

# 自動運転車は 安全になるか？

寄稿

Christopher Giordano  
Vice President UX/UI Technology  
The DiSTI Corporation

**DiSTI**<sup>®</sup>

Jim Carroll  
Chief Technology Officer  
Digica

**DiGiCA**  
Intelligent Digital Solutions



## 自動運転車は安全になるか？



### 自動運転車のUI機能安全

一見すると、自動運転車にはユーザーインターフェイス (UI) の機能安全に関する重要な要件がないように見えるかもしれません。自動運転車では、ドライバーによる制御があったとしても少なくなり、UIの重要性が低下します。残っているこれらのコントロールは、エンターテインメントに関連している可能性が高く、機能安全に準拠する必要はありません。

ただし、従来のようなドライバーによる制御はそれほど重要ではなくなる可能性があります。自動運転車の性質により、車両制御に新たな追加要件がもたらされます。また、自動運転車の標準化も一夜にして行われることはありません。これは、従来のドライバーが制御する車と自動運転車が道路を共有する必要があることを示しています。この移行期間中、自動運転車は、安全性の考慮が必要な、ドライバーによる制御を必要とします。このホワイトペーパーでは、これらのUIとは何か、および、機能安全のニーズに対応するためのソフトウェアアーキテクチャの開発と作成プロセスにおける課題と潜在的なソリューションについて説明します。

### 自動運転車に必要なUI

UIがデザインのどの部分に入るかは、最近いくつかの議論があり、UIインターフェイスのサブコンポーネントによって異なります。機能安全ASIL (Automotive Safety Integrity Level) 規格のUIの場合、歴史的に最も重点が置かれているのは、インストルメントクラスターの重要なコンポーネントです。これらは本質的に、車両の動作、エラー、および動作を制御する統合された車両部品に接続します。これらのシステムは、「壊滅的な」障害の環境を作り出し、人命を失う可能性があります。

最近まで、特に機能安全に関連するため、ドメイン毎に非常に異なるUI要件がありました。自動運転車の急速な普及により、自動運転車のさまざまなコンポーネントの要件とユースケースを作成する人々にとって、ASIL規格のUIは全く新しい意味を持ちます。





## インフォテインメント

例えば、インフォテインメントシステムは、歴史的に統合されていないシステムの一部です。ここで障害が発生した場合、移動中の車両に安全上の脅威をもたらすことは、通常ありません。サブシステムの統合（複数のディスプレイを管理する1つのSoC）により、インフォテインメントはコックピットにおいてより重要な役割を果たし始めています。

機能安全要件も、クラスター、および、ヘッドアップディスプレイ (HUD) から、複数のディスプレイを実行する能力と機能を備えた単一のヘッドユニットシステムに移行しています。ハードウェアの観点からは効率的ですが、ソフトウェア開発の観点からはいくつかの課題があります。これについては、このホワイトペーパーで詳しく説明します。ハードウェアとディスプレイ技術の進歩により、量販車でもHUDを採用できるようになりました。これは、UIとソフトウェアアーキテクチャの意思決定に新ラウンドをもたらしました（フロントガラスに表示するそのデータのASILはどのレベルでしょうか?）。クラスターとHUDの両方がインフォテインメントシステムでより多くの役割を持ち始めるにつれて、企業の適切な安全文化とともに、機能的に安全なアーキテクチャの提供が、ブランドの成功にとって最も重要になっています。

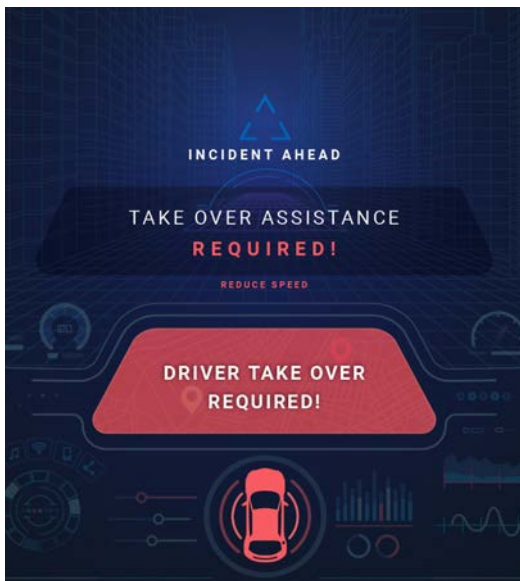
## リモートコントロール

リモート車両コントロールもASILを必要とするUIの有力な候補ですが、UIがどうあるべきかについて明確ではなく、一貫性もありません。リモート機能の検討項目はいくつかあります。

- » 車外のドライバーによる、リモートリクエストキーフォブや接続されたスマートフォンからの、単純な低速自動駐車や情報呼び出し
- » 車内のドライバーによる、レーンキープアシスト、緊急自動ブレーキ、アダプティブクルーズコントロールなどの、走行中の運転操作の自動切替えイベント

