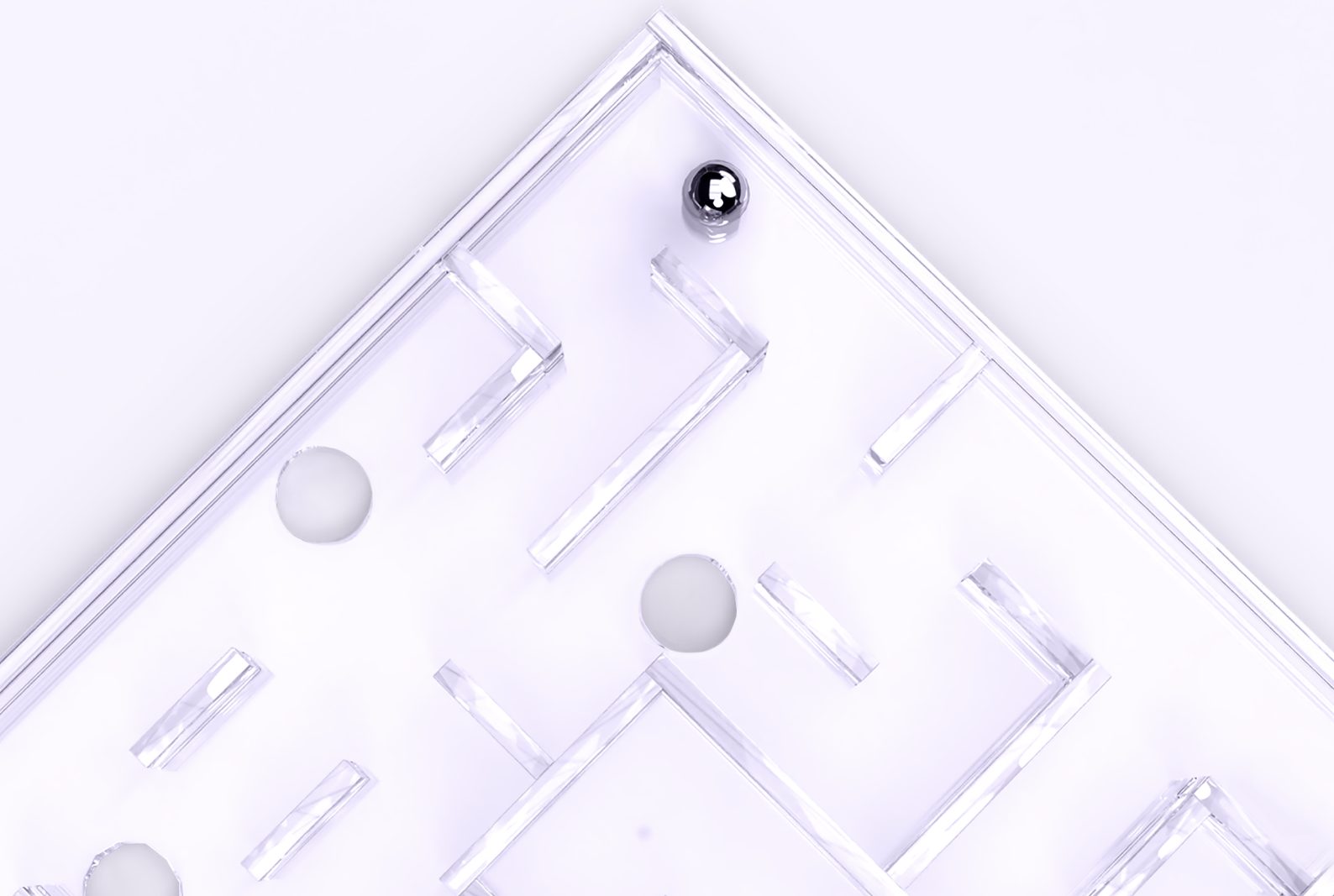
Applied Intuition

安全な自動運転システムの構築

自動運転システムの検証 ・ 妥当性確認 (V&V) のためのハンドブック

第1版 • 2022年7月



目次

エグゼクティブサマリー

A.チャレンジ

自動運転・自律走行システムを安全に開発・デプロイすることは、大変な課題です。自動運転プログラムは、開発するシステムの安全性を確保するために、厳格な検証と妥当性確認(V&V) のプロセスを定義し、それに従う必要があります。V&Vプロセスは、実環境試験、テストコース試験、シミュレーションなど、さまざまな環境において、設計要件に照らし合わせてシステムをテストするためのものです。残念ながら、自動運転・自律走行システムは、運用空間において安全に取り扱うべき条件が制限されていないため、特に複雑なものとなっています。その上、システムの安全性と、安全な量産展開に向けたプログラムの進め方に関する規制ガイダンスは限られています。これらの課題により、一部の自動運転プログラムでは、システムの開発が進むことでV&Vプロセスの確立が容易になると考えられており、開発の初期段階における　V&Vへの投資が躊躇されがちです。しかし、自動運転プログラムにおいては、基礎となる開発とテストの実践方法を、複数のチームに跨って確立することが最善です。早期に適切な基盤を構築し、時間をかけてV&Vプロセスを成熟させることで、チームは明確に定義された目標に対してより効率的にシステムを開発し、開発等の遅延を回避し、より安全で性能の高い製品を最終的に構築することができるようになります。

B. ゴール

堅牢なV&Vは、機能開発そのものと同じくらい重要です。開発および検証戦略を定義する場合、自動運転プログラムは2つの目標に向かって尽力する必要があります。

I. **安全性を論証するための枠組みを確立する：**どの安全原則に従うべきかを概説します。安全性フレームワークは、後に自動運転プログラムがセーフティケース（システムの安全性を正当化する構造化された論拠）を構築し、システムの性能が検証され、危険を許容レベルまで最小化するという規制当局および一般市民の間の信頼を構築するのに役立ちます。
II. **包括的なV&Vプロセスを導入する：**安全な製品を合理的な時間で検証するために、堅牢なプロセス群を設定します。

このハンドブックは、安全性フレームワーク (I. 安全性フレームワークのベストプラクティス) およびV&Vプロセス (II. V&Vのベストプラクティス) を設定するためのベストプラクティスを自動運転プログラムに対して提供することを目的としています。

このハンドブックの第1部 (I. 安全性フレームワークのベストプラクティス)　では、6つの中核的原則を定義しています。そして、自動運転プログラムが、安全性フレームワークの基礎として、これらの原則を採用することを提案しています。これらの原則は、製品が安全に設計されていること、機能的に安全であること、意図された機能の安全性を説明すること、根拠に基づく安全性を示すこと、強力な安全ガバナンス構造を持つプログラムによって生産されること、および安全に運用されることを規定しています。このフレームワークに従うことで、自動運転プログラムは厳格なデータ、トレーサビリティ、ドキュメントを備え、自動運転・自律走行システムが量産展開できるほど安全であると、社内、規制当局、一般市民を説論することができます。

このハンドブックの第2部 (II. V&Vのベストプラクティス) は、開発および検証を通して、自動運転プログラムが参照できる実践的なガイドを提供します。このパートは、自動運転プログラムが以下のような主要なトピックについてアドバイスを必要とするときに、積極的に参照できるようにすることを目的としています。

I. Vモデルとアジャイルを組み合わせたハイブリッドな製品開発プロセスの構築方法
II. 正式なリリース検証プロセスおよび安全性ガバナンスボードの構築方法
III. 包括的なシミュレーションシナリオライブラリの構築方法
IV. シミュレーション、走行、実環境のテスト環境を使用した、シナリオベーステストの経済的かつ効果的な実行方法
V. 運用設計領域 (ODD) のあらゆる可能な側面を包括的にテストすることなく、自動運転・自律走行システムの安全性が展開可能なレベルかどうかを判断する方法

Applied Intuition は、自動運転 ・ 自律走行業界における独自の地位・経験を活用して、このハンドブックを作成しました。長年にわたり、当社のチームは幅広い業界経験を積み、お客様の妥当性検証と安全性の目標達成に協力し、最新の

研究を調べ上げ、規制当局や標準化委員会と協力し、新しいツールやプロセスを開発して、お客様をサポートしてきました。

このハンドブックは、自動運転プログラムの安全性フレームワークを定義し、V&Vのベストプラクティスを実施してシステムを安全に開発し、テストし、商業化するための発展的なリソースとして機能するものです。また、業界全体のステークホルダーとのコミュニケーションを喚起することも目的としています。私たちはこのようなコミュニケーションを楽しみにしており、このハンドブックの今後のバージョンアップのためのご意見をお待ちしています。

C.プレビュー

以下は、このハンドブックで学ぶことができるトピックのほんの一例です。

V&Vのステージ

自動運転プログラムは、その開発と検証の取り組みにおいて、さまざまな段階にあります。このハンドブックの 「V&Vのステージ」では、安全ガバナンス構造、セーフティケース、リリース検証プロセス、要件管理、テスト手法、カバレッジ解析、性能解析など、さまざまなV&Vの次元にわたって、初期、中期、後期の自動運転プログラムがどのようなものになるかを示しています。

このハンドブックでは、「V&Vのステージ」 の後に、初期、中期、後期の自動運転プログラムのベストプラクティスを個別に説明しています。

カバレッジ

このハンドブックの「カバレッジの定義と測定」のセクションでは、自動運転プログラムがシステムを十分にテストしたかどうかを定性的および定量的に判断する方法について説明します。このセクションでは、カバレッジを「既知のもの」と「テストされたもの」の比率と、「システムが直面する可能性のある状況の総和」として定義しています。

各能力、シナリオカテゴリ、および要件のテスト数に従ってカバレッジを測定することは有用でありますが、このセクションでは、 ODDの定義に基づくアプローチについて説明します。ODD分類法（ODDは属性とパラメータのセットを使用して定義）を使用すると、プログラムはODDレベルでカバレッジを評価でき、どのような種類のシナリオにカバレッジを追加する必要があるかを理解できるようになります。このセクションでは、情報理論の観点からより厳密にカバレッジを評価する統計的カバレッジメトリクスと、プログラムレベルでカバレッジを測定および評価することの主な利点について考察します。

シナリオ作成

本ハンドブックの「シナリオ作成」のセクションでは、自動運転プログラムが開発段階に応じて一般的にどのようにシナリオ作成に取り組むかを示しています。初期段階の自動運転プログラムでは、通常、要件とシナリオのカテゴリーにまたがる広い範囲を構築することに重点を置いています。広い範囲を構築した後、後々のステージを取り組むチームはエッジケースシナリオの収集と生成、および新しい領域への拡張に重点を置いています。

テストケースの手続き的な側面はさることながら、シナリオをテストケースに変えるのは、システムの性能をテストする評価基準の指定です。自動運転プログラムは、各テストケースについて、測定可能な総合的な合格・不合格の結果を追跡する必要があります。この結果は、主要な能力、安全性、快適性の要素の合成であり、任意でない評価ルールはすべて合格しなければならず、それぞれとその基礎となるメトリクスを掘り下げることができるものです。
「評価基準と評価基準の定義」 のサブセクションでは、チームがテストケースに対して評価すべき評価基準をリストアップしています。

はじめに

A.自動運転システム検証の課題

自動運転プログラムは、自動運転システムが信頼でき、安全に使用できることを、自分自身、規制機関、そして世間一般に示す必要があります。したがって、これらの規制ガイドラインを満たす堅牢で包括的なV&Vプロセスは、自動運転システムの開発と商業化を成功させるために不可欠です。しかし、従来の自動車、航空、航空宇宙産業などの確立された分野とは異なり、自動運転・自律走行システムのV&Vは、より複雑な課題に直面する新興の分野です。新しい先進運転支援システム（ADAS、すなわちSAEレベル1-2システム）や自動運転システム（ ADS、すなわちSAEレベル3-5システム）は複雑なODDを対象とし、有限のテストセットで検証できる以上のシナリオを扱うと予想されます。その結果、自動運転システムの予備設計では把握できない可能性があり、テストされていない未知のハザードが常に存在することになります。ADASとADSには広範なハードウェアコンポーネントがありますが、ソフトウェアのV&Vは、複雑ではるかに大きな動作空間のために新たな問題を提起しています。

B.規制の状況

本稿執筆時点では、ADAS と ADS の規制については現時点では比較的簡素ですが今後進化する可能性のあるフレームワークが存在します。 以下の概要は、自動運転規制の状況を高いレベルで説明しています。以下の様々なフレームワークの説明は、規制プログラムの現状を要約したものですが、決して包括的なものではありません。

米国の規制状況自動運転システム(ADS)

ADS を適切に検証することは、各自治体プログラムの責任です。現在の規制ガイダンスは、 ADS 開発の生産および生産後の段階において V&V 手法を採用することを示唆しています。連邦政府の規制ガイダンスはある程度存在しますが、自動運転プログラムが ADS を検証する際にどの具体的な方法論に従うべきかが不明確なことが多いのが現状です。

例えば、航空業界では、米国連邦航空局 (FAA) が特定の業界基準を定め、航空機メーカーにその基準による認証の取得を要求しています。 これに対し、米国道路交通安全局 (NHTSA) は、現在、自動運転プログラムに対し、ADSの機能安全やシステム安全に関する規制を要求していません。その代わり、NHTSA は ADS の開発者や製造者が自ら、あるいは NHTSA のウェブサイトを通じて公開する自主的な安全性自己評価 (VSSA) を通じて ADS の「自己証明」を認めています。NHTSA の VSSA テンプレートは、自動運転プログラムが安全への取り組みの要約に使用できる情報を提案していますが、その開示を義務付けてはいません。 NHTSAは、従来の安全リコールプロセスを通じて「安全に対する不当なリスク」をもたらす自動車や自動車機器に対して強制執行を行う権限を保持しており、あるADS開発者に関して一度だけそれを行ったことがあります。

連邦政府は自動車の設計、構造、性能を規制していますが、州政府は自動車の運行を規制する特定の権限を持っており、これには運転免許、交通違反の取締り、保険、自動車の登録などの問題が含まれます。自動運転システムの安全性に関する連邦政府の規制がないため、多くの州政府は、ガイドライン、州法、行政命令を通じて、自国の道路におけるADSの運用を規制し始めています。National Conference of State Legislaturesは制定された州法と行政措置のデータベースを管理しています。

