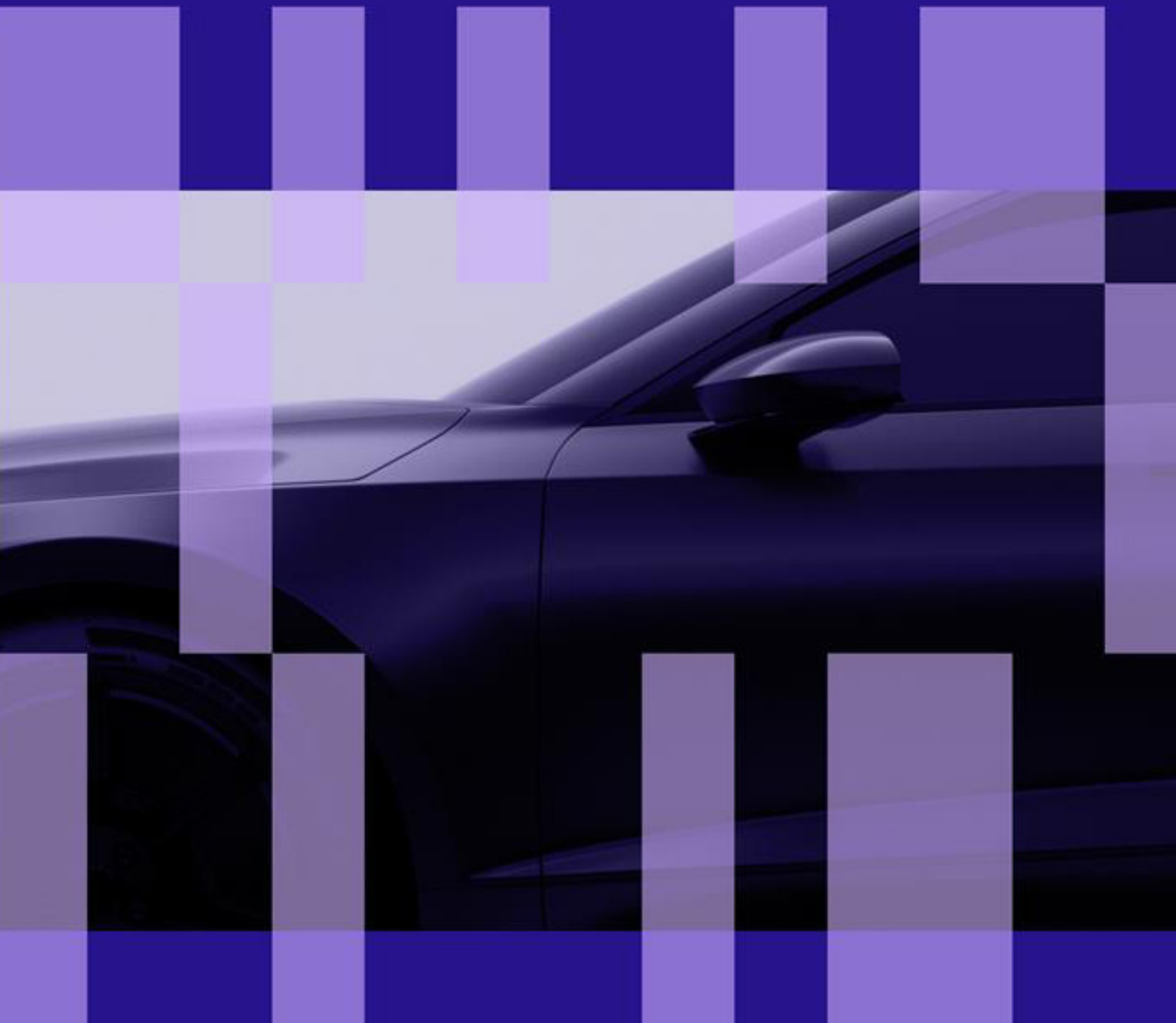
Qt Group

GUIDE

自動車向け機能安全 ISO 26262 に準拠

ASIL-A, ASIL-B, ASIL-C と ASIL-D



目次

機能安全のエキスパートにお任せください

安全性が求められるソフトウェア開発において、ISO 26262への準拠は容易ではありません。

当社のエキスパートは、自動車業界での豊富な経験を持ち、お客様の課題解決をサポートいたします。

ツールやサービスの詳細、またはデモのご依頼については、お気軽にお問い合わせください。

お問い合わせ
https://www.qt.io/ja-jp/quality-assurance/axivion-suite

はじめに

ISO 26262は、IEC 61508を自動車分野向けに適用した国際規格であり、道路車両における電気・電子（E/E）システムの機能安全を確保することを目的としています。この規格では、A（最も緩やか）からD（最も厳格）までの4段階の自動車安全度水準（ASIL: Automotive Safety Integrity Level）が定義されており、それぞれのレベルに応じて安全要件の検証の厳密さが求められます。