前述の最初のケースで定義されているリモート機能は、通常、車両へのお任せ式 (Fire & Forget) コマンドです。ただし、UIは依然として明確で、一貫性があり、信頼できるものである必要があります。車両の動きを制御することにより、あるレベルのASILが適用されますが、UI自体には適用されない可能性があります。

前述の2番目のケースで定義されているリモート機能は、車両操作の切替えに関するものです。ドライバーはこの状況で、これらの動作を止めることができないか、止める時間がない可能性があるため、ここでも、安全で信頼性の高いアーキテクチャが重要になります。



## 緊急解除

自動運転車でUIが非常に重要であることは間違いありませんが、緊急解除の概念は「引き継ぎイベント」とも呼ばれます。あなたの車が時速120Kmで高速道路を走っているところを想像してください。何らかの原因により、突然、あなたの車が状況判断できない状態に陥りました。その時点で、車両は引き継ぎイベントを呼び出し、すべての重要なシステムの状態をドライバーに知らせ、生命にかかわる決定を下す時間を与えます。その情報は通常、聴覚的、視覚的、あるいはその両方で、車両からドライバーに伝えられます。多感覚統合に関する研究は、聴覚と視覚へのデータの組み合わせにより、平均反応時間が改善され、よりタイムリーな認知反応が得られることを示しています。<sup>1</sup> したがって、認知伝達を改善するためには、明確で簡潔で信頼性の高いUIが非常に重要になります。

部分的なシステム化をスキップして完全な自動運転車を実現することで、引継ぎイベントの必要性を排除することについては、長年の議論があります。このホワイトペーパーでは、さまざまな自律レベルとその意味について、次のセクションの後でさらに詳しく説明します。

# 完全な自動運転車に至る長い道のり

前述のこれらのドメインのシステム要件の定義と同様、自動運転車の機能は進歩し続けており、UI に対する要件はまた流動的です。1つ確かなことがあります。システムの複雑さが増すにつれて、UI およびソフトウェアアーキテクチャで、機能安全はますます重要な役割を果たします。同時に、従来の車両と自動運転車の連続体のギャップが狭まります。自動化なしから部分的な自動化、そして、最終的には完全な自動化への移行期間には、明確な定義が必要です。

自動運転に関しては、Society of Automotive Engineers (SAE)<sup>2</sup> によって定義された、5つの異なるレベルがあります。



Constant supervision

## LEVEL 0 手動運転

ドライバーが全ての運転操作をする。



All aspects and some tasks during some circumstances

## LEVEL 3 条件付きの自動化

車両の自動運転システム(ADS)は、特定の状況下で、全ての運転操作を制御することができる。緊急時など、ADSが手動運転を要求した場合、ドライバーは運転操作を引き継ぐ必要がある。特定の状況下以外では、ドライバーが運転操作をする。



Steering OR Braking

## LEVEL 1 運転支援

車両の先進運転支援システム(ADAS)は、ステアリング操作、または、加減速のどちらかを制御することができるが、同時に両方を制御することはできない。



All aspects and all tasks during certain circumstances

## LEVEL 4 高度な自動化

車両の自動運転システム(ADS)は、特定の状況下で、運転環境を監視しながら、全ての運転操作を制御することができる。ADSの運転操作中は、ドライバーは介入しない。



Steering and braking during some circumstances

## LEVEL 2 部分的な自動化

車両の先進運転支援システム(ADAS)は、状況により、ステアリング操作と加減速の両方を制御することができる。ドライバーは、常に十分な注意を払い(運転環境を監視しながら)、残りの運転操作をする必要がある。



All aspects and all tasks during all circumstances

## LEVEL 5 完全自動化

車両の自動運転システム(ADS)は、あらゆる状況下で、運転環境を監視しながら、全ての運転操作を制御することができる。ドライバー不要で、人間は単なる乗客となる。

先に述べたように、多くの人は、一気に自律性を実現することがより安全なアプローチであり、すべての車両をレベル5に直接移行する必要があると信じています。理論的にはそれが理想的かもしれませんが、コスト、市場投入までの時間、システムテスト、そのような技術に対する一般の人々の安心のレベルなど、他の考慮事項があります。

自動運転車の採用に対する人々の態度は、依然としてさまざまです。取得した大量のデータに基づいて生命にかかわる運転の意思決定を行うために、我々は学びながら、新しい人工知能システムを実装しています。この事実は、レベル5の自律性を実現する前に、安心、かつ、少なくとも部分的に進む方法があることを強調しています。現在、レベル5に確実に飛躍するためには、あまりにも多くの要因が関係しています。例えば、次のような質問です。

- » 使用されているテクノロジーはフェイルセーフですか?
- » 車両を設計および製造するOEMおよびTier1の安全文化は受け入れられますか?

- » 自動運転車は、まだ道路を走る非自動運転車と、どのように相互作用しますか?
- » 動作中の車両システムに、ある程度の人間による制御が必要ですか? もし必要であれば、それは何ですか?
- » 完全な自動運転車の事故責任は誰にありますか?
- » 誰が中心になって、創造性とコストのバランスを取ることができる範囲で、OEM開発に義務付ける標準を定義しますか?

