自社に最適な機能安全アセスメント構築へ

～エンジニアとアセッサーの認識ギャップを埋め、設計の手戻りをなくすには～

自動車業界に機能安全が本格導入されて 10 年以上経過し、安全設計の技術や Safety Case（安全論証）、機能安全アセスメントは、多くの会社や組織に定着してきました。しかしその一方で、アセスメントの現場では今もなお、多くの悩みが聞かれます。本稿では、近年の機能安全アセスメントが抱える課題に焦点を当て、エンジニアとアセッサーの間に生じる「すれ違い」を解消するためのヒントを紐解いていきます。

---

1. なぜ議論が噛み合わないのか：すれ違いの背景にある構造的要因

現場の悩みが尽きない背景には、近年の開発環境における 3 つの大きな変化があると考えられます。

- **第一に、SDV と E/E アーキテクチャの進化です。**
  ドメイン統合やセントラル化が進み、1 つのチップに複数のリスクの異なるアプリケーションを混在して搭載する Mixed-Criticality が一般化しました。これによりソフトウェア間の干渉リスクが増加し、安全構想から設計・実装に至るまで、Safety Case の論証がより複雑になっています。

- **第二に、OTA（Over-The-Air）を前提とした開発です。**
  SOP 後も短サイクルで機能追加や改修が繰り返されるため、設計・検証・論証のサイクルを高いレベルで連動させることが不可欠になりました。

- **第三に、AI/ML 技術や SEooC（Safety Element out of Context）の活用拡大です。**
  これらは便利な反面、前提条件の反映漏れやシナリオ網羅性の論拠など、安全性に関わる説明対象を一層増加させています。

近年、「評価者・設計者双方が、アセスメントの負荷に疲弊している」という声をよく耳にします。設計 FIX の直前に想定外の指摘で手戻りが発生する、あるいは、求められる提出物が雪だるま式に増え、プロジェクトが長期化する。これは決して"誰かが悪い"のではなく、E/E 開発の構造変化がもたらした必然的な帰結です。

この環境下で、エンジニアは「短期間で説得力ある Safety Case を構築する難しさ」に、アセッサーは「短期間でシステムの本質を見抜く難しさ」に直面しています。根本的な原因は、両者が「同じ事象を同じ粒度で評価するための共通フレーム」が不足している点にあります。アーキテクチャ図やログ、実測値などがそれぞれ別々に語られ、電源回路、RTOS／ハイパーバイザの設定、共有バスやキャッシュ、ロギング経路といった「見えない連結部」のリスクが見過ごされがちなのです。

---

2. ASIL デコンポジションに潜むリスク：すれ違いの典型例

この「すれ違い」を、機能安全で多用される ASIL デコンポジション（"独立性"の証明に必要な"設計"と"証拠"）を例に見てみましょう。

- **アセッサー**：「今回の電源回路の実装やスケジューリング設定では、これまでの論証がすべて覆ってしまう！物理的な分離だけでなく、ソフトウェアの相互干渉や共通原因故障（CCF）が十分に考

慮されているという証拠はどこにあるのか？」

- **エンジニア**：「安全構想に沿って設計、実装を進めてきた。非干渉性や独立性の証明は、一体何を示せば完了するのか？電源や RTOS のスケジューリングまで遡って、また論証を確認し直しか…」

こうしたすれ違いは、SoC への機能集約と Mixed-Criticality を前提とした昨今の開発で特に起こりがちです。その背景には、典型的な 2 つのギャップが存在します。

1. **アーキテクチャ図では二重化、しかし実装では共倒れの危険性** 設計上は独立したチャネルのつもりでも、電源レール、カーネル、共通ライブラリ、共通ロガーなどが繋がっていれば、一つの障害がシステム全体に波及する CCF（Common Causes Failure）の脆弱性を抱えたままになります。
2. **共有リソースによる実行時干渉** MMU/MPU やスケジューリング設定で分離していても、キャッシュ、バス、メモリなどの共有リソースで予期せぬ干渉が起これば、最悪実行時間の逸脱やデッドラインミスにつながり、デコンポジションの前提そのものが崩壊しかねません。