Axivion Suite は、Axivion 静的コード解析と Axivion アーキテクチャ検証を組み合わせたソリューションであり、ASIL-DまでのISO26262要件に対応するための理想的なツールです。静的コード解析と設計検証に重点を置きながら、ISO 26262:2018で定義された検証ステップをカバーしています。

ソフトウェア定義車両（SDV）が主流となり、車載ソフトウェアがますます複雑化する中で、Axivionのようなツールは、安全性、トレーサビリティ、コンプライアンスの維持に不可欠です。Axivionは現代の開発パイプラインに統合可能であり、安全性を後付けではなく、ソフトウェアライフサイクル全体を通じて継続的かつ検証可能なプロセスとして確保することを可能にします。

Axivion静的コード解析は、ISO 26262に基づく安全システム開発において、ASIL Dまでの使用が可能であることを、SGS-TÜV Saar GmbHによって認証されています。

セーフティケースへの対応

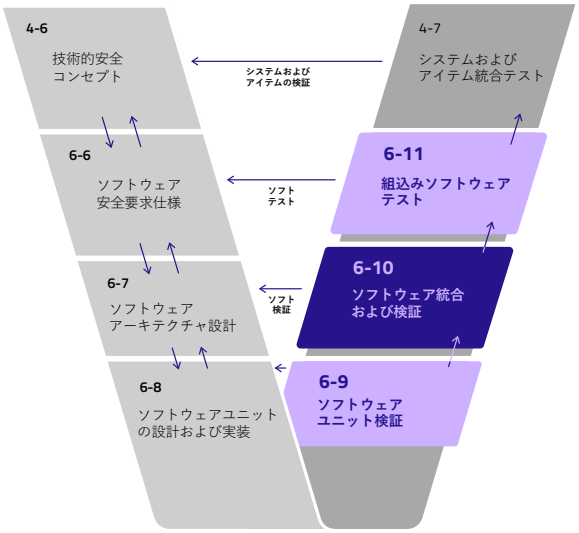
ISO 26262 におけるセーフティケースとは、E/E システムがその想定用途に対して十分に安全であることを、構造化された証拠ベースの議論によって示すものです。これは、要件定義、設計、実装、検証、妥当性確認、構成管理といった開発ライフサイクル全体を網羅します。開発プロセスそのものが、機能安全の達成に直結する重要な要素となります。

Axivionは、トレーサブルで再現可能、かつ認証可能な解析を提供することで、セーフティケースに直接貢献します。開発ワークフローに統合することで、静的コード解析やアーキテクチャ検証を一貫して適用でき、各プロジェクト固有の安全目標への適合を証明する支援を行います。

Axivionをセーフティケースに組み込むことで、認証機関による監査やアセスメントの効率化が可能になります。詳細なレポートの自動生成や過去の解析データの保持により、透明性と説明責任を確保でき、安全性が求められる開発における適切な対応を裏付けることができます。

Vモデルにおける位置づけ

Axivionは、ソフトウェアレベルの製品開発を規定するISO 26262-6に準拠しており、Vモデルにおける各フェーズを幅広くサポートします。具体的には、ソフトウェアユニットの検証、ソフトウェア統合、組込みソフトウェアのテストといった工程に対応しています。



Axivionは、QMからASIL A～Dまでのすべての安全度水準に対応しており、安全要件の異なるプロジェクトにおいても一貫したツール運用が可能です。この柔軟性により、開発チームは複数の安全クリティカリティを持つコンポーネントを横断して、効率的かつ整合性のある開発を実現できます。

さらに、AxivionはVモデル全体にわたるトレーサビリティをサポートしており、アーキテクチャ要素やコード成果物を要件やテストケースと関連付けることが可能です。これにより、影響範囲分析や変更管理が容易になり、開発ライフサイクル全体を通じて安全要件が確実に満たされていることを証明できます。

モデリングおよびコーディングガイドラインへの対応

Axivion Suite は、モデリングおよびコーディングガイドラインの自動適用を通じて、人的ミスのリスクを低減し、チームやプロジェクト間で一貫した安全設計の実践を可能にします。これにより、コード品質の向上だけでなく、開発ワークフローにベストプラクティスを組み込むことで、コンプライアンス対応の迅速化にも貢献します。