米国の規制状況先進運転支援システム(ADAS)

ADSと比較して、NHTSAは自動緊急ブレーキ (AEB) 、渋滞回避、ブラインドスポットインターベンションなどのADAS機能についてより具体的なガイドラインを定義し、衝突回避技術（大型車用AEBなど）の規制のためのルール作りを進めています。NHTSA は、これらのADAS機能に対して、特定の初期および終了試験条件、自動運転プログラムが使用すべきシナリオおよび評価基準の説明という形で、規定の試験プロトコルを提供しています。残念ながら、これらのテストプロトコルは、可能な機能テストの不完全なリストに過ぎません。 ADAS機能がODDで遭遇する可能性のあるすべての状況を網羅しているわけではありません。したがって、自動運転プログラムは、 ADASシステムが連邦政府が推奨する性能のベースライン・ レベルを機能的に満たすことができるかどうかを測定するためにのみ、これらのテスト・プロトコルを使用することができます。

自動運転プログラムでは、NHTSAのテストプロトコルを利用して、 ADASの作動中のあらゆる状況に対する安全性を総合的に確保することはできません。同様に、NCAP (New Car Assessment Program) もAEBなどのADAS機能のテストプロトコルを提供しており、現在も進化を続けています。

例えば、2022年3月、NHTSAは新たなADAS機能の評価に関する意見募集を発表しています。しかし、これらのテストプロトコルも、ADAS機能がODDで遭遇しうるすべてのシナリオを列挙するのではなく、単に機能テストの例を示すにとどまっています。

グローバルガイダンス

自動運転システムに関する世界の規制のほとんどはADASに関するものだが、ADSも含まれるものもあります。2021年の夏、ドイツでレベル4 (L4) 自動運転システムに関する法律が成立しました。この法律は、「道路交通法および強制保険法-自動走行に関する法律」を改正したもので、承認され定義された運転領域でのADSの利用を認めるものです。中国、日本、ヨーロッパには、ADSのテストを取り巻く同様の規制や基準があり、人工知能システムのテストに触れているものもあります。

自動車線維持システム (ALKS) に関する国連 (UN) 規則157号は最も注目すべきものです。42カ国が承認しているこの規制は、レベル3 (L3) ALKSシステム適合のための要件と試験方法の一部を規定しています。2022年5月には、メルセデス・ベンツが自動車会社として初めて、消費者が公道でL3システムを操作できるようドイツ政府から認可を受けました。2022年4月、欧州連合 (EU)は、"完全自動運転車 " の型式承認のための統一手順と技術仕様を提案するADS法草案を発表しました。型式承認申請の一部として、ADSメーカーは型式承認当局にADSの安全性を証明する「安全コンセプト」の文書を提出することが求められることになるでしょう。しかし、これらの要件はまだ発効されていません。

既存のグローバル規制の懸念は、米国の規制と同様です。これらの規制は、ハイレベルな要件、意図された動作条件、およびテストシナリオを定義していますが、自動運転システムの安全性を確保するために必要なテストケース、評価基準、および検証方法の包括的なセットを提供するものではありません。特に、自動運転車が遭遇する可能性のあるすべてのシナリオを列挙することは、どの規制機関にとっても不可能であるため、自動運転プログラムは、そのシステムがなぜ安全で、各関連規制に準拠しているのかを厳密に論証する必要があります。したがって、自動運転プログラムでは、安全性に関する正式な論拠を構築するために、Underwriters Laboratories (UL) 4600、Automated Vehicle Safety Consortium (AVSC) の各種ベストプラクティス文書、Society of Automotive Engineers (SAE) J3018、各種国際標準化機構 (ISO) 規格 (例: ISO 21448、ISO 26262) など、規格を分析し参照することにさらなる価値を見出しています。

I.安全性フレームワークのベストプラクティス

明確な規制ガイダンスがないため、自動運転プログラムは困難な状況に置かれています。　自動運転プログラムは、競争力を維持し、高まる消費者の要求に応えるために自律走行システムを迅速に開発する必要があり、同時に開発したシステムの安全性を確保するための枠組みを確立する必要があります。　すべての自動運転プログラムの検証作業において重要な要素は、セーフティケースです。セーフティケースとは、問題の自律化システムが安全であることを正当化するために用いられる、証拠に裏打ちされた構造化された論拠のことを指します。

安全性フレームワークは自動運転プログラムがどのようにセーフティケースを作成し、安全を正当化するかについての基本原則を定義します。これらは自律走行システムのすべての開発者と、さまざまなレベルの自律性に適用されます。厳格なセーフティケースは、プログラムが自律走行システムを商業的に展開しても安全であると規制当局に納得させるのに役立ちます。また、安全が常に最優先事項であることを保証するために、開発全体を通じて自動運転プログラムが活用できる参照ポイントを提供します。

優れた安全性フレームワークは、学術研究、業界標準やベストプラクティス、政府の規制やガイダンスなど、 いくつかの参考文献を基礎として活用します。さらに、システムに関するより多くの情報や他の業界標準やベストプラクティスの出現に伴い、安全性フレームワークも進化していく必要があります。

このハンドブックのパートでは、上位の自動運転プログラムのVSSA、学術研究、標準、規制、および業界のベストプラクティスのレビューに基づいて、すべての自動運転プログラムが安全性フレームワークの一部として採用すべき6つの中核的原則を扱っています（図1）。　これらの基本原則は、すべてのプログラムが取り入れるべき重要なものですが、プログラムによっては追加の原則が必要になる場合もあります。　例えば、自動運転プログラムでは、UL4600などの規格でこれらの基本原則を補完することができます。　UL4600は同様のテーマを扱っており、セーフティケースに含めるべき事項に関するハイレベルなチェックリストを提供しています。

図1：すべての自動運転プログラムが安全性フレームワークの一部として採用すべき基本原則

A.安全性を重視した設計

以下では、自動運転V&Vのための安全指向設計で最も重要な2つの側面に焦点を当てます。 要件駆動型設計、フォールバック、およびサイバーセキュリティです。機械工学的信頼性のような他の側面は、このハンドブックの焦点ではありませんが、全体的な自律走行システム開発の一部として考慮されるべきものでもあります。

要件駆動型設計

要件駆動設計とは、特定の要件に従ってシステム（すなわち、さまざまなソフトウェアおよびハードウェアコンポーネント）を設計および開発するプロセスです。自律走行システム設計の要件は、機能面から安全性、法的ニーズ（連邦法、州法、地方法などの関連法規を考慮）までカバーする必要があります。 要件主導の設計により、チームは努力を集中させ、最終製品でエラーが発生するリスクを低減することができます。 成功する自動運転プログラムでは、できるだけ早い段階で自律走行システムのODDとともに要件の初期バージョンを定義します。そして、開発およびテスト期間中、これらの要件を繰り返し改良していきます。これらの定義、特にシステム要件一式は、自律走行システムまたはそのODDの範囲が拡大するにつれて進化する可能性があります。 しかし、これらの定義は、システムの設計と開発の初期段階において、より厳密であればあるほど良いのです。明確な要件と ODD の定義は、プログラムが明確な範囲を設定し、内部でチームを調整し、開発者が正しい仕様で構築するのを確実にするのに役立ちます。 この実践により、内部の不整合によるエンジニアリング努力の無駄を回避し、プロジェクトの大幅な遅延を防ぐことができます。

フォールバック

通常時にシステムが安全に動作することに重点を置く一方で、システムが安全に動作しない状況も考慮することが重要です。 ADASシステムの場合、システムが安全でない状態を検知した場合、またはシステムが能力またはODDの範囲内にないシナリオを検知した場合がこれにあたります。この場合、システムはドライバーに安全な制御を取り戻すよう通知する必要があります。 ADSの場合、ループ内に人間のドライバーがいない可能性があり、その場合、システムは最小リスク状態への移行、すなわち「フォールバック」 ができなければなりません。

自動運転プログラムは、運転中のフォールバック戦略 （システムが安全に処理できない問題や状況が発生した場合に、最小リスク状態に移行するためのシステムのプロセス）を文書化する必要があります。

サイバーセキュリティ

サイバーセキュリティは、自律走行システムの安全性に大きく寄与する要素であり、サイバー攻撃は車両の運用や試験中に大きな脅威となります。意図する自律性のレベル（例: L2またはL4 ）に関係なく、プログラムは以下の設計上の選択を慮する必要があります。

I. 自律走行システムおよびその基盤となるソフトウェア、ヒューマンマシンインターフェース （HMI） システムは、許可されたユーザーのみがアクセスできるようにする必要がある。
II. 遠隔介入が許可される場合、認証・ 認可された当事者のみが自律走行システムおよび関連する通信ネットワークにアクセスし、影響を与えることができるようにする必要がある。

このハンドブックでは、サイバーセキュリティ関連の問題にはこれ以上踏み込みませんが、これらのトピックは、自律走行システムの開発とテストに関連して考慮し、議論する価値があります。

B.機能安全 (ISO 26262)

自動運転プログラムでは、開発する自律化機能およびソフトウェアとハードウェアのシステム要素ごとに、サブコンポーネントリスク、システムレベルリスク、機能安全性を分析する必要があります。ISO 26262では、システムをサブシステムに分解し、サブシステムがシステム全体にもたらすそれぞれのリスクに関して評価することができます。いわゆるASIL (Automotive Safety Integrity Level) レベルA~Dは、各サブシステムに要求されるリスク評価のための基礎となります。

I. ほとんどの自動運転プログラムでは、故障モード効果解析(FMEA)、フォールトツリー解析 (FTA) 、ハザード解析とリスク評価 (HARA) どの伝統的な安全解析技術が使用されています。FMEAは車両のシステムアーキテクチャに関わる。FTAは、車両や製品レベルの目標をより低いレベルの安全要件に分解する演繹的なアプローチです。HARAは、ISO26262の文脈では、誤動作を調べ、対応するハザードを特定し、そのハザードのリスクを評価することを意味します。
II. 自動運転プログラムでは、複雑なシステムのハザード分

析を行うためにシステム理論プロセス分析（STPA）を使用し、従来の安全分析技術を補完しています。FMEAやFTAと比較して、 STPAはハザードやリスクを引き起こす可能性のある非明示的な部品の相互作用を特定、分析、緩和する上でより効率的です。

FMEA、FTA、STPA がどのように ADAS と ADS V&V に適用されるかの詳細な説明は、 このハンドブックの範囲外です。

C.意図した機能の安全性(ISO 21448)

自動運転開発では、未知の問題をできるだけ早く前もって特定することが重要です。これらの未知の問題の中には前もって予測することが不可能なものもあるため（"unknown unknowns"）、自動運転プログラムでは、自律走行システムの意図する機能の安全性を確保するために、安全性に関する論証を継続的に検証する必要があります。また、プログラムは、検証作業の進捗に応じて、ハザード（シミュレーションと実環境の両方）を継続的に特定し、テストカバレッジを測定するための堅牢なプロセスを備えていることを実証する必要があります。

演繹的アプローチと帰納的アプローチ

トップレベルの自動運転プログラムでは、自律走行システムの安全性を評価するために演繹的アプローチと帰納的アプローチの両方を組み合わせています。演繹的安全アプローチには、ISO21448の文脈でHARAとSTPA手法を使用することが含まれ、これらは機能不全、性能制限、および予見可能な誤用に起因する安全問題により焦点を当てます。プログラムでは、これらの方法を用いて、安全要件やシステム要件から生じる可能性のあるコンフリクトやハザードを列挙することができます。しかし、製品要求から導かれるこれらの最悪ケースシナリオは、単に理論的な見解を提供するだけで、テスト中に発生するハザードで補完する必要があります。これを実現するのが帰納的アプローチです。

帰納的安全アプローチは、特定の観察から問題を特定することに重点を置くという点で、演繹的安全アプローチと異なります。帰納的安全アプローチに従う場合、指定されたトリアージまたはイベントレビューチームを使用して、実世界テスト（例：安全運転テスト中のドライバー介入）およびシミュレーションテストからすべての失敗とハザードを要件まで遡って追跡することが最善です。これには、これらの事象のそれぞれについてHARAを実施することも含まれます。帰納的アプローチは、実世界試験中にODDに関する新しい情報が出てきたときに、自動運転プログラムがその要件を包括的なものにするのにも役立ちます。

ODDデータベース

演繹的安全アプローチと帰納的安全アプローチを組み合わせることに加え、自動運転プログラムでは、テスト中に自律走行システムのODDで遭遇したシナリオやオブジェクトの内部データベースを管理することができます。 これは、プログラムがシステムが安全に扱えるシナリオとオブジェクトを理解するのに役立ち、運転中に新しいオブジェクトが発生したときにそれを特定することができます。 この内部データベースは、外部データベース（例えば、衝突や自然主義的な運転データベース）で補完することができます。車両データの収集は高価で時間がかかるため、外部データベースは貴重です。 このように、第三者のデータソースは、ODD情報とエッジケースの中立的な補足ソースとなることができます。これらのデータベースを維持し、シナリオ内のオブジェクトと相互作用に従ってテストを分類することにより、自動運転プログラムは、ODDカテゴリ全体のテストカバレッジを測定し、ODD外のイベントを検出し、以前にカバーされていなかった新しいシナリオとオブジェクトを捕捉することができます。

D.エビデンスに基づく安全性

機能安全や意図された機能の安全性 (SOTIF) に加えて、自動運転プログラムが安全性フレームワークで（そして最終的にはセーフティケースで） 示すべきもう一つの安全プロセスは、証拠に基づく安全性または「実証された」安全性です。証拠に基づく安全性は、定量的な統計分析を通じて安全性を実証することに重点を置いています。 成功した自動運転開発チームの多くは、以下のような方法を用いて証拠に基づく安全性を実証しています。

指標と評価

I. システム性能の評価、特に要件に対する評価には自動化されたルールを活用します。例えば、最大車線逸脱や最小横方向安全バッファーに関する許容基準を適用することができます。
II. 衝突の発生頻度は非常に低いため、衝突に先行する、あるいは衝突を予測する事象を追跡することで、代替的な安全性指標として機能することができます。衝突までの時間 (TTC) 、減速度 (DR) 、侵入後時間 (PET) などの代替安全指標を、他の高度な指標と並行して使用