自動運転車は、安全性、機動性、経済的メリットが証明されていますが、標準として完全なレベル5に近づいていくと、1つの要件が明らかになります。少なくとも短期的には、自動運転車に関して、人間によるある程度の制御または介入機能が残るでしょう。この問題は、私たちが技術を証明するまで続く可能性が高く、非常に安全な環境の実現には、人間による制御の排除が必要と考えられています。

2 - Lynberg, M. (2020, June 15). Automated Vehicles for Safety. Retrieved from <https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety>



## 自動運転車のUIで重要なこと

企業は、見栄えが良く、ブランディングと一致するUIを常に求めています。UIのルックアンドフィールは、あるブランドと別のブランドを区別するための主要な差異化点であると主張する人もおられます。そのため、主要な要件の1つに、「見栄えがよい」インターフェイスであることという暗黙の了解が常にあります。これは、設計者ごとに主観的に異なる意味合いを持っています。

安全上の理由から、UIは直感的でシンプルで、注意を乱すものや操作時間を最小限に抑える必要があります。これらのUIに付随する機能安全のレベルに関しては、OEMおよびサプライヤーによる解釈の余地があります。NHTSA、DOT、SAE、およびECのガイドラインがありますが、特定の自律レベルに従って遵守するための厳格な機能安全分析のルールはありません。これらのガイドラインは変更され、自律性レベルが上がるにつれてより厳密になるはずですが、純粋な自律性

は、そこに到達したときに重要な独自のレベルがありますが、移行中の短期間に自動運転車のUIに不可欠なものは、安全性の観点から、シンプル、パフォーマンス、および、安全の3つです。

- » **引き継ぎイベント中において、認知的負荷と重要な情報伝達量を最小にするシンプルさ。**
- » **重要なコンテンツを期待どおりに描画するためのランタイムパフォーマンス。**
- » **重要なコンポーネントを保証する機能安全は、システムの安全設計戦略によって実現される。**

## 運転の未来へ

UIの観点については、実績のある商用ソフトウェアツールと、安全文化を念頭に置いて適切に設計されたソフトウェアアーキテクチャにより、前述のすべてのポイントを達成することができます。ハードウェアとソフトウェアは進化を続け、より効率的になり、より多くのデータをより高速に処理します。そのため、全てのシステムの全ての開発プロセスを1つにまとめることはできないでしょう。むしろ、異なるプロセスを許容し、将来にわたって最新の技術を適用する柔軟性を持つことが、OEMおよびTier1サプライヤーの成功の基礎となります。

例えば、10年前には、1台の組込み機器で複数のディスプレイを管理することは、性能的にも経済的にも容易ではありませんでした。ハードウェアが、より小さく、より速く、より少ない電力で動作して、バス全体に大量のデータを伝送できるようになると、各ディスプレイに個々のSoCを割り当てるのではなく、1つのSoCで複数のディスプレイを管理するというアイデアが生まれ、経済的にも実現可能になりました。このハードウェアアーキテクチャの変更により、ASILが適用されないシステム (IVI) と、ASILが適用されるシステム (Cluster 及び HUD) を、個々のシステムを障害のリスクにさらすことなく、安全に統合することができるソフトウェアアーキテクチャが必要となりました。このソフトウェアアーキテクチャを実現する上で生じる技術開発の課題は、柔軟性の高い商用ツールを使用することで軽減することができます。



## 商用ツール

機能安全のために設計されたツールは、そうでないツールよりも高価であるという認識があります。隣接市場ツールの活用は、自動車OEMのソフトウェア認証のためのコストを抑え、かつ、非反復エンジニアリングコストの削減をもたらします。

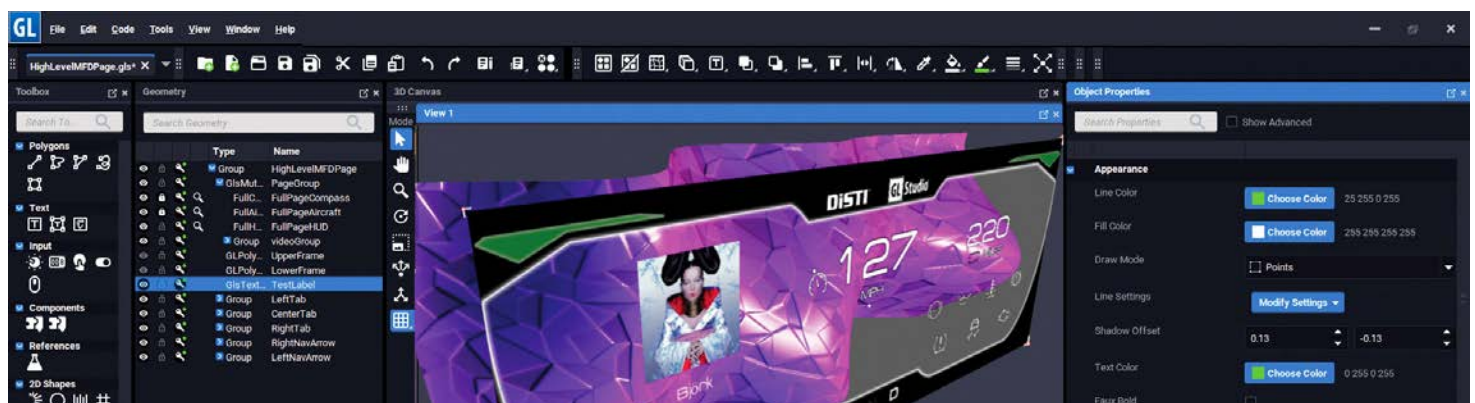
### 「Aviation Week & Space Technology」 2017年11月掲載、Rockwell Collins