Axivionは、安全性準拠において重要とされる以下のようなソフトウェア開発原則の遵守を自動的に支援し、開発プロセス全体にわたって一貫性と品質を確保します：

- 低複雑性の維持
- 信頼性の高い設計原則の適用
- 曖昧さのないグラフィカルな表現の使用
- プログラミング言語のサブセットの使用
- 強い型付けの徹底
- 防御的な実装技法の活用
- スタイルガイドの遵守
- 命名規則の適用
- 並行処理に関する考慮

ISO 26262-6:2018に基づくモデリングおよびコーディングガイドラインで取り扱うべきトピック

		ASIL				Axivion Suite							
	項目	A	B	C	D	アーキテクチャ検証	コードクローンの検出と管理	循環依存の検出	デッドコード解析	メトリクス監視	コーディングガイドライン	静的および意味的コード解析	データ競合の検出
1a	低複雑性の維持	++	++	++	++	•	•	•	•	•	•		
1b	信頼性の高い設計原則の適用	+	+	++	++	•	•	•		•			
1c	曖昧さのないグラフィカルな表現の使用	+	++	++	++	•							
1d	プログラミング言語のサブセットの使用	++	++	++	++						•		
1e	強い型付けの徹底	++	++	++	++						•	•	
1f	防御的な実装技法の活用	+	+	++	++		•				•	•	
1g	スタイルガイドの遵守	+	++	++	++						•	•	
1h	命名規則の適用	++	++	++	++	•					•		
1i	並行処理に関する考慮	+	+	+	+							•	•

これらの原則は、アーキテクチャ解析、コードクローンの検出、循環依存の検出、到達不能コードの特定、メトリクスの算出といった機能によって支えられています。また、Axivion Suiteは、MISRA C、MISRA C++、AUTOSAR C++14、CERTなどのコーディング規約の適用も強制し、業界のベストプラクティスに沿った開発を実現します。

システムレベルおよび日常開発における解析

Axivionは継続的インテグレーション(CI)システムとシームレスに統合でき、自動かつ再現可能な解析を実現します。これにより、トレーサビリティと一貫性が向上します。また、差分解析機能により、時間経過に伴う変更の追跡が可能となり、リグレッションの検出や影響範囲の分析を支援します。

日常の開発作業においては、統合開発環境(IDE)内でのローカル解析をサポートしており、即時フィードバックを提供することで、修正サイクルの短縮に貢献します。このように、Axivionはシステム全体を俯瞰する戦略的な視点と、開発者の日々の作業を支える戦術的な支援の両立を実現します。

システム全体と開発者視点の解析を組み合わせることで、チームはアジリティを損なうことなく高いソフトウェア品質を維持できます。開発者は開発初期段階で有益なインサイトを得られ、プロジェクトリーダーや安全管理者は、システムの健全性やコンプライアンス状況を包括的に把握することが可能です。

アーキテクチャ解析

Axivionのアーキテクチャ検証機能は、設計仕様に対する実装の整合性を検証することを可能にします。設計仕様は、ボックス&アロー図のような非形式的なものから、UMLモデルのような形式的なものまで対応可能です。また、仮説駆動型のアーキテクチャリカバリ機能により、レガシーシステムやドキュメント化されていないコードベースからアーキテクチャを再構築することもできます。

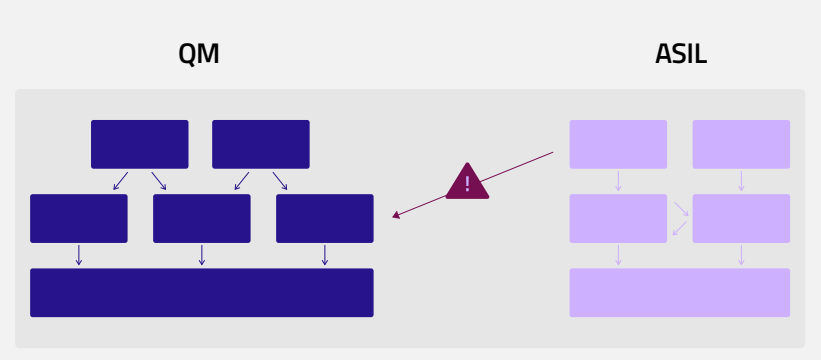
AUTOSARベースのプロジェクトにおいては、Axivionはarxmlファイルのインポートと、IBM Rhapsody や Enterprise Architectなどのツールで作成されたアーキテクチャモデルとの整合性検証をサポートします。これにより、複雑なモデル駆動型環境においても、設計意図と実装の一貫性を確保できます。