します。

III. 自律走行システムの行動を観察された人間の行動（例えば、操縦の実行時間や反応時間） と比較する指標を定義、使用します。これらの測定基準により、プログラムは "自律走行システムは人間のドライバーのように車線を変更するか？"のような高度な質問に答えることができます。

IV. シミュレーション、構造化されたテストトラック試験、実環境試験、実運用から得られるデータを用いて、上記の指標を検証し、更新します。

V. プログラムのすべての安全関連領域について、安全性能指標(SPI) を定義、分析、そして監視します。SPIは自律走行システムの安全性のある側面を測定する指標であり、特定の安全性の主張を評価するための閾値を含みます。

VI. 上記の指標を定期的に（日報などで）モニターし、安全な開発・テスト・運用に役立てます。

統計範囲

I. シナリオの分類、予想されるパラメータの範囲と寸法（先行車速度、車線変更時間、静止障害物の大きさなど）を列挙した正式なODD定義を作成します。システムの能力が向上し、より多くのデータが収集されるにつれて、ODDの定義も拡張されます。

II. フルカバレッジに必要なODDスコープ全体で、考えられるすべてのシナリオのバリエーションをテストするために、規模に応じたシミュレーションを使用します。

III. カバレッジとパフォーマンスメトリクスを組み合わせ、ODDの十分な領域で許容できる システムの安全性を統計的に証明します。

敵対的テスト

I. 独立した検証および妥当性確認 (IV&V) テストを活用して、一連の敵対的テストを定義および実行し、システム性能の境界を調べます。 IV&Vテストは、自律走行システムの設計方法に関する先験的な情報がないため、何をテストすべきかという客観的な第三者の視点を提供します。

II. シミュレーションとテストトラックでストレステストを実施し、基盤となる自律走行システムの限界の証拠を提供し、理解します。

再現性

I. 特に車両の制御とダイナミクスのモデリングにおいて、シミュレーション結果が実世界の試験で再現可能であることを実証するための実験を実施します。これらの実験結果を、徹底的かつ成功したテストの証拠として規制当局に提供します。

II. これらの実験には、テストコースでのシナリオの実行、（車両テレメトリまたは外部測定装置からの） 姿勢データの記録、検証用主要性能指標 (KPI) の比較、および対応する再シミュレーション （すなわち、その状況下で車両がどう振る舞ったかを決定論的に再現する） との定性的相関の実行が含まれます。

E.安全に関するガバナンスと文化

業界のベストプラクティスに基づき、自動運転プログラムでは、安全ガバナンスと文化について以下の2つのアプローチを取ることができます。

I. 部門横断的なチームを定期的に結成し、実環境試験、リスク管理、製品またはソフトウェアの導入決定など、主要なソースに起因する安全性の問題を分析します。

II. エンジニアリング、安全、製品の各チームからリーダーや専門家を集め、 会社の安全性フレームワークが最新で組織全体が遵守していることを確認すします。 正式な安全委員会またはガバナンス構造を確立します。

F.安全な運用

自動運転プログラムは、その自律走行システムの安全な運用を保証するために、いくつかの方策に従うことができます。

I. 安全運転者の厳格な採用、トレーニング、継続的な業績評価プログラムを活用し、業務上の安全性の基準バーと車両の操作およびテストプロセスの一貫性を確保します。

II. 安全な試験方法についてドライバーを指導 ・ 訓練するため、試験車両にドライバーモニタリングシステムを使用します。

III. 顕著なHMIを使用して、ドライバーに車両を引き継ぐタイミングを明確に指示し（例: ADAS動作中またはADSテスト中）、システムの現在の状態を伝えます（例: 現在自律走行システムが作動しているか、 システムに異常が発生しているか）。

IV. ディスエンゲージ解析 (ADS試験中にセーフティドライバーが自律モードを解除する状況を解析すること) と再シミュレーションに関するプロセスを自動化します。

V. クラッシュを診断し、そこから学ぶことができるように、データロギングシステムに冗長性を持たせます。

VI. 事故に迅速に対処し、さらなる分析のために事故の詳

細を適切に記録するために、明確に定義された事故後のオペレーションを実行します。

VII. システムの冗長性を確保することで、予期せぬハードウェアやソフトウェアのエラーや不具合が発生した場合の運用の安全性を確保します。

上記のこのセクションの6つの原則を組み合わせると、堅牢な安全性フレームワークの構築に向けた最初のステップが確立されます。自動運転プログラムでは、安全性フレームワークを構築し、自律走行システムの配備を検討する際に、適切な要件の決定、ハザードと欠陥の評価、 ODDの未知数の予測、安全エビデンスの定量化、安全文化の施行、安全運転のサポートという課題を解決する必要があります。

II.V&Vの
ベストプラクティス

安全性フレームワークを定義することは、自動運転プログラムが安全について考え、安全を確かなものにする方法を定義するための重要な第一歩となります。 このことは、次のような問いかけにつながります。「どのようにすれば自動運転プログラムが実際に安全性フレームワークに準拠し、安全な製品を提供するためのロバストなV&Vプロセスを構築することができるでしょうか?」

このハンドブックの前編（I.安全性フレームワークのベストプラクティス） で概説した6つの安全性フレームワークの原則に基づき、次のパートでは、自動運転プログラムが安全性フレームワークの原則を遵守するのに役立つロバストなV&Vプロセスを実装するためのステップとベストプラクティスを順を追って説明します。まず、Vモデルとアジャイル開発のバランス、リリースバリデーションの管理、安全を重視した組織の確立など、運用のニュアンスについて説明します。 成熟した自動運転プログラムでは「ロバストな」V&Vが極めて重要ですが、自動運転システムの開発初期段階で大規模な開発が行われている間は、包括的なアプローチの導入を正当化することは難しいかもしれません。Applied Intuitionは、開発の進捗に合わせてV&Vを成長させ、継続的なテストと評価を活用して開発を導く漸進的戦略を推奨しています。次のセクション (A. V&Vライフサイクル) では、V&Vワークフローを構成する主要なプロセス （要件定義、 大規模テストの作成、どのテストをいつ実行するかの決定、テストパフォーマンスの計測など） を紹介します。 また、このセクションでは、チームが活用すべきテストの種類 （シミュレーションベースのテストが有効な条件と、実世界での運転が必要な条件） についても説明します。

A. V&Vライフサイクル

自動運転プログラムは、テストエンジニアから開発者、セーフティドライバーに至るまで、組織全体に渡ってそれぞれが貢献する厳格なV&Vプロセスを設定することを目指すべきです。厳格なV&Vプロセスは、製品のリコールやネガティブな世評を防ぐことができ、チームが安全で検証された、成功する製品を開発するのに役立ちます。安全は決して一過性のものではないため、チームはシステムの設計、開発、デプ

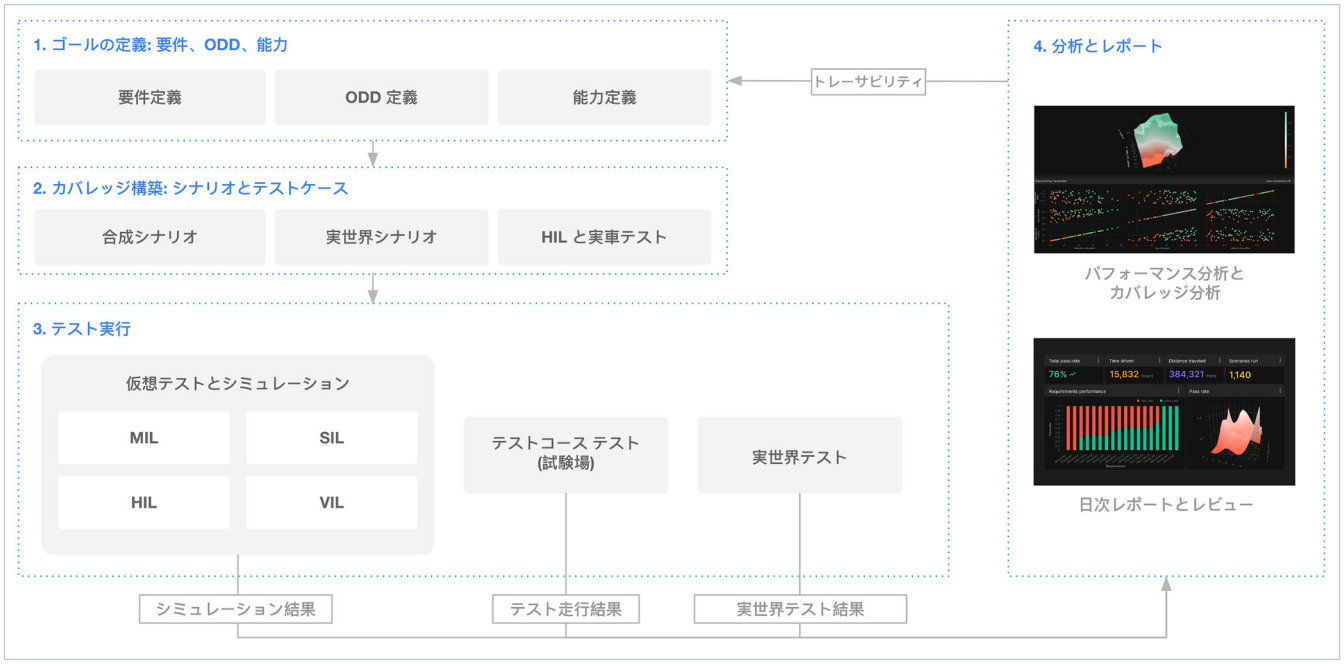


図2:V&Vライフサイクルは、4つの明確なステップからなる連続的なプロセスです

大まかに言えば、自動運転プログラムは次のようにあるべきです。

I. まず、要件とODD(図2の「1.目標を定義する」を参照)を定義し、見直すことから始めます。多くの場合、この作業の責任は、システムエンジニアリングおよびテストエンジニアリングチームにあります。 これらのチームは、自動運転システムの要件を定義するために、 製品の機能、ユーザーの期待、法規制、および安全性に関する文献を検討します。
II. そして、定義された要件に基づき、自動運転システムを評価するための適切なテストケースを設計する必要があります(「2.ビルドカバレッジ」参照)。
III. 次のステップは、これらのテストケース (テストライブラリ) を開発期間中はもちろん、リリース候補を評価する際にも継続的かつ徹底的に実行することです(「3.テストの実行」を参照)。ほとんどの場合、テストは低コストの環境から高コストの環境へと進んでいきます。すべてのテスト環境は開発期間中に使用されますが、シミュレーションから始まり、テストコースや実世界でのテストに移行することも可能です。
IV. すべてのテストを通じて、テスターと開発者は結果を分析します(「4.分析と報告」を参照)。性能に関する結果は、システムの改良の指針となり、次に何を修正すべきか、また、要件の調整が必要かどうかをチームが判断するのに役立ちます。性能に関する結果だけでなく、 カバレッジ分析により、自動運転プログラムが新たにどのようなテストを作成し、実施する必要があるかを判断することができます。

B. V&Vのステージ

それぞれの自動運転プログラムは、開発と検証に関して異なるステージにいる可能性があります。 次の表は、さまざまなV&Vの次元において、初期、中期、後期の自動運転プログラムがどのようなものかを概説したものです(図3)。 「初期」 段階とは、 V&Vの旅を始めたばかりのチームを指します。「中期」段階とは， V&V プロセスの一部を既に確立しているが，商用展開までおよそ2年以上あるチームを指し、「後期」段階は、今後1~2年以内に商業展開を考えているチームを指しています。

次の表は、自動運転の状況の現状を示すスナップショットです。 特定の自動運転プログラムが現在どの段階にあり、 業界の他のプログラムとどのように比べられるかを評価するための簡単な参考資料として利用できます。 しかし、この表は、 3つの段階のいずれにおいても、推奨されうる網羅的な指標であるとも、あるいは自動運転プログラムの目標の定義としても機能するものではありません。 この表は、このハンドブックの他の部分と合わせて、自動運転システムの安全かつ迅速な商業的展開を達成するためにプログラムが現実的に活用できる検証手法に関するガイドとしてお使いいただけます。

	初期	中期	後期
チーム	機能開発者（プランナー、認識など）がシナリオ作成やテストも行うチーム	シナリオ作成とテストを担当するテストエンジニアリング・チームが拡大	システム、テスト、バリデーションエンジニアリング、シミュレーションオペレーション、トリアージのサブチームを持つ成熟したバリデーション組織
安全性に関するガバナンス体制	正式な組織体制がない	シニアメンバーが正式に安全性を担当	エンジニアリング、安全、製品の各チームからリーダーや専門家を集め、会社の安全性フレームワークが最新で正しく実施されていることを確認する、正式な安全性に関する委員会またはガバナンス構造が整っている
セーフティケース	まだ存在しないが、ターゲットODDとエンドユースケースは定義されている	安全性証明のための定義づけと根拠の収集を開始	定義を確定し、十分なエビデンスを収集、安全性フレームワークを公開
リリース検証・承認プロセス	正式なプロセスがない	新しいソフトウェアを以前のビルドと比較し、修正すべきリグレッションを見つけるリリースレビュープロセスを取っている	新しいリリースは厳格に検証され、承認またはフリートへのデプロイ前に問題が解決される。マイナーチェンジは自動承認プロセスがある一方、メジャーチェンジは安全性委員会主導でより厳格な承認プロセスが求められる
要求仕様とその管理	最小限の要件が文章化されており、表計算ソフトや文書で管理されている	正式な要件を作成中であり、指定されたアプリケーションライフサイクル管理（ALM）ツールを使用して、要件の作成とテストへのトレースを行い、承認プロセスの確立を開始	ODD の完全な要件一式を、以前から使用されている ALM ツールと統合可能な統一ツールで定義（自動運転検証のための既存の ALM ツールの欠点を補うため）、要件の変更は安全性委員会による厳格な承認プロセスを経て行われる
使用されるテスト方法と環境	MIL (Model-in-the-loop) テストや HIL (Hardware-in-the-loop) テストといった高品質なシミュレータを利用できる場合を除き、クローズドなテストコースや場合によっては実環境でのテストを実施	MIL, SIL (Software-in-the-Loop) シミュレーション、テストコース試験の混合で、HIL、実車両との統合、実環境試験（ドライブログ収集含む）を強化している	MIL、SIL、HIL、車両試験（車両統合、VIL: Vehicle-in-the-loop、テストコース、実環境）を含むすべての試験環境の利用。規模とコストの観点から、試験の大部分をシミュレーションで実施
シミュレーションシナリオライブラリ	シミュレーションのスモークテストの初期セットとクローズドコースでのテスト、現在開発中の機能エリアをカバーするスモークテストの初期セットの拡張に注力	ODDの各シナリオカテゴリーや要件に応じた少数のシナリオ、ファジングやパラメータ化の利用を開始しており、シナリオの作成は機能リリースの頻度や活発な機能開発に合わせて行われる	ODD、要件、シナリオのカテゴリーにまたがる完全なシナリオライブラリ。各シナリオは、徹底したパラメータカバレッジのために多くの具体的なバリエーションを生成する。（実世界のドライブやランダム化手法による）エッジケースやロングテールシナリオ、次のODD用の新しいシナリオセットの発見・作成に注力