Rockwell Collinsの答えは、自動車業界標準の側面の1つであるリスクベースのシステムです。アビオニクスと自動車にとって、そのメッセージは明確です – 安全に多くのお金が費やされています。OEMは、安全基準を維持し、創造性を阻害することなく、コストを削減するにはどうすればよいでしょうか。すでにコストの矢面に立たされているのであれば、機能安全のために隣接市場でその安全が証明されている、適切な商用ツールを活用してください。

「以前は、新しいアビオニクスシステムでは、1ドルあたり75セントがエンジニアリングに費やされ、25セントが国際機関による認証に費やされていました。今日、その比率は大幅に変化しています。」<sup>3</sup>

自動車用ソフトウェアの認証は、過去10年にわたって自動車OEMに努力を強いてきました。認証の取り組みは、自動車の安全基準を改善することができますが、イノベーションと創造性を抑制し、開発コストを上昇させる懸念があります。

また、認証の取り組みは、官僚主義を助長するという側面を持っており、自動車OEMが開発スケジュールの短縮を目指している中で、逆に開発スケジュールを延長させる原因となっています。



半導体ベンダーは、自動車セクターで競争力を獲得するために、機能安全準拠のツールを提供することの重要性を認識しています。Armは、多くのSoCに同社のシリコンIPを採用してもらうために、セーフティクリティカルなアプリケーション向けのコンパイラパッケージを提供しています。これは、半導体ベンダーが最小限の投資でこのパッケージをサポートできることを意味します。また、このセクターで競争力を維持するためには、他のSoCおよびシリコンIPベンダーが、ドミノ倒しの、同様のサポートを提供する必要があることを示しています。

リスクを少なく、低コストでイノベーションを促進するためには、他のドメインで既に採用されている同様のツールを活用するアプローチが適切です。たとえば、セーフティクリティカルな分野において20年以上の実績を持つGL Studio UI Toolkitは、現在も運用中の航空機、宇宙船、およびヘリコプターに搭載され、世界中のライフクリティカル医療機器で使用されています。

GL Studioは、2015年4月にISO 26262規格のASIL-Dグレードに認定された、最初のUIツールでもあります。また、原子力施設のNQA-1事前認証にも合格しています。これは、アビオニクスのためのFAAのDO-178C DALA認証よりも、さらに厳格なプロセスです。<sup>4</sup>これらのツールの開発コストは、自動車の過去10年間で同じように、30年以上にわたって同様のセーフティクリティカルニーズを持ってきた、さまざまな業界のさまざまな市場と顧客によって処理されます。したがって、より厳密なテストとより優れたパフォーマンスによって大幅に高い品質を生み出す航空などの隣接市場向けの経験と既存のツールを活用することは、低リスクで低コストのソリューションになります。

ADASから自動運転への移行が間近に迫っている現在、これらのすべてを合せることにより、はるかに信頼性の高いソフトウェアアーキテクチャを生み出すことができます。これとUIは何の関係がありますか？ このホワイトペーパーで前述した引き継ぎイベント（緊急解除）です。

3 - Statler, K. L. (2017, November 13). The World Needs Seamless Aviation Certification Standards. Aviation Week & Space Technology.

4 - HMI and UI Design Software: Embedded Target Systems. (2019, June 12). Retrieved August 28, 2020, from <https://glstudio.distil.com/features/safetycritical/>

## 組み込みソフトウェア

ソフトウェアにおけるUIの機能安全のサポートは2つの主要なドメイン：セーフティクリティカルなアプリケーション作成を可能にするプラットフォーム、ユーザーが対話するアプリケーション、により構成されます。

プラットフォームソフトウェアは、GPU、ミドルウェア実装プロトコルとコーデック、アプリケーション層に一貫したAPIを提供する抽象化層など、基盤となるハードウェアコンポーネントを直接制御するデバイスドライバやその他のBSPコンポーネントなど、多くのコンポーネントで構成されています。Khronosなどの組織は、OpenGL、OpenGL ES、Vulkanなどのアプリケーション開発者が使用する業界標準のAPIを定義しています。これらのAPIのセーフティクリティカル (Safety-Critical: SC) 版はすでに利用可能であり、新しいバリエーションが開発されています。これらのAPIをサポートするためのセーフティクリティカル準拠のドライバは、GPUベンダーから入手可能になっています。