Axivionアーキテクチャ検証は、設計と実装のギャップを埋めることで、長期運用される自動車ソフトウェアにおいて頻発するアーキテクチャの劣化を防止します。これにより、システムはライフサイクル全体を通じて、保守性、拡張性、安全性、そして性能目標に沿った状態を維持できます。

Freedom from Interference (FFI)

FFIは安全性を重視したプロジェクトにおいて重要な目標であり、ISO 26262-1:2018で定義されています。これは、異なるASILレベルのコンポーネント同士が互いに悪影響を及ぼさないことを保証するために不可欠です。Axivionは、異なるクリティカリティ（ASIL分類やASILとQMのパーティション間）を持つコンポーネント間での誤った関数呼び出しなど、ソースコードレベルでの意図しない相互作用を検出します。これにより、開発後期の動的実行フェーズにおけるMMU/MPU例外の発生頻度を低減し、開発時間の短縮に貢献します。

さらに、Axivionの可視化ツールは、ソフトウェアモジュール間の依存関係を開発者やアーキテクトが把握しやすくすることで、安全なパーティショニング戦略の設計と検証を支援します。これは、特にマルチコアやミックスド・クリティカリティシステムにおいて、干渉リスクが高まる場面で非常に有用です。



これらの問題を早期に検出することで、Axivionはランタイム例外の発生を未然に防ぎ、開発後期におけるコストのかかるデバッグ作業の削減に貢献します。このようなプロアクティブなアプローチにより、堅牢なパーティショニングおよび分離戦略が実現され、システム全体の安全性向上につながります。

要件の分解

ISO26262-9:2018で定義されているASIL分解は、安全要件を独立したアーキテクチャ要素に分配することを可能にします。Axivionは、ソフトウェアアーキテクチャが必要な独立性を備えているかを検証し、分解された要件が正しく実装され、適切に分離されていることを保証します。

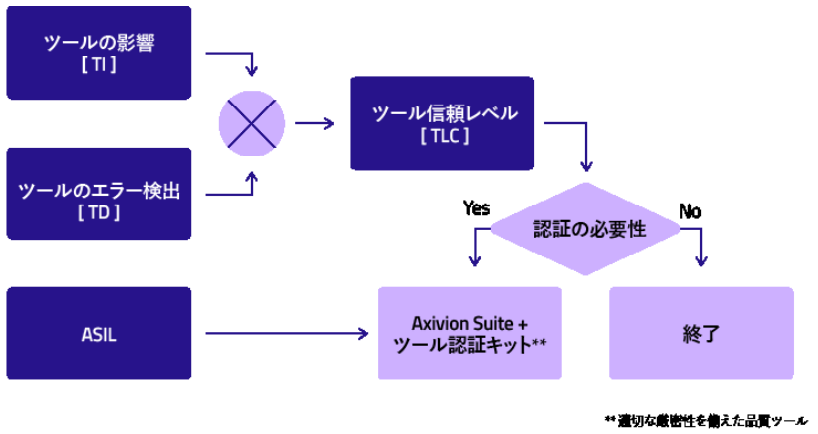
この機能により、特にASIL Dの完全な実装が現実的でない、または不要なシステムにおいても、コスト効率の高い安全戦略を維持しながら、規格への準拠を実現できます。

Axivionはコードレベルでアーキテクチャの独立性を検証できるため、分解戦略が理論上だけでなく、実際の実装においても確実に適用されていることを保証します。これにより、設計上の潜在的な欠陥リスクを低減し、プロジェクト間で再利用可能なモジュール型コンポーネントの構築を支援します。

ツール認証

安全性が重視される開発において、ソフトウェアツールの使用にはツール認証が不可欠です。要求されるツール信頼レベル（TCL）に応じて、Axivionはその出力結果が追加の検証なしに信頼できることを証明する必要があります。

Axivionの静的コード解析ツールは、MISRA、CERT、AUTOSAR C++14に対応した事前定義済みのコード違反やテストケースを含む、包括的なツール認証キットを提供しています。このキットは、あらゆるASILレベルでの認証をサポートし、ツールチェーンが必要な信頼性要件を満たしていることを保証します。認証プロセスは、詳細なドキュメントとテスト成果物によって支援されており、開発チームの認証作業を簡素化します。これにより、ツール認証に伴う負担を軽減し、安全性やコンプライアンスを損なうことなく、プロジェクトのスケジュールを加速させることが可能です。



Qt Group

https://www.qt.io/ja-jp/quality-assurance/axivion-suite