テストコースや公道でのテストの役割	チームがシミュレーション・テストに移行するまでの間、ADASのテストのほとんどは、バグや問題を発見するためにクローズドなテストコースで実施される	シミュレーションの忠実度を確認するためのテストコースでのテストと、新しいシナリオのためのインスピレーションを得るための小規模な公道でのテストの利用	大規模な公道車両による継続的な新バージョンテストと再シミュレーションのためのエッジケースデータの収集、シミュレーションの忠実度を検証するためのテストコースの利用
カバレッジ分析	カバレッジはほとんどトラックされていないか、シナリオのカテゴリーごとに分割された全体的なテスト数で簡単に測定されている	カバレッジはシナリオのカテゴリーごとに分割したテスト数で測定され、要件カバレッジ、シナリオパラメーター空間カバレッジ、（ターゲットODDの限られた領域に基づく）マップカバレッジ、およびODDカバレッジ	厳密な要件カバレッジ、シナリオパラメータ空間カバレッジ、（ターゲットODDの全領域に基づく）マップカバレッジ、およびシナリオカテゴリ毎の統計的ODDカバレッジ
パフォーマンス分析	スモークテストで発生したすべての失敗を修正（重要度により優先順位が決まる）、初期オブザーバー（結果が合格か不合格かを判断するルール）はまだ開発・調整中（安定性の追跡が必要）、KPIとSPIを毎週確認している	開発の反復モデルを反映し、リグレッションを修正するための正式なA/Bテストの強化、オブザーバーの初期セットとアドバンストなオブザーバーの開発および調整（安定性の追跡が必要）、KPIおよびSPIを毎日確認している	SPIとKPI間のトレードオフの評価を含む、前回のリリースからの定量的な改善に関する大規模なA/Bテストと統計分析、KPIとSPIの日々の追跡調査
パフォーマンスの目安	合格率 0 - 65%	合格率 50 - 90%	合格率 90%以上

図3：成熟段階に応じた自動運転プログラムのV&Vの典型的な状態

自動運転プログラムが安全かつ迅速な商業展開のための最適なポジションに到達できるよう、ハンドブックの以下のセクションでは、チームが正しいV&Vプロセスを実施し、V&Vライフサイクルの4つのステップをプログラムに組み込むための規範的なステップとベストプラクティスを紹介します。

C.プロセスと人材

V&Vの詳細に入る前に、このセクションでは、プログラム成功の基礎となるプロセスと人材について説明します。 最初のセクションでは自動運転プログラムが製品開発プロセスをどのように管理できるか、自動車産業からの過去のモデルをどこまで適用できるのか、そしてそれらのモデルを自動運転開発に具体的にどのように適応させるかについて概説します（「製品開発プロセスの管理」参照）。次のセクションでは、 自動運転システム開発における課題の多くはソフトウェア側に存在するため、 チームがソフトウェアのリリース検証プロセスを設定し、それを量産化して、迅速でありながら充分な開発およびテストサイクルを確保する方法についても説明します（「リリース検証プロセスの量産対応」参照）。最後のセクションでは、V&Vの取り組みを主導する責任を負うべき立場について説明します。組織内では、安全は全員で責任を持って取り組むべきです。しかし、確固な安全文化を確立し、維持するためには、任命された安全指導者とガバナンスが重要です（「安全ガバナンス委員会の設立」参照）。

製品開発プロセスの管理

自動運転の開発・検証チームは、（自動車業界で一般的な）伝統的なVモデルと（ソフトウェア開発で一般的な）アジャイル手法の両方を組み合わせた手法を使用することが推奨されます
（図4）。

Vモデルは、自動車メーカーが数十年にわたり伝統的に使用してきたトップダウン方式です。この手法では、まず要件を定義し、次にシステムの一部を設計、 実装、 テストし、 その後、システムの他の部分すべてについて同じプロセスを繰り返すことが規定されています。このアプローチでは、検証チームは、システム、最終製品、あるいは安全性や法的要素など、 いずれの要件に対しても、前もって定義するために労力を費やす必要がありますが、 正しく遂行することにより、 安全性と設計プロセス全体に関する要件が明確に定義されます。 これにより、誤りや最適でないシステムおよび製品設計が発生しにくくなり、チームの時間とコストの節約につながります。しかし、Vモデルアプローチの欠点は、ODDの完全な複雑性をあらかじめ理解していることが前提であることで

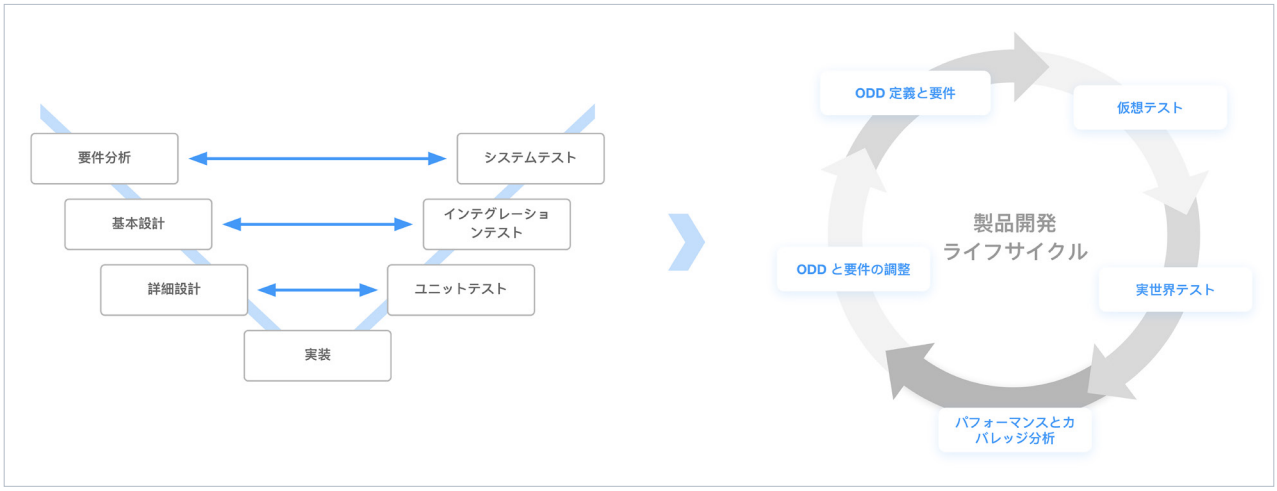


図4：自動運転開発では、自動車業界で一般的かつ伝統的なVモデル（左）とアジャイル手法を組み合わせた、反復的な製品開発ライフサイクルを形成することが推奨されます。（右）

す。Vモデルアプローチに従う場合、チームはシステムの実装を開始する前に、 すべての要件を包括的に記述する必要があります。

自動運転プログラムは、システムの開発、テスト、および現場での運用の中で、 ODDとその複雑性に関する新しい情報を知る可能性があるため、 開発全体、 さらには配備後も要件の更新が可能である必要があります。そのため開発者は、これを達成するために、伝統的なVモデルとアジャイルアプローチを組み合わせる必要があります。アジャイルアプローチを取ることで、自動運転プログラムは、限定的に開始し、そこから反復的に成熟したシステム設計を構築することができます。 アジャイルアプローチでは、まずは製品の初期バージョンを定義して完成させ、この製品を成熟に向けて反復的に改良していくことを目指します。 また、アジャイルアプローチは、初期の動作可能なソリューションをより早く配備してテストできるため、システム設計に関する重要な情報、最適なテスト方法、さらにはチームが検証すべき新しい要件などを提供するのに役立ちます。
自動運転プログラムは、以下のプロセスを設定し、それをリリースごとに反復することで、 ここで提案するハイブリッドアプローチに従った自動運転システム開発を効率的に行うことができます。

I. 最初にプロセスを開始する場合、 新しい機能を構築する場合、または新しい領域に拡張する場合、最初の要件セットとODDを定義します。 既存の製品や機能に対してプロセスを反復している場合、必要に応じて既存の要件とODDの定義を更新します（例： 実走行テストで、電動スクーターなどの新しい車種がODDに含まれることが判明）。
II. 新たに構築または更新する機能のテストを設計します。
III. 自動運転機能のための新規または更新される機能の開発を行います。
IV. 開発中も、システムに対して一連の関連するテストを実行します。 継続的インテグレーション(CI)テストを活用します。
V. ソフトウェアリリース候補を完成させ、テストセット全体で包括的にテストします。
VI. 結果を分析して問題を特定し、既存のシステム機能の更新、または新しい機能の構築によってその問題を解決します。

反復サイクルを最小化し、 ハイブリッドアプローチをより効率的に行うために、自動運転チームはテストリソースを効果的に配備することを意識する必要があります。従来の自動車システムは主に物理的に現実世界でテストされることが多くありましたが、仮想シミュレーションの登場により、これまでの様な数日やそれ以上を待つことなく、数分でテスト結果を得ることができるようになりました。一方で、自動運転システムも最終的には物理的なシステムであるため、実環境でのテストは避けて通れません。 さまざまなテスト環境と、それぞれを活用するタイミングについては、このハンドブックの「各テスト環境の有効活用」 のセクションでさらに詳しく説明します。

Vモデルとアジャイル手法のハイブリッドアプローチを導入することで、自動運転チームはそれぞれの利点を組み合わせて、注意深く開発された最終システムを迅速、効率的、かつ安全に開発することができるのです。

リリース検証プロセスの量産対応

自動運転チームは、リリースのための開発を完了したあと、配備を承認する前に、出来上がったソフトウェアリリース候補を厳格にテストし評価する必要があります。 チームは、安全なソフトウェアの配備を保証しつつ、開発全体の速度も維持するために、正式なリリース検証プロセスを遵守する必要があります。開発および検証の速度は、市場投入までの時間を最適化するための鍵です。したがって、検証および承認プロセスは継続的、自動的、かつ可能な限り仮想環境で行われるべきです。自動運転プログラムで以下の図を、新しいソフトウェアリリース候補ごとの、シナリオベースでのテスト、 評価、および承認を組み込んだリリース検証ワークフローとして、検討することができます（図5）。

この図にある通り、自動運転プログラムは、事前に定義された基準を満たしている場合、ソフトウェアリリースを自動的に承認することができます。 そうでない場合は、チームが不具合のトリアージを迅速に行う必要があり、一方で安全委員会や中心となる責任者がリリースを条件付きで承認することも可能です。 このプロセスが厳密であればあるほど、開発と検証のスピードは上がります。プログラムは、リリース検証プロセスを量産化する際に、以下のキーポイントとステップを考慮することができます。

I. シナリオベースのテストスイートを定義し （シナリオ作成参照） 、その上で各新規ソフトウェアリリース候補をテストします。 テストはシステムの幅広い機能をカバーする必要がありますが、特定のリリースに合わせて調整することも可能となります。 たとえば、車線変更機能の改善に重点を置いたリリースの場合、チームは車線変更に関するシナリオをより多くテストする必要がありますが、その場合でも、意図しないリグレッションを排除するために、他のすべての機能に対するテストを実行する必要があります。 シミュレーションによるテストは、実環境や閉鎖されたテストトラック環境ではセットアップや遭遇が困難な状況をテストできるため、実環境でのテストに比べて時間とコストの削減につながります。

II. 自動的にリリースが承認されるために満たすべき基準を定義します。 この基準は、前回のリリースから定量的な改善見られていること、重大な問題が検出されないこと、すべての主要要件と機能が許容レベルまで満たされていることを中心に設定することができます。 マイナーリリースの場合、リグレッションや致命的な不具合がない限り、厳格なレビュープロセスを回避して時間を節約することもできます。

III. リリースのパフォーマンスを評価します。 大規模なリリースの場合、これにはパフォーマンス結果の徹底的な分析が含まれます。 一部の自動運転プログラムでは、正式なトリアージチームやソフトウェア品質管理チームが、障害のトリアージ、優先順位付け、実際には不具合でないものの除外、レビュー用のリリースレポートの作成などを専門に行っている場合があります。

IV. リリースレポートでは、 どの様な既知で重大な問題が存在するかを網羅し、能力ごとの全体的なパフォーマンスを評価する必要があります（「パフォーマンスの分析」参照）。リリースレポートが作成されたら、安全チームはそれをレビューし、能力ごとのパフォーマンスと既知の問題のリストを考慮して、そのリリースが安全で、 条件