車両機能間でハードウェアリソースを共有する場合、機能間の分離を実現するために、通常、ソフトウェア仮想化を活用します。このような環境では、ホストOS、ゲストOS、および、ハイパーバイザが機能安全に準拠している必要があります。

また、ハードウェア仮想化やGPU分離の機能を有するSoCも存在し、ハードウェアを含めた全体のアーキテクチャの検討に影響を与えます。このホワイトペーパーでは、この点については言及せず、ソフトウェアアーキテクチャに絞って解説します。

ソフトウェア仮想化環境は、右の図のアーキテクチャが示すように複雑になります。このアプローチは、製造前の欠陥解決のプロセスを複雑にして、開発サイクル全体を長くしてしまう可能性があります。



セーフティクリティカル準拠のツールとソフトウェアコンポーネントは急速に成熟しています。中期的には、ハイパーバイザがそれほど重要でなくなり、ソフトウェアアーキテクチャが簡素化され、セーフティクリティカルに準拠したシステムの構築が容易になると予想されます。また、ツールのSC版は、SC版ではないツールと同じ価格となる可能性があり、追加コストなしで、セーフティクリティカルに準拠できるようになります。これらにより、完全にセーフティクリティカルに準拠したシステムの開発作業量、市場投入までの時間、および、ソフトウェアライセンスのコストの削減を図ることができるようになるでしょう。

車両UIアプリケーションは、車両の乗員が車両と対話するためのユーザーインターフェイスです。UIアプリケーションには、クラスター、環境制御、インフォテインメントなどの機能が含まれます。このようなUIは、セーフティクリティカルなAPIとGL Studioなどの機能安全に対応したUIツールを使用して、機能安全に対応したアプリケーションとして構築することができます。

適切な機能安全ソフトウェアコンポーネントの選択は、コードをプログラムする前のプロジェクトの設計段階で重要です。ベースとなるテクノロジーを開発サイクル中に変更すると、多大な技術的負債と追加の統合作業を発生させ、結果として、開発したソフトウェアの機能安全を損う可能性があります。



## 自動車安全水準 (ASIL)

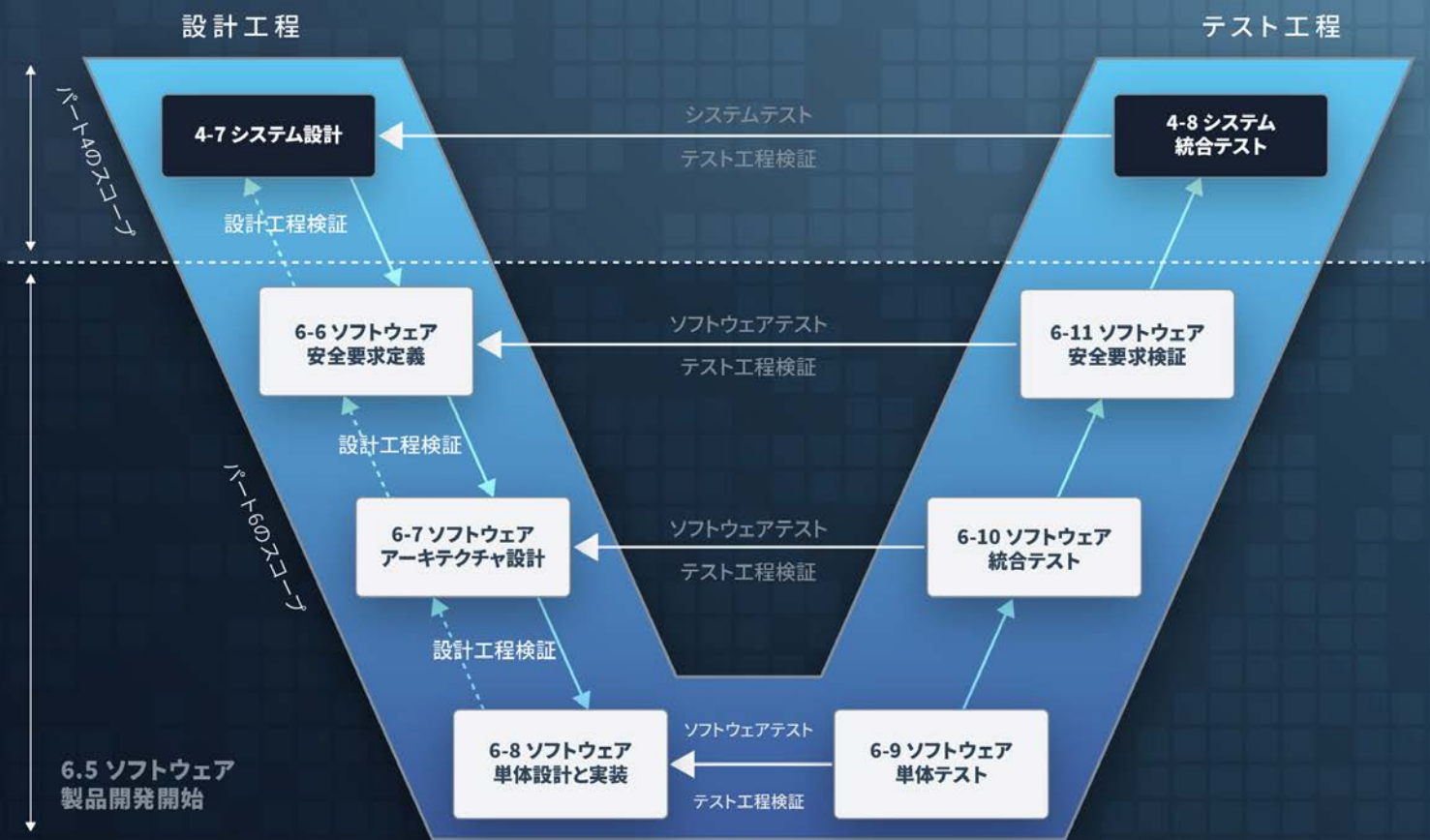
機能安全の鍵はリスクの評価です。市販車のユーザーのリスクを最小限に抑えるために、車両の開発中に厳格なプロセスが課せられます。ASIL (Automotive Safety Integrity Level) として知られるリスク分類スキームは、このようなリスクを評価するために頻繁に使用されます。ASIL A (最低) からASIL D (最高) という4つのリスクレベルがあります。ASIL Dとして評価されたハザードは生命を脅かすリスクを表しており、そのような開発は最も厳しい開発プロセスの対象となります。