ここで重要なのは、いきなり解決策を議論するのではなく、まず不足している証拠と「見えない連結部」を、関係者全員が同じ解像度で可視化することです。例えば、以下のようなアプローチが有効です。

- **連結部の可視化** 電源（PMIC/レール）、OS/カーネル、ミドルウェア、ロギング経路、メモリ/I/O マップなどを一枚の図に整理し、依存関係を明らかにします。
- **干渉評価の目線合わせ**「干渉源（タスク/割り込み/DMA）× 共有リソース（キャッシュ/DRAM/バス）× 評価指標（遅延/逸脱/ロス率）」という観点で、どこを優先的に実測・評価するかを事前に合意します。

さらに、アセスメントのキックオフ段階で「独立性を担保する設計」と「独立性を証明する証拠の設計」をセットで合意できれば、開発終盤の「今さら…」という手戻りを減らすことができるはずです。

---

3. 実践的な学びは“自社製品”から：現場に即したトレーニングのすすめ

ここまでご紹介した論点は、一般的な座学だけではなかなか腹落ちしにくいテーマです。実際の議論は、電源系統の構成、RTOS の各種設定、ロギング経路といった、“製品固有の事情”に深く関わるからです。だからこそ、自社製品を教材として実践的に学ぶことが、課題解決への一番の近道です。 当社では、お客様の目的に合わせてカスタマイズ可能なトレーニングを提供しています。2 つのプライベートトレーニング例を紹介します。

3-1. 基礎を固める：社内の共通言語を揃えるワークショップ

- **事例：**機能安全実装ワークショップ『機能安全アセッサーのスキルアップ』
- **狙い**：例えば、確証方策やセーフティケースなどは各社固有の言葉や表現が使われることが多いため、機能安全に関する用語・観点・評価手法の認識を自社の用語や捉え方に短期間で統一します。
- **内容**：通常公開している講座のカリキュラムを、守秘義務契約の下、お客様社内で開催します。アセスメントの演習や議論の題材も、貴社の状況に合わせた事例、製品に置き換えて進行しま

す。

- **効果**：参加者全員が共通の視点（例：安全構想の捉え方、干渉リスクの分析手法）を持つことで、以降の議論や理解が格段にスムーズになります。

3-2. 実践力を磨く：自社製品で「使える型」を作るカスタマイズトレーニング

- **事例：**機能安全実装ワークショップ『機能安全アセスメントの活用方法』
- **狙い**：自社製品の"特徴"に合わせて、実務ですぐに使える分析手法やレビュー観点を「型」として確立します。
- **内容**：守秘義務契約の下、貴社の回路図・メモリマップ・テストログ等を教材に、「独立性の証拠の揃え方」といった自社の課題にフォーカスしたテーマで議論と演習を行います。すれ違いの要因を、貴社の成果物と照らし合わせながら整理します。
- **形式**：講義形式から、特定の論点を議論するワークショップ、ハンズオンまで柔軟に対応します。アセッサーとエンジニアの混成チームでの実施を推奨しています。
- **効果**：実際の設計現場での困りごとや課題に関する具体的な議論や Q&A を通して、自社製品の機能安全対応や機能安全アセスメントを促進させることができます。

「規格の知識は学んだが、現場の業務に活かせない」この壁を越えるには、自社（現地）で開発している製品（現物）を対象に実践を積むのが最短ルートです。本稿で取り上げた機能安全アセスメントに限らず、システム設計からハードウェア、ソフトウェア設計の各領域でも同様のトレーニングをご用意できます。プライベートトレーニングを有効活用し、現場の「困りごと」を、競争力を高める「自社ならではの強み」へと変えていきませんか。まずはお気軽にご相談ください。

【リンク】
機能安全　実装ワークショップのご案内　：当社 Web ページ「機能安全実装ワークショップ」
実装ワークショップのプライベート開催　：当社 Web ページ「機能安全実装ワークショップの設計現場でのご活用」
プライベートトレーニング　：当社 Web ページ「プライベートトレーニングについて」

2025/8/21　吉川　初芽　kikkawa@biz3.co.jp