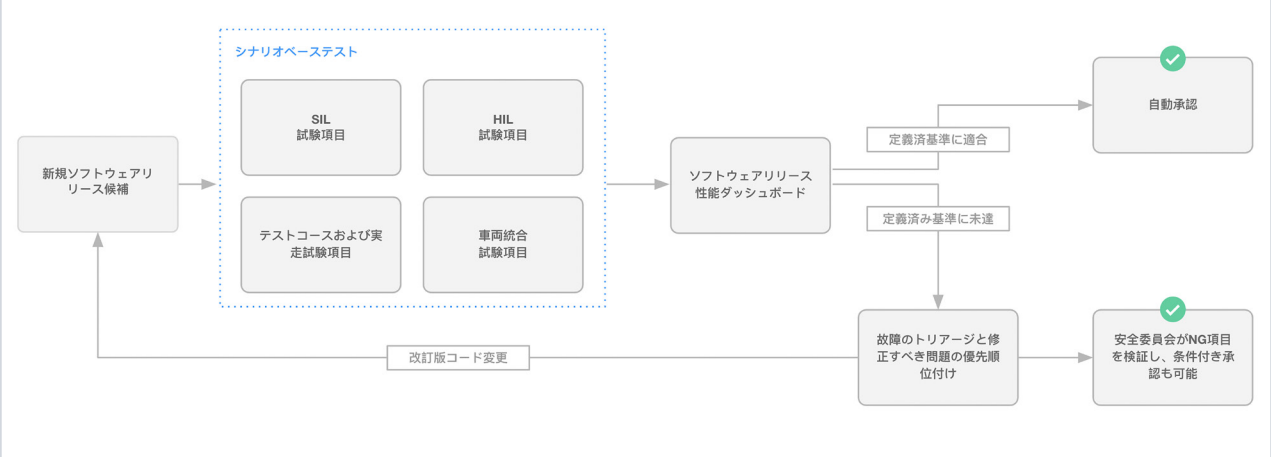


図 5: 自動運転プログラムは、正式なテスト、分析、レビュー、承認プロセスを用いて、各ソフトウェアリリース候補の妥当性を確認する必要があります。

付き、 または最終的な承認を与えられるかを決定する必要があります。

V. 安全チームがリリースを承認すると、リリース担当者は変更を配備することができます。 開発チームは、優先順位付けとトリアージが行われた問題リストを含むリリースレポートを受け取り、次のリリースで修正または対処を行う必要があります。

安全ガバナンス委員会の設置

安全ガバナンス委員会は、確立された安全基準とベストプラクティスが組織全体で順守されることを保証するために極めて重要となります。 ほとんどの自動運転プログラムでは、開発および検証プロセスの後半になるまで、正式な安全ガバナンス機構を制定していません（図6）。その代わりに、安全に関するテーマを担当する個人または小規模なチームが存在する場合があります。分散型のアプローチは、初期および中期段階のプログラムには有効ですが、試験が増えるにつれて規模を拡大することが出来なくなります。確固な安全文化を確保するためには、できるだけ早い時期に正式な安全ガバナンス委員会を設立することが望ましいです。

自動運転プログラムは、 以下のステップを踏んで、 正式な安全ガバナンス委員会を設立することができます。

I. プログラムのエンジニアリング、安全、製品の各チームから主要なステークホルダーを選び、安全委員会を構成します。

II. 定期的なミーティング（月1回を推奨）を設け、プログラムの現在の安全性フレームワークを見直し、それが遵守されているかどうかをチェックします。

    a. 現在および将来のテストまたは運用時に向けて、現在の安全レベルの調査および見直しを推進します。

    b. 一般に公開される外部向けのものも含めたセーフティケースの起案と管理を行います。

    c. システム要件とODDの定義を確認します。

    d. 社内外から報告された安全事故を検証する場を提供します。

D.要求管理 ・ トレーサビリティ

要件定義とODD

自動運転プログラムは通常、開発の初期段階で可能な限りの要件定義を行うことから始まります。その後、これらの定義を拡張し、開発の中期段階で指定された要件管理ツールを使用し始めることもあります（図7）。また、開発の早い段階から要件やテストケースの管理ツールを使用することで、先手を打つことも可能です。

自動運転プログラムは、要件とODDの定義を円滑かつ正確に作成するために、以下の推奨事項を考慮することができます。

I. 開発前と開発中に、アジャイルな方法で要件を定義する（「製品開発プロセスの管理」を参照）。これにより、チームは、自分たちが知っている領域を適切に確認することと、さらにデータを収集する必要がある領域を反復的に改良することのバランスをとることができます。 システムを全面的に展開する前に、組織は、各サブシステムとシステムの能力を含むシステムのすべての側面について要件を列挙し、システムの能力が拡大したときに新しい要件を反復するためのプロセスを持つ必要があります。

II. 機能の開発者から独立し、 適切な利害関係者の意見を取り入れ、要件定義を担当するチームを任命します。 これにより、必要なシステム能力を客観的に定義すべき要件作成者と開発者間の利害の衝突を回避することができます。 これは通常、システムエンジニアリングチームの役割です。

ステージ	説明
初期	なし
中期	シニア社員が正式に安全担当となる
後期	正式な安全委員会またはガバナンス機構により、エンジニアリング、安全、製品の各チームからリーダーや専門家を集め、会社の安全性フレームワークが最新で遵守されていることを確認する

図6:V&Vの段階ごとの代表的な安全ガバナンス機構

III. 要求を満たすか否かの評価基準を、文献や実験結果によって正当化し、明確に定義します。評価基準が明確でないと、システムが各要件を十分に満たしているかどうかが不明確になります。 例えば、具体的な要件として、「自律走行システムは、自転車と安全な距離を保つ必要がある」といったものがあります。この場合、評価基準には、 自動運転システムとその場の他のアクターの両方にとって安全と考えられる正確な距離または距離の範囲 （例えば、自律走行システムの速度および/または天候条件に依存する条件付き距離） が含まれることになります。この評価基準を選択する場合、チームはこの距離または距離の範囲が安全と考えられる理由を正当化する必要があります。

IV. ODDは、自律走行システムが安全に機能すべき風景、環境条件、動的要素に関する詳細なデータ駆動型の定義として定義します。これには、正確な値と潜在的な範囲（例えば、正確な時間帯、気温、地図上のゾーンなど） を持つすべての列挙された属性が含まれます。 さらに、カバレッジ分析のために、各テストをODD属性に従ってタグ付けできるように、 正式な分類法でODDを定義する。例えば、ある自動運転プログラムでは、環境条件として「天候」を指定し、属性として「晴れ」「雨」「霧」「雪」「あられ」「煙」を挙げることができます。雨の属性には、 降水強度のパラメータ範囲を列挙することもできます。

要求事項の管理

定義された後、チームはすべての要件を適切に管理 ・ 維持し、要件が最新で正しく使用されていることを確認する必要があります。自動運転プログラムは、以下の推奨事項を考慮する必要があります。

I. 機能の開発者が、すべての要件を表示できるようにします（直接割り当てられた要件や自分の作業領域内の要件だけでなく）。要件は、エンジニアリング作業が予期しない要件に影響を与える可能性があるため、すべての開発者が洞察する必要があるコンテキストを提供します。

II. 組織全体および各サブシステムからの要件を一元的に管理します。これにより、ユーザーは実装の変更が個々の要件に及ぼす影響を評価し、要件の性能間のトレードオフを測定できます。

III. バージョン管理されたシステムで要件を管理し、アクセスを制限します。 要件が進化しても、過去のバージョンを維持します。要件はセーフティケースの中核をなすものであるため、プログラムでは、編集権限を適切なコンテキストを持つチームメンバーに制限する必要があります。チームメンバーが編集を行う前に、プログラムは、安全ガバナンスプロセスに基づく正式なレビュープロセスを実施する必要があります。 レビュープロセスには、なぜその編集が必要なのかの明確な証拠が含まれていなければなりません。

IV. 要件に対するすべての変更を追跡します。 将来の監査では、異なる時点でシステムが処理することが期待されていた要件のスナップショットを提供するよう、 自動運転プログラムが要求される可能性があります。 これは、自動運転システムが継続的に導入され、 改善されていく中で、 将来的に事故が発生した場合に不可欠なものです。

要求のトレーサビリティを設定する

要件のトレーサビリティは、自動運転プログラムの全体的な目標と最終製品との間にリンクを確立します。 これは、実装プロセス、各テスト、および各テスト結果を包含し、チームがあらゆる変更の下流および上流への影響を理解するのに役立ちます。 自動運転プログラムでは、 要件トレーサビリティを成功させるために、 以下の2つのステップを考慮する必要があります。

I. SIL、HIL、車両環境にわたるすべてのテストケースに要件をリンクさせ、安全性を徹底的に実証し、システム要件を検証します。 これにより、すべてのテストが検証のために適切に利用され、テスト結果から要件へのトレーサビリティが確保されるようになります。

II. 要件、テストケース、シナリオ、テスト結果、問題、実装の間で双方向のトレーサビリティを可能にする要件管理ツールを使用する。 これにより、チームは新しいテスト、ソフトウェアの更新、要件の更新の影響を理解することができます。 システム実装とのトレーサビリティにより、ユーザーは要件と実装の関連性を確認し、コードが異なる要件に対してどこをテストしているかを示すことができます。これにより、チームは明確な目標を設定し、監査人や規制当局に対して完全な透明性を確保することができます。 未解決の問題を解決済みの問題にリンクさせることで、 チームは要件検証の現在のブロッカーを確認でき、 後で同様の問題やリグレッションが発生した場合に備えて過去の問題の履歴を確認することができます。

ステージ	説明
初期	最小限の要件がスプレッドシートやドキュメントに書き出され、追跡されている段階
中期	ALMツールを使って、正式な要件を構築している段階
後期	ODDの完全な要件セット。これまで使用されてきたALMツールと統合された統一ツールが必要となる段階（自律性検証のための既存のALMツールの欠点を補うため）。

図7:V&Vの段階による典型的な要求管理プロセス

E.シナリオ作成

次の表は、自動運転プログラムの開発段階によって、シナリオ作成にどのようにアプローチするのが一般的かを示しています（図8）。初期段階のチームは通常、要件とシナリオのカテゴリーにまたがる広範な範囲を構築することに重点を置いています。 強力で広範なカバレッジを構築した後、後期のチームはエッジケースシナリオの収集と生成、および新しいドメインへの拡張に重点を置きます。

総合シナリオライブラリーの作成

自動運転プログラムでは、 意図した展開のためのODD全体をカバーする包括的なシナリオ ・ ライブラリを構築する必要があります。 このライブラリを使用することで、 チームは、ODDで起こりうる一連のシナリオに対して、 主要なパフォーマンスと安全性のベンチマークに対して自律走行システムをテストすることができます。

SOTIF (Safety of the intended Functionality) によれば、自動運転プログラムの目標は、自動運転システムが実世界で、1) 安全に処理できない、2) 見たことがなく安全に処理できない可能性があるものに遭遇するリスクを最小にすることです。 自動運転プログラムでは、包括的なシナリオライブラリを構築するために、2つのアプローチを用いることができます。

- **演繹的アプローチ：** システム要件と製品要件から、シナリオと潜在的なハザードを特定します。テストエンジニアとシステムエンジニアが協力して、各システム要件について何が問題になるかを予測します。これにより、ハザードのリストを作成し、チームはシナリオを作成してテストする必要があります。演繹的アプローチは、ハザードを決定し、自律走行システムが安全に処理するために必要なシナリオを構築するための第一原理的な方法です。しかし、演繹的アプローチはどちらかというと理論的であり、実際のテストによる観察と組み合わせる必要があります。したがって、帰納的アプローチも必要です。
- **帰納的アプローチ：** 実際の試験や運転から観察されたハザードや故障のシナリオを特定します。これらのハザードは通常、 1) 実世界のテストや運転中に観察されたドライバーの介入、 離脱、 または交代イベント、 2) 実世界のテストや運転中に観察された事故または事故寸前、 3)ドライブログの再シミュレーションやファジングからもたらされます。セーフティドライバーは、自律走行システムがある事象を安全に処理できたと確信できないときはいつでも、トリアージチームによるさらなる調査のためにその事象をラベル付けする能力を持つべきです。また、チームは、これらの離脱やハザードを要件まで遡り、新しいデータに基づいて継続的に定義を改良する必要があります。

自動運転プログラムは、以下のテクニックを活用し、演繹的・帰納的に包括的なシナリオライブラリを構築することができます。

I. **通常の合成シナリオ：** まず、合成シミュレーションシナリオで、各要件と能力の幅広いシナリオカバレッジを構築します（演繹的アプローチ）。チームは、テストが必要なコアシーンと属性を定義する必要があります。次に、各属性をパラメータ化し、パラメータ値の各組み合わせのクロス積トがテスト可能であることを確認する必要があります。システムの属性（意図する操作、アクタータイプ、アクター操作、道路障害物、天候、道路インフラ、場所など）は、シナリオライブラリでパラメータ化し、テストする必要があります。理想的には、これらの合成シナリオは、チームが ODD のマップ間でスケールアップできるように、マップに依存しないものであるべきです。
II. **IV&V シナリオライブラリ：** 演繹的アプローチは、内部チームがどのように要件をテストするか（および要件自体） に大きく依存するため、自動運転プログラムは、内

部のシナリオ作成作業を外部のシナリオ作成作業で補完する必要があります。IV&Vチームは、内部チームのテストとは別に作成された一連の敵対的テストを提供することができます。 これらの独立したテストは、システムの機能テストとストレステストに役立ちます。外部チームは、システムに対する偏見や予備知識を持っていません。外部チームは、独自のベストプラクティスに基づいて機能テストを作成し、不具合を発見し、内部チームが見逃している可能性のある状況を考慮することができます。

III. **実世界での走行データ：** 自動運転プログラムがテスト車両を開発する際、ODDから代表的なデータを収集するために、実世界で自律的または手動的に運転させることができます。チームは、これらのドライブからのログをすべて記録するか、最も興味深い断片（通常、ハザードまたは離脱とみなされるもの） をシナリオライブラリの一部として保存する必要があります。 チームは、これらのドライブログを新しい合成シナリオテストのインスピレーションとして再利用したり、収集したドライブログの再生とファジングを可能にするシミュレーションフレームワークを採用して、より忠実なテストを実現することができます。

IV. **エッジケースシミュレーションシナリオ：** 実環境でのテストが進むにつれ、収集されたデータから、カバーすべき新たなシナリオがひらめくでしょう。 自動化プログラムは、これらのエッジケースに対応する新しい合成シナリオを構築する必要があります。 合成シナリオは、センサーシミュレーション用の合成データを含め、実世界ではほとんど発生しない状況や、実世界で収集するのが危険な状況を想定して作成することができます。 例えば、子供や交通弱者、稀な気象条件、極端な行動や危険な行動を含むシナリオを、合成シミュレーションで作成することができます。