一般に、ASILを使用する場合に適用する原則は、業界標準と規制の

枠組みに準拠しながら、追加コストを発生させない最高レベルで作業することです。このようなアプローチにより、製品の信頼性が向上します。また、要件が変更された場合の柔軟性の確保も可能になります。安全性が重要な環境でソフトウェアを開発するコストを考えると、その寿命を延ばし、長期的なコストを削減することが重要です。

ASILは、より広範な規格であるISO26262のサブセットであり、自動車業界の製品開発プロセスのガイダンスも提供します。車両UIの開発におけるこのような措置の遵守は、機能安全を保証します。

## ISO 26262におけるシステムおよび組込みソフトウェアのV字プロセス



このプロセスは、ASILで特定されたリスクが文書化され、ソリューションが適切に設計、実装、およびテストされるように設計されています。「V字プロセス」は、いくつかの重要な機能を強調する規格によって説明されています。

- » 文書化された要件、設計、および実装によるトレーサビリティを含むエンジニアリングプロセス。
- » テストサイクルのピアプロセスによる、開発の各フェーズの出力の構造化された検証。
- » ASILが特定したリスクに対処する、システムの安全要件に特に焦点を当てます。

V字プロセス (およびそのバリエーション) はよく理解されており、組込みソフトウェア開発プロジェクトで一般的に展開されています。その使用法は、インフォテインメント、クラスター、HUDなどのアプリケーション、UIライブラリ、グラフィックスドライバを含む、全ての自動車アプリケーションスタックコンポーネントの作成に等しく適用されます。

すべてのUIソフトウェアコンポーネントは、セーフティクリティカルに準拠するために、これらの標準に従って開発する必要があります。





## GPU

最新のUIは非常に複雑であり、展開されたすべてのコンテキストで、最新のオペレーティングシステムの隅々まで浸透しています。UIエクスペリエンスは、メニューシステムにおけるOEMのスプラッシュ画面からナビゲーションアプリケーションに至るまで、ユーザーが車両に足を踏み入れた瞬間からユーザーエクスペリエンスの最前線にあります。ユーザーエクスペリエンスは複雑で目立つため、パフォーマンスは非常に重要です。最新のUIはすべて、GPUを含むハードウェア上に構築されているため、アプリケーションプロセッサからグラフィック関数をオフロードできます。

SoC企業がサードパーティからGPU設計をライセンスして、チップに含めることが多いように、GPUは複雑です。Imagination TechnologiesとArmは、「ライセンス可能な」GPU設計の主要ベンダーです。他のSoC企業は、アプリケーションプロセッサと十分に統合されたGPUを設計しています。このアプローチは、Nvidia、Qualcomm、Intelなどの企業によって採用されています。

業界は、ソフトウェア統合を簡素化するために、Khronosなどのグラフィックプログラミング用の標準APIを定義するための組織を形成しました。OpenGLは、デスクトップPCのために、ほぼ30年前に初めて定義されました。基盤となるハードウェアが進化し、組み込みシステムで使用するためのOpenGL ESなど、いくつかの関連するAPIを生み出したため、多くの改訂が行われました。