自動運転プログラムでは、合成シナリオの作成や収集した車両データからのイベントのレビューに運用チームを充てることで、シナリオライブラリーを構築することがあります。 しかし、 このプロセスを管理するための運用負担は、特にリソースに制約のある初期段階のチームにとっては大きな負担となる可能性があります。 別のアプローチとしては、ゼロからシナリオを作成するのに時間をかけるのではなく、一般的な機能テストや実世界のログのデータセットを事前に購入して、そこからシナリオを作成することもできます。シナリオの品質が高く（現実的な動作、パラメータの選択、評価基準など）、チームの要件や能力に応じて構築されている限り、事前構築されたライブラリの購入は有効な選択肢となります。

評価基準 ・ 指標の明確化

シナリオは、システムの性能をテストする評価基準の指定を行うことでテストケースになります。自動運転プログラムは、各テストケースについて、測定可能な総合的な合格 ・ 不合格の結果を追跡する必要があります。 この結果は、 主要な能力、安全性、快適性の要素の合成であり、任意でない評価ルールは、 それぞれとその基礎となるメトリクスを掘り下げる能力ですべて合格しなければなりません。以下のリストには、チームがテストケースについて評価すべき指標と評価基準が記載されています。

I. **テストの妥当性：** シナリオが意図したテスト設計を表していることを確認します。例えば、アクターカットインシナリオでは、アクターは常に自律走行システムの前でレーンチェンジを実行します。

II. **システムの中核的な意図：** 自律走行システムがテストケースの主目的を完了することを確認します（例： 左折シナリオで意図した目的地に到達する）。

III. **安全性：** 衝突は、安全性に関して追跡すべき最も明白な測定基準です。 衝突の発生頻度は低いので、チームは代理として別の指標を測定することができます。チームは、TTC、PET、衝突回避減速度 (DRAC) 、および衝突時の速度などの重大性を把握する測定基準などの安全性の代用指標を計算し、追跡する必要があります。また、RSS (Responsibility-Sensitive Safety) 、他のアクターとの横方向および縦方向の距離、車線中央からのずれなどの情報的な指標も測定および追跡するのに有益です。

IV. **快適性：** 自動運転システムの滑らかさ（ジャークなど）を測定します。

V. **アサーティブネス：** システムが最初に安全な操作の機会を得たかどうかを測定します。 例えば、保護されていない右折の場合、システムが安全に曲がることができる「隙間」 がいくつかある可能性があります。 この場合、システムが可能な限り早い段階で安全な隙間に入るかどうかを測定することが有効です。

VI. **効率性：** トリップまたはタスクが完了したかどうか、および自動運転システムのルーティング効率を測定します。例えば、完了したタスクとは、 自動運転トラックが合理的な時間内に希望の出発地と目的地から貨物をピックアップし、ドロップオフするというケースの場合があります。

VII. **道路のルール：** システムが道路や公共環境の規則に従っているかどうかを測定します。 道路交通車両の場合、

ステージ	説明
初期	一般的な不具合を明らかにするために大部分がシミュレーションで実行され、クローズドトラックのテストも一部行われるスモークテストの初期セット。新しいシナリオ作成の焦点は、スモークテスト用のノミナルシナリオの構築と基本的なドライブログの取得と、ログからのシナリオ生成である。
中期	各シナリオカテゴリまたは要件ごとの少数のシナリオ。通常、機能開発者のリリースケーデンスと活発な機能開発に合わせて、少数のシナリオカテゴリまたは要件についてカバレッジを深くすることに焦点を合わせる。
後期	ODD、要件、シナリオのカテゴリーを網羅した完全なシナリオライブラリー。エッジケースやロングテールシナリオ（実世界のドライブやランダム化手法によるもの）、または次のODDに向けた新しいシナリオの構築に重点を置いています。

図8:V&Vのステージ別シナリオ作成

これには通常、道路標識（停止、譲るなど）、信号、速度制限などが含まれます。

シナリオの健全性を長期的に維持する

シナリオの健全性とは、システムやテストインフラが変化しても、各テストケースの「鮮度」または生存率を維持できることを指します。 例えば、アクターのカットインシナリオが数ヶ月あるいは数年前のものである場合、自律走行システムの動作が大幅に変化し、アクターはもはや自律走行システムの前にカットインしていないかもしれません。この時点で、シナリオはもはや意図されたものをテストしていないため、「陳腐化」 しているということになります。したがって、自動運転システムには、新鮮さを維持し、時間の経過とともに更新される必要のある古くなったシナリオを特定するシステムが必要です。 例えば、新しいシナリオを数回実行し、それが有用であるか、意図するものをテストしているかを評価する必要があります。 開発が進むにつれて、チームはシナリオを監視し、古くなっていないことを確認する必要があります。これには、シナリオがまだ有効かどうか、偽陽性や偽陰性を引き起こしていないかどうかをチェックすることが含まれます。チームは、この2つのニーズに対応するために、次のようなステップを踏むことができます。

I. シナリオを作成する際には、数回の実行とソフトウェアビルドを監視します。ビルドのリリースやテストバッチの実行の頻度にもよりますが、1週間から3ヶ月かかることもあります。
II. シナリオを計画的なアドホック評価で実行し、CIパイプラインまたはナイトリーテストケーデンスに追加します。CIパイプラインにシナリオを追加する場合、テストは安定した段階になく、まだ正式なリリース検証や分析ワークフローの一部であってはならないため、 シナリオをノンブロッキングとラベル付けしてください。
III. 評価基準と測定基準の定義の項で定義した測定基準について、シナリオを追跡します。 特に、テストの妥当性とテストの中核的な意図を監視し、プレイバックを見て、シナリオに予期せぬものがないこと、意図したものをテストしていることを確認します。
IV. シナリオに問題が発生した場合、シミュレーション運用チーム、または指定されたシナリオ作成チームやトリアージチームは、 シナリオの成熟度と堅牢性が証明されるまで、原因を診断し、修正を行い、プロセスを再開する必要があります。
V. シナリオが安定したとみなされると、チームはそのシナリオをアドホックテストに積極的に使用したり、 CIパイプラインや夜間評価に追加したり、シナリオやODD分類によってタグ付けして全体のカバレッジに貢献させたりすることができます。
VI. チームは、シナリオが古くならないように、安定したとみなされた後でも、すべての シナリオの健全性を追跡調査する必要があります。 これは、ダッシュボードでシナリオの指標を毎日確認し、定期的な（月例やリリースに依存した）レビューを設定することで可能です。このレビューでは、 各シナリオに予期せぬものが含まれていないか、意図したとおりのテストが行われているかを確認する必要があります。 例えば、スタックのテスト対象部分に変更がないのにシナリオが失敗し始めた場合、シナリオが陳腐化していないかテストすることをお勧めします。シナリオが失敗するのは、スタックの他の部分の変更による下流への影響か、 あるいは陳腐化したかのどちらかです。 チームは特に、自律走行システムソフトウェア、マップデータ、メトリクスまたはオブザーバロジックに大きな更新があった後に、このレビューを行うべきです。

また、自動運転プログラムでは、カスタムメトリックとオブザーバーの健全性を追跡 ・ 管理するために、同様のプロセスを設定する必要があります。

F.テスト実行

次の表は、自動運転プログラムが開発の各段階で通常使用するテスト方法と、各段階で実環境テストが果たす役割を整理したものです（図9）。自動運転開発チームは、できるだけ早くシミュレーションの利用を拡大することで、スケーリングとコストの問題を防ぐことができます。また、車両試験を主要試験よりも最終的な検証やエッジケースの発見に重点を置いて移行することも効果的です。

各テスト環境の有効活用

国連欧州経済委員会 (UNECE) は、公表している自動運転の検証手法 (VMAD) と自動運転の新評価／試験手法 (NATM) (図10) を通じて、自動運転システム の検証に対する多種組み合わせのアプローチを提唱しています。 自動運転プログラムが使用すべきさまざまな試験方法と環境は、このハンドブックの次小節に最も関連性があります。

同様に、ASAM (Association for Standardization of Automation and Measuring Systems) は、2022年にADASと自動運転システムの検証に必要なさまざまな試験手順と環境を示した報告書を発表しています (図11)。

図10と図11で紹介したテスト環境には、それぞれ長所と短所があります。上の表は、各環境の長所と短所について述べたものである (図12)。

すべてのテスト環境は必要ですが、それぞれのテスト環境には最適な利用方法があります。以下のリストは、テストリソースを最大限に活用するために、自動運転プログラムが各テスト環境をどのように使用すべきかを示したものです。

I. 合成テストと再シミュレーションテストを組み合わせて、テストの大部分をSILシミュレーションテスト環境で実行する必要があります。シミュレーション優先のテスト戦略は、迅速かつ安全で、経済的にも実行性が高いです。
II. 最初にシミュレーションテストでパスする必要があります。システムがシミュレーションテストの大部分にパスしたと確信するまでは、HILと実環境のシナリオベースのテストリソースは節約されるべきです。 HILテストは、ネットワークの設計と開発（通信と診断テストを含む）の初期によく使用されます。
III. その後、テストコースでシナリオの一部を物理的にテストする必要があります。 また、テストコースを利用して、シミュレーション試験が適切にモデル化されていることを検証し、実環境試験に進む前に閉鎖的な環境で車両の全体的な性能をテストする必要があります。
IV. シミュレーションと閉じたテストトラック環境での状況をシステムがクリアできるという確信が十分に得られたら、チームは公道での実環境試験でシミュレーションの忠実度をさらに検証することができます。また、実車テストは、テストコーステストや仮想テスト（合成シナリオと再シミュレーションの両方） で使用する新しいシナリオを特定するためにも最適な方法です。

段階	説明
初期	テストは主にテストコースで行い、チームが高品質なシミュレータを利用できシミュレーションの利用を増やしたい場合を除いて、ほとんどのテストはテストコースで行われる。場合によっては、チームがシミュレーションに移行する間、実走テストもバグや問題を見つけるため行われる。MILテストも初期のアルゴリズム開発に利用し、HILテストはネットワーク設計と開発 (通信と診断のテストを含む) に利用する。
中期	MIL試験、SIL試験、テストコースを組み合わせての試験が中心だが、HIL試験、車両統合試験、実走試験 (走行ログ収集含む) も強化。
後期	MIL、SIL、HIL、車両 (車両統合、VIL、テストコース、実走) テストを含むすべてのテスト環境を使用。主に車両テストを使用して、シミュレーションの忠実度を検証し、車両統合をテストし、公道でのエッジケースシナリオを見つける。規模やコストの面から、テストの大部分をシミュレーションで実施。

図9:V&Vの段階別テスト方法

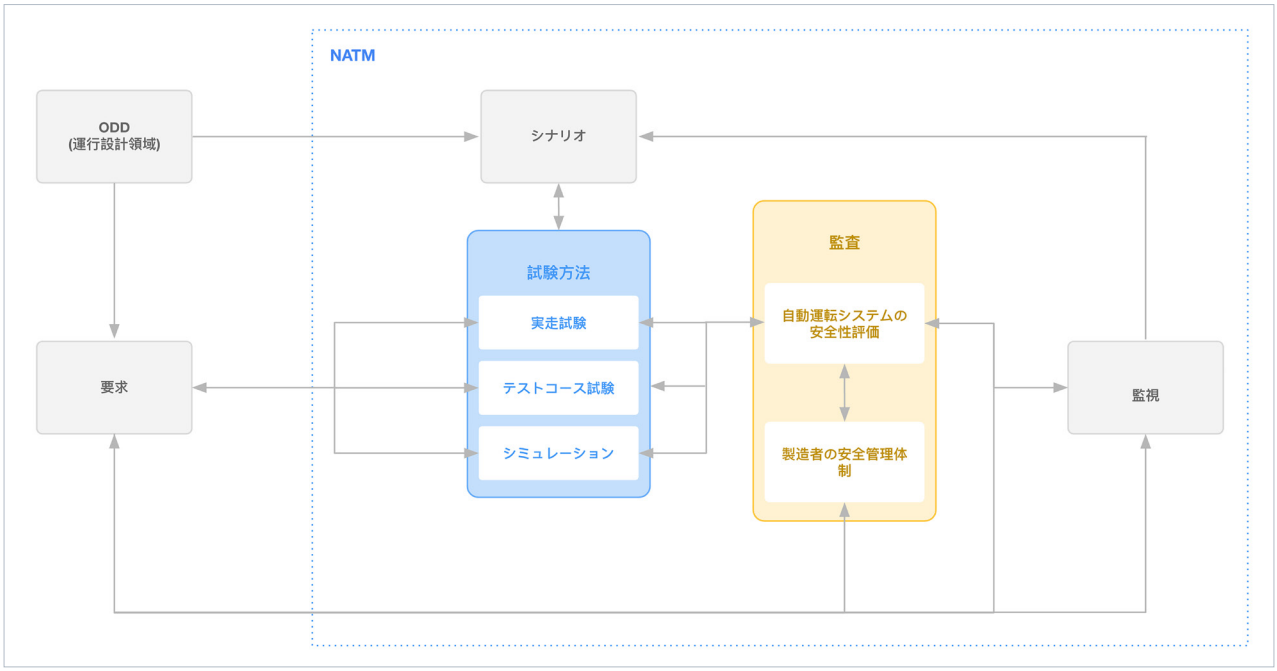


図10:NATMは、UNECEとGRVA（自動運転／自律走行／コネクテッドカーに関する作業部会）による自動運転システム検証のための多種組み合わせのアプローチ

	要求仕様に基づくテスト	インターフェーステスト	フォールトインジェクション	リソースの使用性能試験	シナリオに基づく試験
テスト環境					
MIL	要求仕様に基づくMIL試験		MILのフォルトインジェクション		制御部品の検証
ソフトウェア再処理	オープンループによるADAS/自動運転システムソフトウェアのテスト		ロバスト性の評価		
クローズドループSIL	ADAS/自動運転ソフトウェアスタックのクローズドループでのテスト	ソフトウェア統合テスト	安全機構の検証		シナリオベースSILクローズドループテスト
ハードウェア再処理/データ再生		ハードウェアの再処理とデータ再生	ハードウェアを含む安全機構の検証		
クローズドループHIL	ADAS/自動運転機能のクローズドループにおける完全な影響連鎖のテスト		統合されたシステムによる安全機構のテスト	車両ネットワークの性能試験	エレクトロニクス統合の検証
VIL	要求仕様に基づくVIL試験	システムレベルでのADAS/自動運転影響連鎖全体のテスト			システムレベルでの検証
ドライバインザループ（DIL）			システム全体の動作検証		ドライバーと安全関連車両機能との相互作用の検証
実験場	制御された実験場環境下でのテスト		システム全体の性能の検証		制御された環境下でのシステム反応テスト
オープンロードテスト/フィールドモニタリング	ADAS/自動運転機能のフィールドでの実ユースケースでのテスト				実ユースケースでシステム全体の検証