## その他の考慮事項

自動車のOEMは、継続的に車両のBOMコストの削減に取り組んでいます。このアプローチは、一部のSoCを削除する場合があります。これは、車両機能間で物理プロセッサを共有することを意味します。このホワイトペーパーで前述したように、セーフティクリティカルな機能とセーフティクリティカルでない機能が単一のプロセッサに共存する場合、そのプロセッサがホストするすべての機能で、単一のセーフティクリティカルなUIソフトウェアスタックを使用する方法が簡単です。

歴史的に、車載システムは標準のSoC SKUに基づいていました。SoCは非常に複雑なデバイスであるため、比較的高価です。特定のアプリケーションでは、FPGAやMCUなどの低コストのシリコン上にシステムを確立しながら、高品質のUIとセーフティクリティカルの要件を達成できる場合があります。いくつかのISVは、ソフトウェアレンダラーを提供してGPUを削除しました。OS間でGPUを共有する複雑さがなくなる他、GPUはSoC中の最大のプロセッサであるため、システムのフットプリントを小さくできます。

CoreAVIなどの企業がリリースする、セーフティクリティカルに準拠したAPIは、GPUベンダーのドライバと同様に利用することができます。Microsoftは、競合APIであるDirect3Dを採用しています。Windowsはインフォテインメントシステムで一般的に使用されていますが、そのようなシステムのほとんどはLinuxベースであり、Khronos APIを使用しています。Appleは、競合APIであるMetalを定義しています。現在、セーフティクリティカルに準拠したDirect3DまたはMetalの版はありません（Metalは、まだ商用の自動車システムに導入されていません）。

GPUは一般的な計算にも使用されるため、Khronosは計算APIであるOpenCLも定義しています。彼らは、Vulkanと呼ばれる、これまでのAPIセットの代替技術を開発しました。このメソッドは、グラフィック、埋め込み、およびコンピューティングAPIを統合します。Vulkanは、やがて古い「兄弟」を置き換える可能性があります。これは確実ではありません。プログラミングモデルは、以前のモデルよりも大幅に複雑です。以前はGPUドライバにカプセル化されていたロジックの多くを、グラフィックスとコンピューティングを単一のAPIに統合するために、アプリケーション開発者、または、UIツールとライブラリのベンダーが実装する必要があります。これにより、ツールベンダーの機能をさらに差別化できますが、これまでハードウェアの依存関係から抽象化されていたコンポーネントに、ハードウェア関連の複雑さが大幅に追加されます。

2019年、Khronosは、セーフティクリティカル準拠版のVulkan APIの開発を開始しました。

OpenGL ESなどのAPIの導入時点では、ソフトウェアレンダラーは車載のSC環境で十分なパフォーマンスを発揮できませんでしたが、今日、そのパフォーマンスは大幅に改善されています。GPUを含むFPGAの可能性は、アーキテクチャ上の選択肢をさらに増やしています。

MCUは通常、UIを含まないことが多い「小さな」アプリケーションに使用されます。これらは、ブレーキシステムなどのSCコンポーネントのECUのプラットフォームとして、車載システムで既に頻繁に使用されています。FPGAと同様、MCUも処理能力が向上し、ソフトウェアレンダラーがより幅広い車載アプリケーションで使用できるほど十分に機能するようになりました。MCUの電力要件はSoCやFPGAよりもはるかに低くなっています。これにより、コストも削減される可能性があります。

ハードウェアを選択することでコスト削減が可能かもしれませんが、ソフトウェアのライセンスや開発などの関連コストへの影響を考慮することが重要です。

## 結論

レベル5の完全自動運転車の採用が近づき続けるにつれて、これを安全に達成する必要があることは明らかです。このホワイトペーパーで定義されている多くの理由により、自動運転車への道のりは長く、私たちがそれに向けて努力し続けていることは明らかですが、完全なレベル5の実現は私たちの近い将来にはありません。レベル5への移行には多くの要因が関係しています。純粋な自律性から得られるメリットはたくさんありますが、自動車業界はインフラの問題、社会

経済的問題、人間の感情、そして最も重要なこととして、このペーパーで示した技術的課題を克服する必要があります。

私たちは進歩を続ける必要がありますが、このペーパーでは主に安全で経済的に実行可能な方法を取り上げました。モデルベースデザインに関するよく知られた研究で述べられているように、「ソフトウェアプロジェクトにおいて、統合フェーズでの欠陥の特定と修正に、開発コストの80%が費やされています<sup>5</sup>」。