図11: ADASと自動運転システムのテスト手法と環境を網羅するASAMのテストの全容

テスト環境	説明	長所	短所
仮想テスト・シミュレーション	MIL、SIL、HIL、VIL の各試験において、さまざまなレベルのソフトウェアとハードウェアの要素が仮想環境と相互作用する。	• システム変更時の再テストを迅速に行うことができる • 危険で、収集や実環境での再現が困難なシナリオなど、テスト対象を高度にコントロールすることができる。 • 同じテストを再度実行しても、前回の実行とほとんど変わらない。 • 物理テストと比較して、仮想テストのセットアップと実行にかかるコストを削減できる。	• シミュレーションの忠実性を実証する必要があり、すべてのテストシナリオで実現するのは難しい。 • テストの質は、シナリオライブラリの質にも依存する。 • セーフティドライバーの運転性能やスムーズさなどの「主観的な感覚」を客観的に評価し、合格・不合格の基準を定めることが困難。
テストコース	閉じた試験場で、本物の障害物やサロゲートを使った物理的な車両試験	• 実走行よりもテスト内容を高度にコントロール可能。 • 仮想テストよりも高い忠実度。 • 同じテストを再度実行しても、前回の実行結果との乖離が比較的小さい。 • 実環境テストより安全。	• セットアップと実行に多大なコストと時間がかかる。(人員と専門機器が必要なため) • テストのばらつきは、テストコースのインフラ、条件、利用可能な機器に制限される(例: 異なる天候条件、異なるアクタータイプのテスト) • セーフティドライバーの安全上リスクの考慮が必要
実走行テスト(公道など)	実際の交通状況や真の動作環境でのシステムテスト	• 他のすべてのテスト環境と比較して、最も高い忠実度。 • 真の動作環境でのテストが可能。 • これまで考慮されていなかった新しい事象を発見する可能性。 • テストコースでは不可能な条件下での車両テストが可能 (例: 橋、トンネル)。	• 現実の世界でどのような状況が発生するかを正確にコントロールすることが難しい。 • 同じテストを、前回の実行から中〜大規模な変化なしに再実行することは容易ではない。 • テスト回数を増やすとコスト増になる(オペレーター、専用テスト機器) • 重要ではあるが、実世界でテストするのは稀であったり、危険な状況もある(例: 子供が道路を走って渡るなど)

図12:仮想テスト、テストコーステスト、実環境テストの長所と短所(NATMより引用)

V. 自動運転プログラムでは、成熟したプログラムのみが自動運転モードでの実環境試験を実施することが認定されます。 初期段階では、自動運転プログラムは、システムに十分な信頼性が得られるまでは、実環境試験の実施は控えることとなります。システムが十分に成熟したら、シミュレーションやテストコース試験と並行して実環境試験を実施することができます。 しかし、新しい機能をテストする場合、 チームはよりコストの高いテスト環境に進む前に、最初のパスとチェックポイントとしてシミュレーションテストを使用する必要があります。

シナリオに基づくテストにおける組合せ急増への対応

自動運転プログラムは、検証、安全、開発チームに最も多くの情報を提供するため、セーフティクリティカルな状況やシナリオにリソースを集中させる必要があります。しかし、シナリオライブラリは、テストプログラム全体が成熟するにつれて、常にサイズが大きくなっていきます。テストする必要のあるシナリオの数は、通常、新しい要求の数に対して直線的に増加します。シナリオ空間のボリュームと、チームが実行する必要のあるシナリオの総数は、カバーする必要のあるODD属性とパラメータの数によって、指数関数的に増加します。

下の図は、自動運転プログラムが 1つのテストケースのすべての順列を網羅的に評価するために、160 万のバリエーションをテストする必要があることを示しています（図13）。この例では、必要なテスト数をさらに指数関数的に増加させるであろう、環境条件（例：時間帯、降雨）、地図の位置、および高い粒度の行動パラメータを除外しています。 さらに、この図は単一のテストケースしか含んでいませんが、自動運転プログラムではリリースごとに数千のテストケースを実行する必要があります。この例は、自動運転システムの網羅的テストにおける組合せ急増の問題を示しています。

自動運転プログラムは組合せ急増に実用的に対処するにはどうしたらよいでしょうか？まず最初に、Applied Intuitionはシミュレーション優先のテスト戦略を推奨します（「各テスト環境の有効活用」を参照）。しかし、シミュレーションを用いても、ソフトウェアのリリースごとに何億ものシナリオを実行しなければならない可能性があるという問題があります。

スケーリングされたシミュレーション戦略を補完するために，自動運転プログラムは，そのニーズに基づいて多次元パラメータ空間を賢くにサンプリングする必要があります。 プ

例: カットインシナリオ
自動運転システムの車線に隣接車両が割り込む

アクターパラメーター	バリエーション
1. アクタータイプ	5
2. カットイン側	2
3. アクターの初期位置	10
4. アクターの最終位置	10
5. カットイン速度	10
自動運転システムパラメータ	
6. 自動運転システムの初期速度	10
7. 自動運転システムの初期レーン	4
道路と環境	
8. レーン数	4
可能なバリエーションの総数	
160万バリエーション $(5 \times 2 \times 10^4 \times 4^2)$	

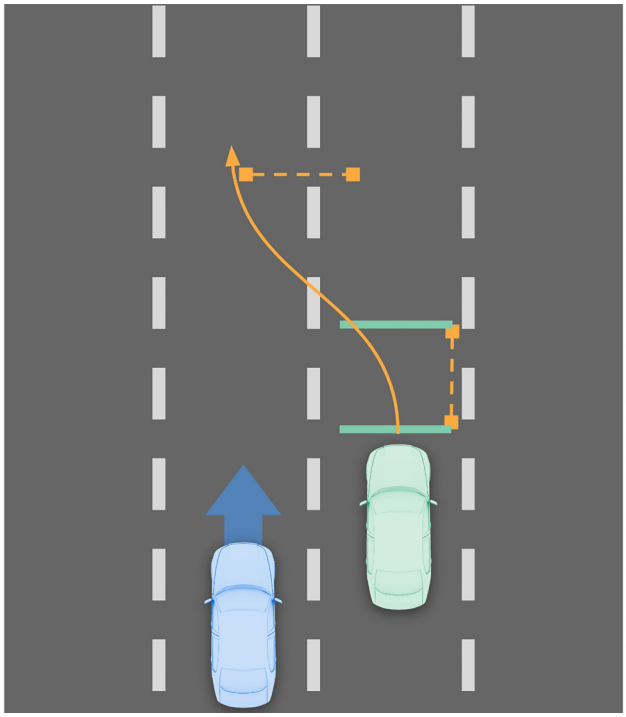


図13:カットインシナリオの例：各テストパラメータについて一握りの値だけでも網羅的にテストすると、1つのテストケースで160万通りのバリエーションが必要になる

ログラムの成熟度に応じて、最適化の目標は、一方ではカバレッジとODDに関する情報取得のためのテスト、他方では開発を前進させるためのセーフティクリティカルシナリオの発見を組み合わせたものにすべきです（図14）。

プログラムは、特定の目標に基づき、テスト、開発、重要な情報収集のスピードを上げるために、さまざまなテクニックを活用することができます。

I. **初期のプログラム**では、重要度サンプリングのような統計的手法で、最も多くの新しい情報を提供するシナリオのバリエーションを特定することから始めるべきです。重要度サンプリングを使用する場合、関心のあるパラメータに対する走行およびシミュレーション中に収集された指標の分散を測定することによって、シナリオ空間の特に敏感な、したがって重要な領域を特定することができます。 これにより，グリッドサンプリングアプローチ（パラメータのすべての組み合わせをテストすること）に依存するよりも，自動運転プログラムがシステムの問題を迅速に発見することができます。
II. **中期のプログラム**では，重要度サンプリングと敵対的サンプリングの手法を組み合わせて，多次元パラメータ空間のうち失敗しそうな領域を特定する必要があります。敵対的サンプリングでは、特定の機能 （カットイン性能など） をストレステストし、合否確率に大きな影響を与えるパラメータの組み合わせを特定して、失敗領域を分類することができます。一方、重要度サンプリングは、より広範なカバレッジを構築するのに役立ちます。
III. **後期プログラム**では、ほとんどのテストが合格する成熟段階にあります。 その目標は、エッジケースを発見することです。チームは、リグレッションをチェックするために、一連の名目上のケースを実行する必要があります。次に、敵対的サンプリング （生成的でより複雑な特徴づけ技術） を用いて、セーフティクリティカルな変数を特定し、その変数にテストを集中させる必要があります。 この方法は、名目上のテストケースや扱いやすいテストケースから大きなバイアスをかけるため、実行するバリエーション数を何桁も減らすことができます。

自動運転プログラムでは、すべてのシナリオの組み合わせをテストすることは不可能です。したがって、最終的な安全性評価とセーフティケースの一部として、プログラムは規制当局、監査人、および顧客に対して、1) なぜ特定のテストケースとシナリオのバリエーションをテストすることにしたのか、 2) なぜそれらのテストケースとシナリオのバリエーションが十分であるかを実証する必要があるのです。カバレッジの定義と測定では、このトピックについてさらに詳しく説明します。

G. 分析とレポーティング

カバレッジの定義と測定

カバレッジは、自動運転システムがこれまでに何をテストされたかを測定することです。自動運転プログラムにとって、テストの包括性を証明するためには、カバレッジの正式な測定が必要です。カバレッジの方法論が十分であるためには、既知/未知の情報 （すなわち、存在していてテストできることを知っている/知らない情報）およびカバー/未カバーの情報（すなわち、すでにテストされている/されていない情報） の空間を測定しなければなりません。下の表は、この実践を図解したものです（図15）。 現実的には、初期段階のプログラムは既知のものに大きな焦点を当て、後期段階のプログラムは未知のものに大きな焦点を当てるべきでしょう。

言い換えれば、カバレッジとは、既知でテストされたものと、システムが直面する可能性のある状況の総空間との間の比

ステージ	説明
初期	重要度サンプリングなどの標準的な統計手法を用いた広範なカバレッジの構築
中期	重要度サンプリングにより広範なカバレッジを構築し、その後、コストと時間の大半を、そのリリースが焦点を当てている機能の敵対的サンプリングに優先させる。
後期	重要度サンプリングにより広範なカバレッジを構築し、敵対的サンプリングで生成的でより複雑な特徴づけ技術を組み合わせ、システム全体のストレステストを行う。

図14:V&Vステージ別インテリジェントシナリオベースのテスト手法の推奨最適化目標

率を意味します。自動運転プログラムでは、まずODDを定義する必要があります。そして、ODDの一部が十分にテストされているかどうかを判断する方法を定義する必要があります。　これら2つの側面が定義されれば、チームは、個々のシナリオ／テストケースのレベルから、自動運転プログラムのODD全体まで、あらゆる粒度でカバレッジを独自に測定することができます（図16）。 初期段階の自動運転プログラムでは、カバレッジは、各シナリオカテゴリーのテスト数を単純に数えることから始めることができます。 後期のプログラムでは、カバレッジは、テストされた内容の包括性を示す統計的な指標に発展させる必要があります。

V&Vの初期段階にあるプログラムでは、主要な目標は機能開発であるべきです。 このため、初期のプログラムにおけるカバレッジの測定は、潜在的な機能のギャップを特定し、それを埋めるのに役立ちます。 したがって、カバレッジの測定基準は、以下の質問に答えるためのものでなければなりません。

- 開発する上で最も重要な機能は何でしょうか？
- どの機能がもっとも多くのワークが必要になるでしょうか？
- どの機能がもっとも十分にテストされていないでしょうか？

この段階では、上記のプロセスを通じてODDを徹底的に定義することが重要です（要件とODDの定義、包括的なシナリオライブラリの作成を参照）。機能要件が定義されたら、最も関連性の高いカバレッジの指標は、既知のODD空間に対してテストされた要件とテストケースの割合になります。

$$\text{カバレッジ} = \frac{\text{既知のカバーされた空間}}{\text{既知の空間}}$$

自動運転プログラムが中期段階に成熟し始めると、その焦点はセーフティケースの作成に移行する必要があります。 この段階では、カバレッジの役割は機能開発の推進から、成熟度と安全性の証明へと移行します。したがって、カバレッジメトリクスに関する主要な質問は次のようなものになるでしょう。

- 機能が成熟していると見なされるためには、どのような追加のワークが必要ですか？
- ある機能の動作が不明となるような何らかの状況は考えられますか？

プログラムが後期に入り、正式に定義された要求のほとんどが完全にカバーされると、焦点は再びテストされていない未知のものをカバーすることに移ります。ここでは、カバレッジメトリクスの主な質問は次のものになります。

- 既知のODD情報空間において、自動運転システムはどの程度安全ですか？
- まだ遭遇していない状況で、自動運転システムはどの程度安全ですか？

この段階では、このような新しい問題に対処するために、チームは新しい情報の到着率を用いてカバレッジを測定し、ODDの未知のものの上限を定量的に測定することも必要です。

自動運転プログラムでは、以下の手順でカバレッジメトリクスを定義し、カバレッジ解析ワークフローを設定することができます。

I. 形式化された分類法（タクソノミー）を用いて、できるだけ早く ODD を定義します。ODD タクソノミーとは、自動運転システムが ODD で扱う必要のある環境条件、物体、行動、道路インフラを定義します、パラメータ（連続値の場合はパラメータ範囲）の列挙された各属性の整理された集合です。これは 2 つの主要な構成要素から成り立っています。
    - ODD 属性は、ODD の側（例えば「道路タイプ」、「道路曲率」、「時間帯」）を定義します。
    - ODD パラメータは ODD 属性を定義し、定量化します（例えば、ODD 属性「道路タイプ」に

	既知の情報空間	既知と未知の情報空間	未知の情報空間
カバーされた情報空間	初期段階プログラム	中期段階プログラム	中期段階プログラム
未カバーな情報空間	初期段階プログラム	中期段階プログラム	後期段階プログラム

図 15：自動運転プログラムが V&V の段階ごとに測定すべき情報（既知/未知、カバー/未カバー）。

ステージ	商品説明
初期	ハイレベルな能力、シナリオカテゴリ、ODD属性ごとのテスト数によるカバレッジの測定
中期	各要件のテスト数の計測を追加し、各 ODD 属性のパラメータ毎の ODD カバレッジの詳細な追跡を開始する。また、マップの早期カバレッジ分析も開始する。
後期	カバレッジの統計的指標を追加

図 16：V&Vステージ別の推奨カバレッジ測定方法

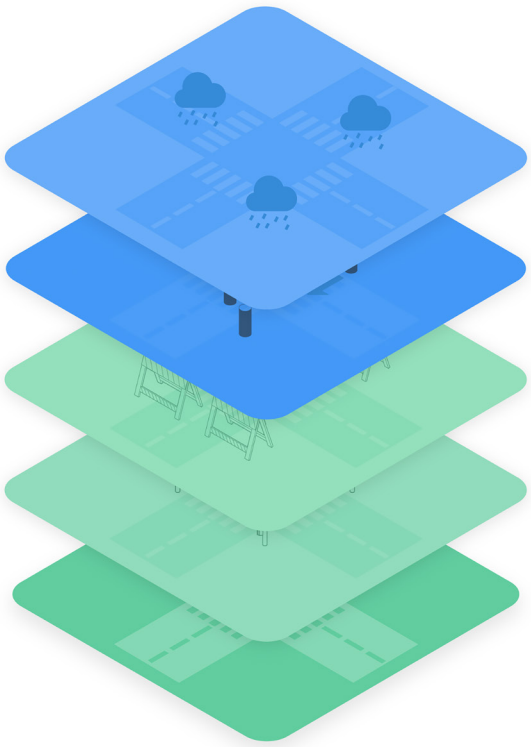
対して、パラメータは「Local（ローカル道路）」、「Arterial（幹線道路）」、「Collector（補助幹線道路）」などが考えられます）。

ODDの属性やパラメータをタクソノミーに従って列挙する際に参考になるのが、PEGASUS の手法で、これは環境トポロジー、交通インフラ、環境状態、オブジェクトとエージェント、環境状態、 デジタル情報の6つの独立した層（図17）に基づいて、シナリオを体系的に記述するためのモデルを提供するものです。

II. 定義されたタクソノミーに従って、各テストを分類します。
III. テストを要件にリンクさせ、 ODD カバレッジ （すなわち、これまでに ODD 分類法のどれだけの部分がカバーされたかを示す統計的指標）、および、要件カバレッジ（すなわち、各要件を評価するテストの数）を追跡します。
IV. 開発の初期段階では、能力、シナリオカテゴリ、ODD 属性ごとにテスト数を評価することで、カバレッジを測定します。
V. 中期段階では、ODD タクソノミー（各属性のパラメータの組み合わせ）で定義された、考えられるすべてのテストケースとシチュエーションのカバレッジの加重和を計算することで、ODDカバレッジを測定します。さらに、各要件が作り込まれた段階で、各要件のテスト数の評価を開始します。
VI. 実環境でのテストが始まったら、ODD タクソノミーで定義されていない新しいオブジェクトやシナリオに遭遇したかどうかの評価を開始します。ODD 分類法に従って、ドライブテストの分類を開始します。
VII. これらの ODD パラメータの組み合わせを、カバーすべき追加シナリオに定式化します。
VIII. ドライブデータと実世界の分布を使って、各シチュエーションの確率を測定します。 これは、現実の事象がどの程度カバーされているかの客観的な指標となります。
IX. 要件のカバレッジは、テストケースが関連付けられている要件の割合と、要件の検証に使用されたテストケースの数の両方によって測定されます。
X. 異なる地図要素 （例：異なる道路の曲率、道路状況、道路タイプ）および地図セグメント（例：ジオフェンスで囲まれた地域の物理的位置） にわたってテストの分布を評価することによって、地図のカバレッジを測定します。
XI. 開発後期には、新しい情報の到着率に基づいた厳密なカバレッジメトリクスを定義します。
XII. これらのカバレッジメトリックスを開発期間中追跡し、機能開発、シナリオ作成、および実環境でのテストの指針とします。これらのメトリクスは、特にSOTIFとエビデンスベースの安全性の一部として、セーフティケースに大きく参照されます。

最終的には、カバレッジを定義・拡大することが、次のような点で自動運転プログラムに役立ちます。

I. 機能、バグフィックス、システムチューニングを、 ODDにおけるこれらの項目の頻度と重要度に照らして定量的に優先順位付けを行うことができます。
II. ODD のより稀な部分集合を繰り返し発見し、カバーし、実行する。これにより、自動運転システムは単純な状況をカバーすることから、非常に複雑で微妙な状況をカバーすることへと進展できます。
III. カバレッジの不確実性とカバレッジギャップの指標に基づいて、どのような状況でより多くの実データを収集する必要があるか、またはシナリオを作成する必要があるかを理解することにより、データ収集とシナリオ作成を最適化できます。
IV. 運用中に新たに発生する未発見の事象の統計的な可能性と、それらの事象がセーフティクリティカルな問題



環境条件

降雨の種類と速度、降雨の蓄積量、照明条件、路面摩擦係数

オブジェクトとエージェント

挙動とトリガーによる静的および動的なオブジェクトの定義

環境状態

一時的な変更およびイベント (例: 道路工事や車線閉鎖など)

交通インフラストラクチャ

信号機、標識、ストップバーなどの交通規制装置

環境トポロジー

ジオメトリ、ネットワークトポロジ、路面標示

図17: シナリオモデリングのためのPEGASUSのレイヤー。図には描画されていないデジタル情報 (Vehicle-to-everything、デジタルデータ／地図情報など) のための6番目のレイヤーも含まれる。

を引き起こす可能性を厳密に定義できます。

V. ODDが十分にテストされたかどうかを測定し、テストすべき欠落したシナリオがないかどうかを評価できます。

VI. 安全性の議論に重要な役割を果たし、システムが安全に動作することが予期されるすべてのシナリオでテストされているということに関して消費者や規制当局との信頼関係を構築することに貢献します。

パフォーマンスの分析

このハンドブックのこのセクションは、「リリース検証プロセスの生産」、「評価基準・測定基準の定義」と併せてお読みください。パフォーマンス分析の目的は、自動運転システムが安全に処理できる条件とできない条件を理解することです。 前者はカバレッジ分析と相まって、システムが性能を発揮できるはずの安全な動作領域として機能します。後者は、開発側で改善すべき重点領域となります。 また、パフォーマンス分析は、前回のリリースからのプログレッションやリグレッションを測定し、そのリリースが承認に値するかどうかを判断するのに役立ちます。次の表は、初期、中期、後期の自動運転プログラムに対して推奨されるパフォーマンス分析プロセスを整理したものです (図 18)。すべてのチームが KPI と SPI を追跡する必要があります。後期プログラムでは、正式な A/B テストが重視されるようになります。A/B テストとは、マスターブランチと開発ブランチのパフォーマンスを同じテストセットで評価・ 比較し、制御された実験でプログレッションとリグレッションを判断する方法です。

チームは一般に、開発およびテストを通じて、 次のような方法でパフォーマンス分析を行うのが良いでしょう。

I. システム全体の性能と安全性を測定するKPIとSPIを定義します (「評価基準と測定基準の定義」セクションの例を参照)。

II. 各 KPI と SPI のパフォーマンスをライブダッシュボードの形でハイレベルに表示します。 エグゼクティブを含む組織のすべてのメンバーは、これらのダッシュボードを使用して、次のことを行うことができます。

   a. 現在のソフトウェアビルドまたは最新のテストバッチ (夜間テストなど) から得た最新の情報を提示し、以前のビルドと比較します。

   b. リリースを承認するかどうかの判断を行い、プログラム全体の安全性とパフォーマンスレベルを監視します。

c. 各指標のリリース別、テスト別の傾向を時系列で確認できるよう、可視化機能を追加します。
d. 各指標の値、分布、期待値や安全値との乖離をチームメンバーが確認できるような可視化機能を追加します。

III. このダッシュボードを定期的（例：毎日）にレビューし、障害とリグレッションに対処します。 トリアージチームは、問題の根本原因を診断し、テスト結果のコピーとプレイバックを添付した障害レポートを、その問題に対処する担当の開発者に共有します。

IV. 開発者が使用できる、より詳細なダッシュボードを作成します。 開発者は、ハイレベルなパフォーマンス・ダッシュボードを活用しつつ、より詳細なダッシュボードを厳格な A/B テストに使用する必要があります。開発者は、コードを変更するたびに、その影響を理解する必要があります。そのために、統計やデータサイエンスなどの他の部門と協力して、前回のリリースから定量的に改善されているかどうかを評価することもあります。また、このグループは、すべてのテストタイプにおけるパフォーマンスを調べ、あるメトリクスでプログレッションが起こり、他のメトリクスでリグレッションが起こるという避けられないケースにおけるトレードオフを評価するために協力することもあります。

V. 選択した各リグレッションを注意深く精査し、その原因を理解します。リグレッションごとに課題チケットを作成し、次のリリースで解決されるように追跡します。チームはプログレッションを喜ぶことができますが、同時に、将来のリリースで逆戻りしないように注意深く追跡する必要があります。

VI. 各リリースを要件レベルで分析します。各要件が許容レベルまで合格しているかどうかを測定します。

VII. パフォーマンスレポートは、履歴の文書化とトレーサビリティのために定期的に作成されるべきです。

パフォーマンスの測定と評価は、次のような点で自動運転プログラムを支援します。

I. どの機能がうまく機能しているか、どの機能が最も多くの作業を必要としているかを定量的に判断する能力を向上させることができます。これにより、チームはどの機能に取り組むべきかの優先順位を客観的に判断することができます。

II. 動作のリグレッションを防ぎ、すべての作業が自動運転プログラムのプログレッションに寄与するようにすることで、開発速度を速めることができます。

III. 安全性の議論に重要な役割を果たし、システムが安全に動作できると期待されるすべてのシナリオで良好な性能を発揮するという、消費者や規制当局との信頼関係を構築することに貢献します。

ステージ	説明
初期	スモークテストで発生した全ての不具合を、重要度によって優先的に修正。初期オブザーバはまだ開発・調整中（安定性の追跡が必要）。KPI と SPI の週次の追跡
中期	正式な A/B テストの強化（開発の反復モデルを反映し、リグレッションを修正するため）。初期オブザーバーセットの設定。アドバンストなオブザーバーの開発と調整（安定性を追跡するため）。KPI と SPI の日次での追跡
後期	SPI と KPI 間のトレードオフの評価を含む、前回のリリースからの定量的な改善点に関する大規模な A/B テストと統計分析。KPI と SPI の日々の追跡

図18:V&Vステージ別パフォーマンス分析プロセス

結論

このハンドブックは、安全性フレームワークの構築と堅牢なV&Vプロセスの確立において、自動運転プログラムを支援することを目的としています。 私たちは、自動運転プログラムが安全性フレームワークを定義し、自動運転システムを安全に開発し、テストし、デプロイするためのアクティブなリソースとして役立つことを期待しています。

このハンドブックは、安全性フレームワークの構築方法とV&Vの実施方法に関する多くの質問に答えていますが、お客様の自動運転プログラムの詳細については、まだ答えが見つかっていない部分もあると思われます。Applied Intuitionは、業界をリードする自動運転開発 ・ 検証ソリューションと業界のベストプラクティスに関する専門知識を組み合わせ、お客様のプログラムのV&Vと商業化の目標をサポートする準備が整っています。

Applied Intuition のV&Vプラットフォーム「Basis」に関するご質問や詳細は、applied.co/contact までご連絡ください。

用語集

ACC: アダプティブクルーズコントロール
ADAS: Advanced Driver Assistance System（先進運転支援システム）
ADS: 自動運転システム
AEB: 自動緊急ブレーキ
ALKS: 自動車線維持システム
ALM: アプリケーション・ライフサイクル・マネジメント
ASAM: Association for Standardization of Automation and Measuring Systems（自動化および測定システム標準化協会）
ASIL: Automotive Safety Integrity Levelの略。
AVSC: Automated Vehicle Safety Consortium（自動運転車安全コンソーシアム）
CI: 継続的インテグレーション
DIL: ドライバ・イン・ザ・ループ
DR: 減速率
DRAC: 衝突回避のための減速率
EU: 欧州連合
FAA: 連邦航空局
FMEA: 故障モード影響解析
FTA: フォールトツリー解析
GRVA: 自動運転/自律走行/コネクテッド・ビークルに関するワーキングパーティー
HARA: ハザード分析およびリスクアセスメント
HIL: Hardware-in-the-Loop
HMI: ヒューマンマシンインターフェース
ISO: International Organization for Standardization（国際標準化機構）
IV&V: 独立した検証および妥当性確認
KPI: 重要業績評価指標
L2: SAEレベル2
L3: SAEレベル3
L4: SAEレベル4
MIL: モデル・イン・ザ・ループ
NATM: 自動運転に関する新しい評価・試験方法
NCAP: New Car Assessment Program（新車アセスメントプログラム）
NHTSA: 米国高速道路交通安全局
ODD: 運用設計領域
PET: ポスト・エンクローチメント・タイム
RSS: 責任感ある安全性
SAE: 米国自動車技術会（Society of Automotive Engineers）
SIL: ソフトウェア・イン・ザ・ループ
SOTIF: 意図した機能の安全性
SPI: セーフティパフォーマンスインディケーター
STPA: システム理論に基づくプロセス分析
TTC: 衝突までの時間
UL: Underwriters Laboratories（アンダーライターズ・ラボラトリーズ）
UN: 国際連合
UNECE: 国連欧州経済委員会
V&V: 検証・妥当性確認
VIL: Vehicle in-the-Loop
VMAD: 自動運転のための検証方法
VSSA: 安全性に関する自主的な自己評価

applied.co/contact