## なぜモデルベースデザインなのか？

### 機会損失

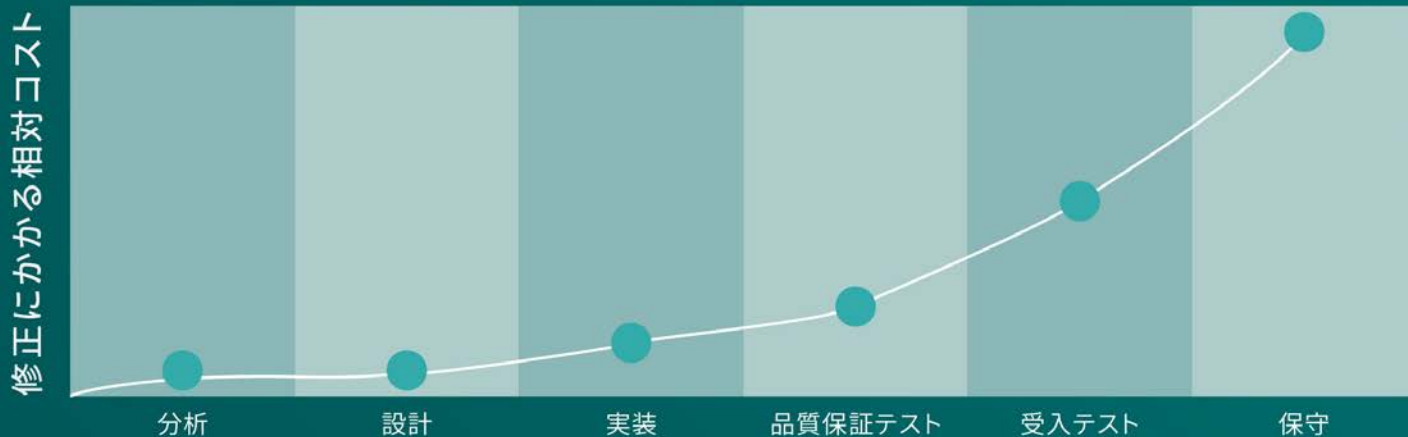
市場投入が6か月以上遅れると、組織は**5年間のROIの33%**のコストがかかる  
(出典：マッキンゼーグループ)

### コスト

- ・ **開発コストの80%**が、統合中の欠陥の特定と修正に費やされる
- ・ **IT開発予算の40%以上**が不十分な要件によって消費される  
(出典：IAGコンサルティング)

### 従来の品質保証テスト

- ・ **納期までの25~30%**がテスト  
(出典：IBMの調査)
- ・ 上流の品質の悪さが手戻りを生む
- ・ スケジュール短縮がそれを悪化させる



Software quality optimization: balancing business transformation and risk, Michael Lundblad, program manager, Rational software, IBM Software Group, Moshe Cohen, offering manager, Rational software, IBM Software Group

初期費用の考慮において、一般的に受け入れられている原則ではありませんが、適切なハードウェア、ソフトウェアのアーキテクチャ、およびワークフローに前もって投資することが重要です。ソフトウェア設計を将来にわたって保証し、新しいテクノロジーやアーキテクチャに柔軟に対応するためには、事前にコストをかける必要があります。上記の調査を考慮すると、適切に準備をした場合、プロジェクトの存続期間にわたって、大幅に労力とコストを節約できます。

組込みグラフィックソフトウェアを開発する際の主な目標は安全性です。さらに、創造性を阻害したり予算を超えたりしない、安全なソフトウェアをサポートしている既存のCOTSツールを活用しながら、適切な先行アーキテクチャ設計をすることで、コストとリスクとフィールドのバランスを取ることができます。

## 寄稿者について



**Christopher Giordano**  
Vice President UX/UI Technology  
The DiSTI Corporation



1997年以来、Chrisは、米国海軍とセントラルフロリダ大学でUIとHMIソフトウェアの開発に注力してきました。Chrisは、1999年からDiSTIで、60を超えるさまざまなプログラムのリードエンジニア、または、プログラマネージャーとして、そして、最終的にはすべてのDiSTIのUI開発ツールのプロダクトマネージャーとして働いてきました。Chrisは、ボーイング、ヒュンダイ、ジャガーランドローバー、ロックード、NASA、日産自動車、ノースロッププログラマン、スペースシップカンパニー向けのDiSTIのHMI/UIプログラムを管理し、現在は、DiSTIのUX/UIテクノロジーの担当役員を務めています。彼は10年以上にわたってDiSTIのUX/UIビジネスの管理に成功し、HMI/UIおよび機能安全の専門家として、グローバルなリーダーシップを発揮してきました。Chrisは、ノースカロライナ大学で財務の学士号を、セントラルフロリダ大学でコンピューターエンジニアリングの学士号を取得しています。



**Jim Carroll**  
Chief Technology Officer  
Digica



Jim Carrollは、組み込みシステムのバックグラウンドを持つ経験豊富なテクニカルディレクターです。Jimは、ソフトウェアエンジニア、テクニカルアーキテクト、CTOとして、25年以上にわたり、数百万人のユーザーにリーチする大規模なソフトウェアソリューションを設計および構築してきました。Jimの仕事の大部分は半導体セクターであり、Intel、Arm、Imagination Technologiesなどの企業と協力しています。彼はまた、ソフトウェア標準グループであるKhronosの寄稿者でもあります。

